大阪大学大学院情報科学研究科
博士前期課程

コンピュータサイエンス専攻
情報システム工学専攻
情報ネットワーク学専攻
マルチメディア工学専攻
バイオ情報工学専攻

【情報工学】

参考問題

1 【必須問題】アルゴリズムとプログラミング　（情報工学 1/10）

配点：(1) 15, (2-1) 30, (2-2) 20, (2-3) 15, (2-4) 20, (2-5) 25

図 1 に示す ANSI-C 準拠である C 言語のプログラム（program）A は，先入れ先出しキュー（first-in first-out queue）のプログラムである．プログラムに対する入力は図 2 に示すような形式（format）のファイル qdata.txt で与えられ，1 行目に操作の回数，2 行目以降の各行はキューに対する操作を表す．操作として 0 が与えられた場合はキューから 1 つデータを取り出し，正の整数（positive integer）が与えられた場合はその数をデータとしてキューに格納する．以下の各問に答えよ．

(1) プログラム A を図 2 に示す qdata.txt を与えて実行することを考える．プログラム A が動作開始してから終了するまでに出力する内容を書け．

(2) 図 1 の 13〜18 行目で定義されている関数 enqueue を以下の図で示すように変更して，図 1 に示すプログラム A を優先度つきキュー（priority queue）のプログラム B としたい．プログラム B では，要素をキューに挿入する際，関数 enqueue でキューに用いる配列の要素を優先度（priority）の高い順に整列する。なお，優先度はキューに挿入される数字自身で与えられ，値が大きいほど優先度が高いとする．以下の各問に答えよ．

```
void enqueue (int d) {
    int i;
    if ((qtail + 1) % NMAX != qhead) {
        for (i = qtail; i != qhead && [ (A) ] < [ (B) ]; i = (i - 1 + NMAX) % NMAX) {
            qd[i] = [ (C) ];
        }
        [ (D) ] = d;
        qtail = (qtail + 1) % NMAX;
    } else exit(1);
}
```

(2-1) 上記図中の関数 enqueue にある空欄(A)〜(D)を適切な式で埋めて，プログラム B を完成させよ．

(2-2) プログラム B で実現される優先度付きキューが保持できるデータ数は，最大でいくつか書け．

(2-3) プログラム B を図 2 に示す qdata.txt を与えて実行することを考える．プログラム B が動作開始してから終了するまでに出力する内容を書け．

(2-4) 図 2 の qdata.txt における 2 行目以降を並び変えたデータを入力としてプログラム B を実行することを考える．上記図中における「qd[i] = [(C)];」で示す行の実行回数が最大となるような qdata.txt を示せ．

(2-5) プログラム B の入力に与える操作の回数を N とする（ただし N>NMAX とする）．上記図中における「qd[i] = [(C)];」で示す行の実行回数が最大になる時は，入力 qdata.txt がどのようなデータとなっている場合か，説明せよ．

```
 1 #include <stdio.h>
 2 #include <stdlib.h>
 3 #define NMAX 256
 4
 5 int qd[NMAX], qhead, qtail;
 6
 7 void printqueue (void) {
 8         int i;
 9         for (i = qhead; i != qtail; i = (i + 1) % NMAX)
10                 printf("p %d¥n", qd[i]);
11 }
12
13 void enqueue (int d) {
14         if ((qtail + 1) % NMAX != qhead) {
15                 qd[qtail] = d;
16                 qtail = (qtail + 1) % NMAX;
17         } else exit(1);
18 }
19
20 void dequeue (void) {
21         int i;
22         if (qhead != qtail) {
23                 printf("d %d¥n", qd[qhead]);
24                 qhead = (qhead + 1) % NMAX;
25         } else exit(1);
26 }
27
28 int main (void) {
29         FILE *fp;
30         int n, i, d;
31
32         qhead = 0; qtail = 0;
33         fp = fopen("qdata.txt", "r");
34         fscanf(fp, "%d¥n", &n);
35         for (i = 0; i < n; i++) {
36                 fscanf(fp, "%d¥n", &d);
37                 if (d == 0) dequeue();
38                 else if (d > 0) enqueue(d);
39         }
40         fclose(fp);
41         printqueue();
42         return 0;
43 }
```

図１　プログラムＡ

```
7
4
0
2
3
0
5
1
```

図２　qdata.txt

2 【必須問題】計算機システムとシステムプログラム （情報工学 3/10）

配点： (1-1) 36, (1-2) 18, (2-1) 24, (2-2) 24, (2-3) 23

(1) ハードディスク（hard disk）に関する以下の各問に答えよ．

(1-1) 空欄 [(あ)] ～ [(う)] に入る適切な語を選択肢(A)～(G)から選べ．ただし，一つの選択肢を複数回用いてはならない．また，空欄 [<エ>] を適切な値で埋めよ．

ハードディスクは，データを記録する 1 枚以上のディスクを有するが，あるディスクの，ある表面に同心円上に並ぶ記録エリアをまとめて [(あ)] と呼ぶ．ハードディスク全体における同心円上にある全ての [(あ)] をまとめて [(い)] と呼ぶ．[(あ)] は，ハードディスク側の読み書きの最小単位 [(う)] の集合である．OS（operating system）側は [(う)] を一つ以上まとめたブロック（block）単位で読み書きを行う．

ディスク枚数 1, [(い)] 数 31,250, [(い)] あたりの [(あ)] 数 2, [(あ)] あたりの [(う)] 数 1,000, [(う)] あたりの容量 4 [キロバイト] (kbyte) であるハードディスクの容量は, [<エ>] [テラバイト] (Tbyte) である．ただし，1 [テラバイト] = 10^9 [キロバイト]とする．

選択肢

(A) スピンドルモーター（spindle motor） (B) セクタ（sector） (C) プラッタ（platter） (D) クラスタ（cluster）
(E) シリンダ（cylinder） (F) トラック（track） (G) ヘッド（head）

(1-2) ディスクの回転数が 10,000 [回転/分]（rpm）であるとき，ヘッドの移動を終えた後，ディスクが必要な位置に回転するまでの時間である回転待ち時間（回転遅延時間）（rotational delay）の平均を，単位を秒（s）で求めよ．

(2) ファイルが使用しているブロックを表 1 のように管理し，1 ブロックが 4 [キロバイト]であるファイルシステムを想定する．すでに表 1 のように 12 [キロバイト]のファイル A が格納されている場合について，以下の各問に答えよ．表 1 のファイルの使用ブロックの管理では，ファイルを格納する先頭のブロックのブロック番号を記憶し，行 ii に，行 i のブロック番号が示すブロックの次に来るブロックのブロック番号を格納し，次に来るブロックがない場合は“end”を格納する．

表 1

ファイル A の先頭のブロック番号：0

i：ブロック番号	0	1	2	3	4	5	6	7	8	9	10	11
ii：次のブロック番号	1	2	end									

(2-1)　ブロック番号が不連続な空きブロックの列で空き領域を管理する仕組みを表 1 に追加した表 2 のようなファイルシステムを想定する．表 2 における空き領域のブロック番号の列は，空き領域の先頭のブロック番号 3 で始まり，行iが 3 の列の行iiにより 6 と続き，最後は 11 で終わる．行 i が 11 の列の行 ii は空き領域の末尾であるため “end” となる．空きブロックを n 個使用するときは，この空き領域のブロック番号の列を先頭からそのままの順序で n 番目まで使用し，$n+1$ 番目を空き領域の新しい先頭のブロック番号とする．表 2 に対し，15 [キロバイト]のファイル B を追加した状態を解答欄に記述せよ．ただし，表 1 のように矢印を解答欄に記述する必要はない．

表 2

ファイル A の先頭のブロック番号：0　空き領域の先頭のブロック番号：3

i：ブロック番号	0	1	2	3	4	5	6	7	8	9	10	11
ii：次のブロック番号	1	2	end	6	7	4	8	9	5	10	11	end

(2-2)　表 1 の空き領域をビットマップ（bitmap）で管理するファイルシステムを想定する．空きブロックを取得するときは，表 3 の行 ii に対し未使用を示す 1 を，ブロック番号の小さい方から探し，1 を見つけたら即時に，行 i のブロック番号を取得し，ファイル B の格納に使用し，行iiの空き状態を 1 から 0 に更新するものとする．表 1 に対して，15 [キロバイト]のファイル B を追加した状態を解答欄に記述せよ．ただし，表 1 のように矢印を解答欄に記述する必要はない．

表 3

i：ブロック番号	0	1	2	3	4	5	6	7	8	9	10	11
ii：空き状態	0	0	0	1	1	1	1	1	1	1	1	1

(2-3)　大量のブロックを有し，問(2-2)のように空き領域をビットマップのみで管理し，空きブロックを先頭から探すファイルシステムを想定する．多数のファイルを追加し続け，空きブロック数が極めて少なくなったとき，空きブロックのブロック番号の探索時間は，空きブロック数が多いときと比べてどのように変化するか，理由とともに簡潔に述べよ．

3 【選択問題】離散構造　　（情報工学 5/10）

配点：(1) 15，(2) 25，(3) 25，(4-1) 10，(4-2) 15，(5-1) 10，(5-2) 10，(6) 15

非負の整数全体の集合を $\mathbb{Z}_0$ と書く．実数 x に対して，x を超えない最大の整数を $\lfloor x \rfloor$ と書く．任意の $n \in \mathbb{Z}_0$ に対して，実区間 $[0, \infty)$ から $[0, \infty)$ への写像（mapping）f_n を以下の漸化式（recurrence relation）により定める．

$$f_{n+1}(x) = x\lfloor f_n(x) \rfloor,\ f_0(x) = 1,\ x \in [0, \infty)$$

以下の各問に答えよ．

(1) 次の値をそれぞれ求めよ：$f_1(1.5)$，$f_2(2.5)$，$f_3(3.5)$．

(2) 命題（proposition）「ある $n \in \mathbb{Z}_0$ が存在し f_n は単射（injective）である」の真偽を判定せよ．その理由を述べよ．

(3) 命題「$n \in \mathbb{Z}_0$ が正かつ偶数ならば f_n は全射（surjective）である」の真偽を判定せよ．その理由を述べよ．

(4) $\mathbb{Z}_0$ における以下の二項関係（binary relation）を考える．

$$R = \{(k, \ell) \in \mathbb{Z}_0 \times \mathbb{Z}_0 \mid k = f_n(h/2) \text{ かつ } \ell = f_m(h/2) \text{ を満たす } h, n, m \in \mathbb{Z}_0 \text{ が存在する } \}$$

以下の各問に答えよ．

(4-1) R が反射的 (reflexive) であるかを判定せよ．その理由を述べよ．

(4-2) R が反対称的（antisymmetric）であるかを判定せよ．その理由を述べよ．

(5) 方程式

$$f_2(x) = 171 \tag{1}$$

を考える．以下の各問に答えよ．

(5-1) 非負の実数 x が方程式 (1) の解であると仮定する．ある実数 $\epsilon \in [0, 1)$ が存在し $x = 13 + \epsilon$ が成り立つことを示せ．

(5-2) 方程式 (1) の解を一つ求めよ．

(6) 方程式

$$f_3(x) = 171 \tag{2}$$

の解を一つ求めよ．

4 【選択問題】計算理論 （情報工学 6/10）

配点：(1) 15, (2) 15, (3) 30, (4) 35, (5) 30

文脈自由文法 (context-free grammar) は一般に $G = (V, T, P, S)$ で定められる．ここで V は変数 (variable; 非終端記号 nonterminal symbol) の集合，T は終端記号 (terminal symbol) の集合，P は生成規則 (production rule) の集合，S は出発記号 (start symbol) であり，$S \in V$ である．文脈自由文法 G_1 および G_2 を，それぞれ以下のように定める．

$$G_1 = (\{S,\ A\},\ \{+,\ *,\ (,\ ),\ a\},\ P_1,\ S),$$
$$P_1 = \{S \to A,\ S \to S + S,\ A \to (S),\ A \to a,\ A \to A * A\}$$

$$G_2 = (\{S,\ A\},\ \{+,\ *,\ (,\ ),\ a\},\ P_2,\ S),$$
$$P_2 = \{S \to a,\ S \to S + S,\ S \to A * A,\ A \to (S + S),\ A \to a,\ A \to A * A\}$$

G_1 と G_2 は，変数の集合，終端記号の集合，出発記号が共通であり，生成規則のみが異なる文脈自由文法である．G_1 および G_2 に関する以下の各問に答えよ．

(1) G_1 によって生成される長さが 5 の文字列を全て示せ．

(2) G_2 によって生成される長さが 5 の文字列を全て示せ．

(3) G_1 によって生成される言語 (文字列の集合) を W_1，G_2 によって生成される言語を W_2 とする．差集合 (difference set) $W_1 - W_2$ に含まれる文字列を，終端記号 'a' が数，終端記号 '$+$' が加算演算子，'$*$' が乗算演算子，'(' および ')' を括弧とする数式として解釈した時，差集合 $W_1 - W_2$ に含まれる文字列に存在する終端記号 '(' および ')' が数式の値に与える影響について記述せよ．

(4) G_1 はあいまい (ambiguous) である．例えば，$a + a + a$ の導出木 (derivation tree) は二つ存在する．G_1 によって生成される言語を生成し，あいまいでない文脈自由文法 G_3 を考える．

$$G_3 = (\{S,\ A\},\ \{+,\ *,\ (,\ ),\ a\},\ P_3,\ S),$$
$$P_3 = \{\boxed{\text{(あ)}}\}$$

空欄 (あ) に入れるべき生成規則を答えよ．生成規則の数は 6 以下とする．また，どのような方針でその生成規則を考案したのかについても記述せよ．

(5) G_1 によって生成される言語を認識する決定性有限オートマトン (deterministic finite automaton) は存在しない．G_1 を一部変更した以下の文脈自由文法 G_{1a} および G_{1b} によって生成される言語を認識する決定性有限オートマトンが存在するかどうかを，それぞれの文法について答えよ．存在しない場合には (正則言語でない場合には) そう考えた根拠を簡潔に記述せよ．存在する場合にはその決定性有限オートマトンの状態遷移図 (state transition diagram) を書くこと．状態遷移図の状態の総数は 4 以下とする．開始状態 (start state) には太い矢印 (thick arrow) を付与し，最終状態 (final state) は二重丸 (double circle) で表現すること．また，全ての状態において，全ての入力記号 (input symbol) に対する遷移を表記すること．

$$G_{1a} = (\{S,\ A\},\ \{(,\ ),\ a\},\ P_{1a},\ S),$$
$$P_{1a} = \{S \to A,\ A \to (S),\ A \to a\}$$

$$G_{1b} = (\{S,\ A\},\ \{+,\ *,\ a\},\ P_{1b},\ S),$$
$$P_{1b} = \{S \to A,\ S \to S + S,\ A \to a,\ A \to A * A\}$$

5 【選択問題】ネットワーク

配点: (1-1) 10, (1-2-1) 15, (1-2-2) 20, (1-3) 10, (1-4) 10, (1-5) 20, (2-1) 20, (2-2) 15, (2-3) 5

(1) イーサネット (Ethernet) を説明する以下の文章を読んで，以下の各問に答えよ．

> Carrier Sense Multiple Access/Collision Detection (CSMA/CD) を採用し，同軸ケーブル (coaxial cable) を媒体として用いるイーサネットでは，ステーション (station) は，同軸ケーブル上の［（あ）］を検出しない場合にのみフレーム (frame) を伝送する．次に，フレームを伝送している間に衝突 (collision) を検出すると，フレームの伝送を中断し，［（い）］を伝送する．その後，フレームの再送 (retransmission) を試みる．

(1-1) 上記の文章の空欄［（あ）］と［（い）］を適切な語句で埋めよ．

(1-2) 2 台のステーション A と B が同軸ケーブルで接続されており，他のステーションは接続されていないものとする．このとき，以下の各問に答えよ．なお，解答にあたって，プリアンブル (preamble) および開始フレーム識別子 (start frame delimiter) の長さは考えない．また，必要に応じて以下の記号を用いても良い．

- C [ビット/秒]：A-B 間の同軸ケーブルの帯域
- L [メートル]：A-B 間の同軸ケーブル長
- $L_{\max}$ [メートル]：イーサネットの最大同軸ケーブル長
- V [メートル/秒]：同軸ケーブル上の信号伝搬速度
- F_A [ビット]：A が送信したフレーム長
- F_B [ビット]：B が送信したフレーム長
- $F_{\min}$ [ビット]:イーサネットの最小フレーム長
- $F_{\max}$ [ビット]：イーサネットの最大フレーム長

(1-2-1) A が送信したフレームが B に到着するのに要する時間を答えよ．フレーム到着に要する時間は，A がフレームの最初のビット送信を開始してからそのビットが B に到着する時間とする．

(1-2-2) A と B が同時にフレームを送信したとする．B がフレームの衝突を検出するために B が送信したフレーム長が満たすべき条件を導出せよ．

(1-3) イーサネットで同軸ケーブルの最大長が規定されている理由を説明せよ．

(1-4) 同軸ケーブルの最大長を超えてフレームを伝送するためにブリッジが果たす役割を説明せよ．

(1-5) 再送によるフレームの再衝突を軽減するための仕組みを説明せよ．

(2) 送信者から受信者へ中継点一箇所を介して通信を行う．送信者から中継点までの通信路は記号誤り率 ε_1 の，中継点から受信者までの通信路は記号誤り率 ε_2 の，2 元対称通信路 (binary symmetric channel) とする ($\varepsilon_1 \leq 1/2$, $\varepsilon_2 \leq 1/2$)．中継点は受信した記号をそのまま送出する．送信者と受信者の間を一つの通信路 Γ とみなしたとき，Γ は 2 元対称通信路であることを以下の順に示せ．

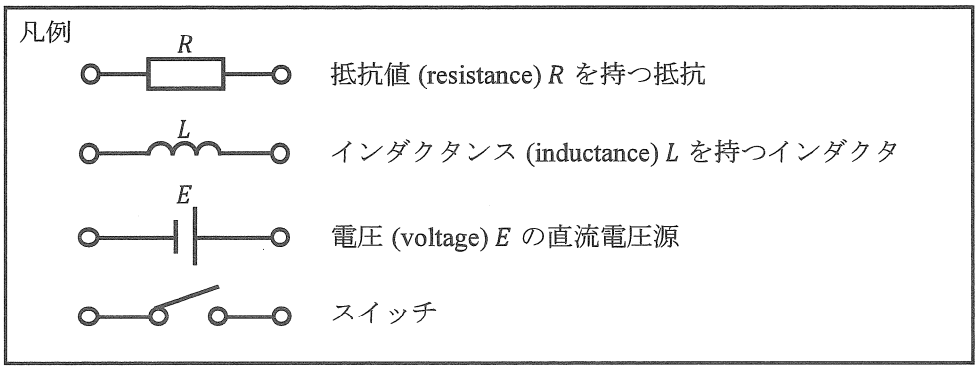
(2-1) 送信者から記号 $a \in \{0,1\}$ が送られたとの条件の下で，受信者が記号 $b \in \{0,1\}$ を受信する確率を $p(b|a)$ で表す．$p(0|0), p(1|1), p(0|1), p(1|0)$ をそれぞれ，$\varepsilon_1, \varepsilon_2$ で表せ．

(2-2) $a \neq b$ ならば，$p(b|a) \leq 1/2$ であることを示せ．

(2-3) 通信路 Γ が 2 元対称通信路であることを説明せよ．

配点：(1-1) 15, (1-2) 15, (1-3) 35, (2-1) 30, (2-2) 30

(1) 図 1 の回路 (circuit) について以下の問に答えよ．なお，図中の記号は以下の凡例に従うとする．また，スイッチ (switch) S は時刻 $t = 0$ で閉じる．時刻 $t < 0$ において回路は定常状態 (steady state) にあるとする．



(1-1) 時刻 $t < 0$ においてインダクタに流れる電流 i を求めよ．

(1-2) スイッチを閉じてから十分時間が経過し，回路が定常状態にあるときのインダクタに流れる電流 i を求めよ．

(1-3) 時刻 $t \geq 0$ においてインダクタに流れる電流 i を時刻 t の関数として表せ．

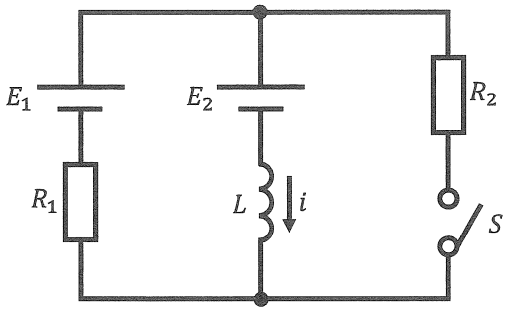


図 1

(2)　図 2 に示すクロック信号 (clock signal) CLK を持つ同期式順序回路 (synchronous sequential circuit) について考える．この順序回路の初期状態 (initial state) では $(Q_2, Q_1, Q_0) = (0,0,0)$ と出力 (output) される．Q_2，Q_1，Q_0 を出力する D フリップフロップ (D flip-flop) の入力 (input) を D_2，D_1，D_0 とするとき，以下の問に答えよ．

(2-1) 図 2 に示す同期式順序回路の状態遷移表 (state transition table) を表 1 に示す．表 1 の空欄 [a] ～ [h] を適切に埋めよ．

(2-2) 図 2 に示す同期式順序回路に組合せ論理回路 (combinational logic circuit) を接続して 4 進ダウンカウンタを実現したい．この組合せ論理回路の出力を (K_1, K_0) としたとき，(K_1, K_0) はクロックに同期して (0,0)→(1,1)→(1,0)→(0,1)→(0,0)→… と出力する．実現する 4 進ダウンカウンタの初期状態では $(K_1, K_0) = (0,0)$ と出力されるとき，K_1 と K_0 の論理関数 (logic function) の最簡積和形 (minimum sum of products) を Q_2，Q_1，Q_0 を用いて表せ．

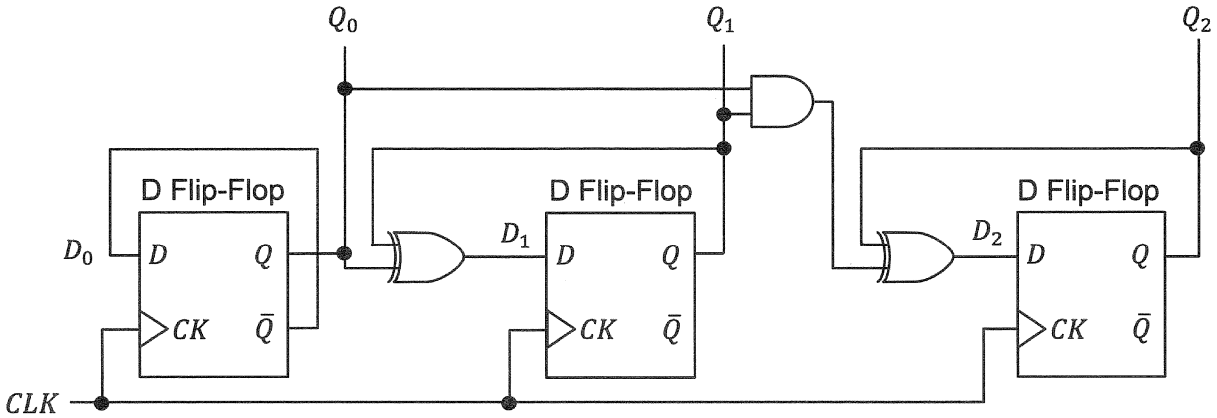


図 2

表 1

現状態 (Q_2, Q_1, Q_0)	次状態 (D_2, D_1, D_0)
(0, 0, 0)	(a)
(0, 0, 1)	(b)
(0, 1, 0)	(c)
(0, 1, 1)	(d)
(1, 0, 0)	(e)
(1, 0, 1)	(f)
(1, 1, 0)	(g)
(1, 1, 1)	(h)

7 【選択問題】数学解析と信号処理 （情報工学 10/10）

配点: (1)40, (2-1)35, (2-2)40, (3)10

区間 $(0, +\infty)$ で定義される関数 $f(t)$ と $g(t)$ のラプラス変換 (Laplace transform)$\mathcal{F}(s) = \mathcal{L}[f(t)](s)$ と $\mathcal{G}(s) = \mathcal{L}[g(t)](s)$ が存在するとする．線形時不変システム (linear time-invariant system) の入力を $f(t)$，出力を $g(t)$ と見なしたとき，このシステムの伝達関数 (transfer function)$H(s)$ は式 [1] のように求まる．

$$H(s) = \frac{\mathcal{G}(s)}{\mathcal{F}(s)} \quad [1]$$

一方，$H(s)$ は式 [2] でも求められる．ただし，$*$ は合成積 (convolution) を示す．

$$H(s) = \frac{\mathcal{L}[f(t) * g(t)](s)}{\mathcal{L}[f(t) * f(t)](s)} \quad [2]$$

以下の問に答えよ．

(1) 式 [3] が成り立つことを証明せよ．

$$\mathcal{L}[f(t) * g(t)](s) = \mathcal{F}(s)\mathcal{G}(s) \quad [3]$$

また，式 [3] を用いて式 [4] が成り立つことを示せ．

$$H(s) = \frac{\mathcal{G}(s)}{\mathcal{F}(s)} = \frac{\mathcal{L}[f(t) * g(t)](s)}{\mathcal{L}[f(t) * f(t)](s)} \quad [4]$$

(2) $f(t) = \cos t$，$g(t) = \sin t$ のとき，それらのラプラス変換と式 [1] を用いて $H(s)$ は求められる．一方，式 [3] を用いずに式 [2] の $\mathcal{L}[f(t) * g(t)](s)$ と $\mathcal{L}[f(t) * f(t)](s)$ を計算することで $H(s)$ を求めることを試みる．以下の問に答えよ．

(2-1) $\mathcal{L}[f(t) * g(t)](s)$ を計算するために，まず $f(t) * g(t)$ を計算する．区間 $(0, +\infty)$ において $f(t) * g(t)$ が式 [5] で求められることを証明せよ．

$$f(t) * g(t) = \cos t * \sin t = \frac{t}{2} \sin t \quad [5]$$

(2-2) 式 [2] の $f(t) * g(t)$ と $f(t) * f(t)$ のラプラス変換を求めることで $H(s)$ を求めよ．

(3) 式 [2] による伝達関数の導出方法は信号処理の分野でノイズ対策としてしばしば用いられる．その理由を述べよ．

1 【必須問題】アルゴリズムとプログラミング　（情報工学 1/17）

配点：(1) 12 点，(2) 12 点，(3) 14 点，(4) 25 点，(5-1) 20 点，(5-2) 30 点，(6) 12 点

図 1 に示す ANSI-C 準拠である C 言語のプログラム（program）は，複数の整数（integer）のデータ（data）を，二分木（binary tree）を利用して昇順（ascending order）に整列（sort）して出力（output）するプログラムである．図 1 のプログラムでは，配列（array）の添え字（index）が二分木の節点番号（node number）に対応している．ただし，二分木の根（root）の節点番号を 0 とし，節点番号が i の節点に子（child）がある場合，左の子の節点番号を $2i+1$，右の子の節点番号を $2i+2$ とする．また，配列に格納されたデータは，二分木の対応する節点のデータを示している．

整列するデータは図 2 に示すような形式（format）のファイル data.txt で与えられ，1 行目には整列するデータの個数 n（$\geqq 1$），2 行目以降の n 行には整列するデータの値（value）が書かれている．図 3 は，図 2 の data.txt を与えて図 1 のプログラムを実行した場合の，28 行目が実行される直前の配列 d に対応する二分木であり，丸が節点，丸の左側の数字が節点番号，丸の中の数字がデータの値，線分が枝（edge）を示している．図 1 のプログラムに関する以下の各問に答えよ．

(1) 40 行目で呼び出されている関数（function）sort で実現されている整列アルゴリズム（sorting algorithm）は，一般に何と呼ばれているか名称を答えよ．

(2) 図 2 の data.txt を与えてプログラムを実行した場合の，28 行目が実行された直後の配列 d に対応する二分木を図示せよ．ただし，図 3 にならい，丸で節点，丸の左側の数字で節点番号，丸の中の数字でデータの値，線分で枝を示すこと．

(3) 11 行目および 12 行目が実行されることにより，節点番号が current の節点のデータとその子のデータの間に成立する関係を説明せよ．

(4) 関数 sort で実現されている整列アルゴリズムの最悪時間計算量（worst case time complexity）を，整列するデータの個数 n を用いて理由と共にオーダ表記（order notation）で示せ．

(5) 関数 sort において，28 行目の実行時に関数 swap が呼び出される回数を $T(n)$ とする．n は整列するデータの個数である．28 行目を変更し，28 行目の for ループの繰り返し回数と $T(n)$ の最大値をできる限り削減（28 行目の実行に要する最悪時間計算量を削減）することを考える．以下の各小問に答えよ．

(5-1) 下記の（あ）～（え）を埋めて変更後の 28 行目を完成させよ．

```
for (i = [ (あ) ]; 0 <= i; i--) downh( [ (い) ] , [ (う) ] , [ (え) ] );
```

(5-2) 変更後のプログラムにおける $T(n)$ の n に関するオーダ表記を理由と共に示せ．

$\sum_{j=0}^{h} \frac{j}{2^j} = 2 - \frac{2+h}{2^h}$ を用いてよい．

(6) 下線（ア）～（エ）で示す条件式を必要に応じて変更し，データを降順（descending order）に整列して出力することを考える．変更後のプログラムにおける下線（ア）～（エ）の条件式をそれぞれ答えよ．

```
1   #include <stdio.h>
2   #include <stdlib.h>
3   void swap(int d[], int p, int q) {
4       int tmp;
5       tmp = d[p]; d[p] = d[q]; d[q] = tmp;
6   }
7   void downh(int d[], int n, int k) {
8       int child, current = k;
9       while (current < n / 2) {
10          child = current * 2 + 1;
11          if ((child + 1 < n) && (d[child] < d[child + 1])) child++;
                    (ア)                  (イ)
12          if (d[current] < d[child]) swap(d, current, child);
                       (ウ)
13          else break;
14          current = child;
15      }
16  }
17  void uph(int d[], int k) {
18      int parent, current = k;
19      while (0 < current) {
20          parent = (current - 1) / 2;
21          if (d[parent] < d[current]) swap(d, parent, current);
                          (エ)
22          else break;
23          current = parent;
24      }
25  }
26  void sort(int d[], int n) {
27      int i;
28      for (i = 1; i < n; i++) uph(d, i);
29      for (i = n - 1; 0 < i; i--) {swap(d, 0, i); downh(d, i, 0);}
30  }
31  int main() {
32      int i, N, *D;
33      FILE* fp;
34      fp = fopen("data.txt", "r");
35      fscanf(fp, "%d", &N);
36      D = (int*) malloc(sizeof(int) * N);
37      for (i = 0; i < N; i++) fscanf(fp, "%d", &D[i]);
38      fclose(fp);
39
40      sort(D, N);
41
42      for (i = 0; i < N; i++) printf("%d ", D[i]);
43      printf("\n");
44      free(D);
45      return 0;
46  }
```

図 1　プログラム

```
6
40
30
50
10
60
20
```

図 2　data.txt

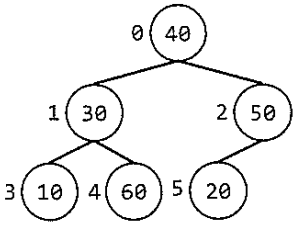


図 3　二分木の例

2 【必須問題】計算機システムとシステムプログラム （情報工学 3/17）

配点： (1-1) 10 点，(1-2-1) 15 点，(1-2-2) 15 点，(1-2-3) 15 点，(1-2-4) 10 点，
(2-1) 18 点，(2-2) 20 点，(2-3) 12 点，(2-4) 10 点

(1) 計算機 (computer) における数の表現に関する以下の各小問に答えよ．解答は全て解答用紙の太線枠内に書くこと．

(1-1) 以下の文章の空欄 [(a)] ～ [(e)] に当てはまる最も適切な語句または数値を，下記の選択肢から選び，記号で答えよ．

数の表現形式として固定小数点 (fixed point number) 表現と浮動小数点 (floating point number) 表現がある．科学技術計算の分野では，[(a)] 等の理由から [(b)] 表現が多く用いられ，小型機器の制御等の分野では，[(c)] 等の理由から [(d)] 表現が多く用いられる．いずれの表現形式においても，指数の底 (base of exponent) が 2 の場合，10 進数 (decimal number) の [(e)] を誤差 (error) 無く表現することはできない．

［選択肢］

(ア) 固定小数点	(イ) 浮動小数点
(ウ) 演算回路 (arithmetic circuit) が簡素になる	(エ) 表現できる数の範囲が広い
(オ) 小数を誤差無く表現できる	(カ) 0.5
(キ) 0.25	(ク) 0.1

(1-2) 以下の浮動小数点表現に関し，(1-2-1)～(1-2-4) に答えよ．(1-2-1)～(1-2-3) については導出過程も示すこと．

- 浮動小数点数 A を $A = (-1)^s \times 2^{e-7} \times (1+m)$ と表し，符号部 s (sign)，指数部 e (exponent)，仮数部 m (significand, mantissa) を最上位ビットから順にメモリに格納する．
- 符号部 s は 1 ビットである．$s \in \{0, 1\}$ をメモリに格納する．
- 指数部 e は 4 ビットである．7 余り表現とし，e を 4 ビットの符号無し 2 進整数 (unsigned binary integer) で表したビット列をメモリに格納する．$e = 0$ および $e = 15$ は $\pm\infty$ などの特殊な数を表現するのに用いる．
- 仮数部 m は 9 ビットである．正規化 (normalization) し，整数部分の 1 は記録しない (隠しビット)．小数部分の m を 2 進数の小数で表し，小数点以下の 9 ビットをメモリに格納する．丸め (rounding) 処理は切り捨て (rounding down) とする．非正規化 (denormalization) 表現は用いない．

(1-2-1) この表現形式で表すことのできる，正の最大値を 10 進数の小数で示せ．但し，無限大 (infinity) は除く．

(1-2-2) この表現形式で表すことのできる，正の最小値を 10 進数の小数で示せ．

(1-2-3) 10 進数 -36.66 をこの形式で表現し，そのビット列を示せ．

(1-2-4) 繰り返し計算を行う際，丸め処理として切り捨てを用いると，問題が生ずる場合がある．どの様な問題か，理由も含めて示せ．

(2)　仮想記憶 (仮想メモリ; virtual memory) に関して以下の各小問に答えよ．

(2-1)　次の説明文を読み，空欄 [(a)] ～ [(i)] にあてはまる最も適切な語句を下の (ア) ～ (ソ) の選択肢から選んで記号で答えよ．一つの選択肢を複数回用いてはならない．

仮想記憶は，主記憶 (primary storage, main memory) と補助記憶 (secondary storage, auxiliary memory) を用いて，主記憶の容量よりも大きい [(a)] を提供する．プロセスが指すアドレスは [(b)] と呼ばれる．プロセスの実行においてメモリアクセスのたびに，アドレス変換テーブル (mapping table) に基づいて [(b)] を [(c)] に変換する．一般的に，アドレス変換テーブルは [(d)] 上に構成する．

アドレス変換を固定長 (fixed length) のブロック単位で行う方式を [(e)] と呼び，可変長 (variable length) のブロック単位で行う方式を [(f)] と呼ぶ．主記憶の利用効率 (memory efficiency) の観点からこれらの方式を比較すると，[(e)] 方式の利点は，主記憶領域の割り付けと解放を繰り返しても [(g)] がほとんど発生しない点である．一方，欠点は，固定長のページ単位で領域を割り付けるために [(h)] が発生しやすい点である．[(h)] は，[(i)] の大きさと比較して使用領域が小さいプロセスを実行する場合に顕著である．

【選択肢】

(ア) 主記憶	(イ) ページ (page)	(ウ) 内部断片化 (internal fragmentation)
(エ) スラッシング (thrashing)	(オ) 実アドレス (real address)	(カ) 仮想アドレス (virtual address)
(キ) アドレス空間 (address space)	(ク) 補助記憶	(ケ) メモリ階層 (memory hierarchy)
(コ) ページング (paging)	(サ) セグメント (segment)	(シ) 外部断片化 (external fragmentation)
(ス) ページフォールト (page fault)	(セ) レジスタ (regsiter)	(ソ) セグメンテーション (segmentation)

(2-2)　プロセスが次に示すページ参照列 S で処理を行う場合を想定する．ページ枠 (page frame) の数が 3 のページング方式を採用した場合に，ページフォールトが発生するページ参照に○を記入せよ．ページ置換アルゴリズムに LRU (least recently used) および FIFO (first in first out) を用いる場合についてそれぞれ回答せよ．但し，初期状態では全てのページ枠は空である．

ページ参照列 S : 0, 1, 2, 0, 3, 1, 4, 3, 2, 3, 1, 2, 4

(2-3)　ページング方式を採用したシステムを想定する．このシステムでは，1 回の主記憶へのアクセスに 2 [マイクロ秒] を要し，ページフォールトが発生するとそれに加えて 8 [ミリ秒] のオーバヘッド (overhead) が発生する．1 命令あたり平均 2 回のメモリアクセスを行うとき，ページフォールトによる性能低下をページフォールトの無い状態と比較して平均 10%以下に抑えるために，許容できるページフォールトの確率 P の上限を求めよ．導出過程も示せ．なお，ページフォールトが発生しない場合のメモリアクセス時間は主記憶へのアクセス時間に等しく，演算時間および他のオーバヘッドは無視できるものとする．

(2-4)　ページング方式を採用し，ページ置換アルゴリズムに LRU を用いたシステムにおいて，ページフォールトを削減する方法を具体的かつ簡潔に記述せよ．但し，実行する各プロセスのページ参照列は変更しないものとする．

3 【選択問題】離散構造

配点：(1-1) 16 点，(1-2-1) 8 点，(1-2-2) 15 点，(1-2-3) 16 点，(1-2-4) 10 点
(2-1) 10 点，(2-2-1) 25 点，(2-2-2) 25 点

(1) 一階述語論理（first-order predicate logic）に関する以下の各小問に答えよ．ただし，論理式（logic formula）の記述には以下の記号を用いる．$\forall$, $\exists$はそれぞれ全称作用素（universal quantifier），存在作用素（existential quantifier）であり，$\Leftrightarrow$, $\rightarrow$, $\wedge$, $\vee$, $\neg$ はそれぞれ等価（equivalence），含意（implication），論理積（conjunction，and），論理和（disjunction，or），否定（negation，not）を表す論理演算子（logical operator）である．

ロボット（robot），バッテリー（battery），移動可能な台（stepladder）が部屋にあり，ロボットは台を移動させることができる．部屋の中の位置 *goal* に台を移動させ設置し，台に登ることで，ロボットはバッテリーを握って取得できる．ロボットが実行可能な動作を論理式で記述し，それらの論理式のもとで，ロボットがバッテリーを取得した状態を表す論理式が成り立つことを示すことで，ロボットがバッテリーを取得できることを示したい．ロボットは動作をすることで，ロボットや台の状況を表す状態を遷移させることができる．状態は，ロボットが台に登っているかどうか，ロボットがバッテリーを取得しているかどうか，および台の位置で決まるものとする．本問では，ロボットの動作を表す関数（function）として，遷移前の状態を s としたときに，遷移後の状態 s' を返すような以下の関数を用いる．

- $move(s, x)$：状態 s からロボットが台を位置 x に移動させ設置した状態 s' を返す．
- $climb(s)$：状態 s からロボットが台に登った状態 s' を返す．
- $grasp(s)$：状態 s からロボットがバッテリーを握って取得した状態 s' を返す．

また，以下の述語（predicate）を用いる．

- $on(s)$：状態 s においてロボットが台に乗っている．
- $sl(s, x)$：状態 s において台が位置 x に設置されている．
- $have(s)$：状態 s においてロボットがバッテリーを取得している．

本問では，以下の閉論理式（closed formula）A〜D が成り立つことを仮定する．

- $A = \forall s\ \forall x(sl(s, x) \rightarrow sl(climb(s), x))$：どのような状態で，どのような位置に台が設置されていても，ロボットはその位置において台に登ることができ，登った後の状態に遷移する．
- $B = \forall s((sl(s, goal) \wedge on(s)) \rightarrow have(grasp(s)))$：どのような状態においても，位置 *goal* に設置されている台にロボットが乗っていれば，ロボットはバッテリーを握ることができ，取得した状態に遷移する．
- $C = \forall s\ \forall x(\neg on(s) \rightarrow sl(move(s, x), x))$
- $D = \forall s\ on(climb(s))$

以下の各小問に答えよ．

(1-1) 閉論理式 C および D が表していることをそれぞれ説明せよ．

(1-2) 閉論理式 A〜D を用いて，ロボットがバッテリーを取得できることを次の手順で示す．以下の各問に答えよ．

(1-2-1) 「ロボットがバッテリーを取得している状態が存在する」ことを表す閉論理式 E を示せ．

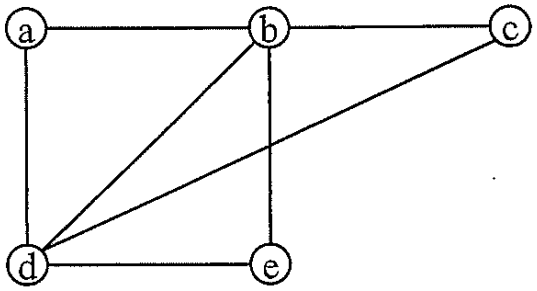
(1-2-2) 論理式 $F = (A \wedge B \wedge C \wedge D) \rightarrow E$ としたとき，$\neg F$ の冠頭標準形（prenex normal form）$\neg F_p$ を示せ．冠頭標準形は，すべての限定作用素（quantifier）が先頭にある閉論理式である．ただし，$\neg F_p$ は連言標準形（和積標準形：conjunctive normal form）で示し，最終的に得られる $\neg F_p$ には，記号 A, B, C, D, E を用いないこと．

(1-2-3) $\neg F_p$ をスコーレム化し，導出原理を用いて，充足不能であることを示そうとしたが，空節を導くことができなかった．これは，ロボットの初期条件に関する論理式 I が考慮されていないためである．導出原理を適用して，E を成り立たせるための論理式 I を示せ．導出過程も示せ．ただし，論理式 I は初期状態 s_0 について記述したものであり，台の初期位置は $goal$ ではないこととする．

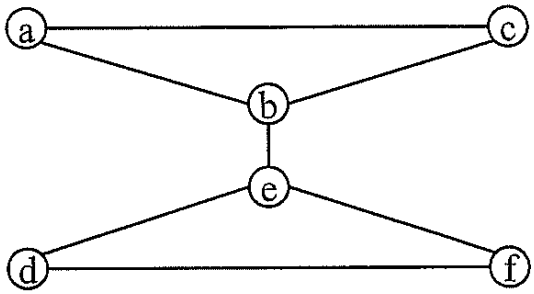
(1-2-4) 論理式 I を F の前提部に追加，つまり $F = (A \wedge B \wedge C \wedge D \wedge I) \rightarrow E$ と修正し，再度導出原理を適用することで，空節を導くことができた．このとき，論理式 E に該当する節の変数への代入から，状態の遷移を知ることができるが，この状態の遷移を，代入された項を示すことにより答えよ．

(2) 頂点（node）の集合，および辺（edge）の集合がいずれも空（empty）でない無向グラフ（undirected graph）に関する以下の各小問に答えよ．

(2-1) 以下のグラフ G_1 および G_2 それぞれについて，それらはどの一辺を取り除いても連結（connected）であるか，そうでないかを答えよ．そうでない場合は，どの辺を取り除くと連結でなくなるかも答えよ．



G_1



G_2

(2-2) 各頂点の次数（degree）が偶数（even number）であり，かつ連結である任意のグラフ G について，以下の各問に答えよ．ただし，G は同じ頂点を結ぶ辺を持たず，かつ，各頂点組に対し，それらの頂点を結ぶ辺は高々一つであるとする．なお，隣接する頂点の系列を，それに対応する辺の系列により表したものを経路（path）とよび，始点と終点が同じ経路を閉路（cycle）とよぶ．例えば小問 (2-1) の G_1 の経路 $(b,c),(c,d),(d,b),(b,e),(e,d),(d,a),(a,b)$ は閉路である．

(2-2-1) G から任意の辺 (u,v) を取り除いて得られるグラフ G' は，頂点 v から頂点 u への経路を持つことを示せ．G' が連結でないと仮定し，頂点の次数に関する矛盾（contradiction）を以下の補題（lemma）を用いて導くとよい．

（補題）任意のグラフにおいて，すべての頂点の次数の和は偶数である．

(2-2-2) 任意の頂点 u に対し，G は u を含み，各辺を高々一つだけ含む閉路を持つことを示せ．(2-2-1) は成り立つとしてよい．

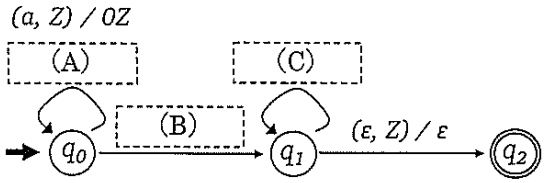
4 【選択問題】計算理論

配点：(1-1) 15 点, (1-2-1) 10 点, (1-2-2) 25 点, (1-3) 10 点, (2-1) 40 点, (2-2) 25 点

(1) アルファベット (alphabet) $\Sigma_{ab} = \{a, b\}$ で構成される言語 (language) $L_{ab} = \{a^n b^n \mid n$ は正の整数 $\}$ を考える．なお，a^n は a が n 個連結した文字列 (string) を表す．例えば，$a^3 = aaa$ である．また，ε は空文字列 (empty string) を表す．言語 L_{ab} を認識 (受理) するオートマトン (automaton) に関する以下の各小問に答えよ．

(1-1) プッシュダウンオートマトン (pushdown automaton) は $(Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ で表される．Q は状態 (state) の有限集合，Σ は入力アルファベット (input alphabet)，Γ はスタックアルファベット (stack alphabet)，δ は遷移関数 (transition function)，q_0 は開始状態 (start state)，Z はスタックの開始記号，F は最終状態 (final state) の集合である．最終状態による受理 (acceptance by final state) を行うプッシュダウンオートマトンを利用して，言語 L_{ab} を認識することを考える．

下図は言語 L_{ab} を認識する非決定性 (non-deterministic) プッシュダウンオートマトンの状態遷移図 (state transition diagram) であり，開始状態は q_0，最終状態は q_2 である．状態遷移図の空欄 (A), (B), (C) に必要な動作を，$(r, s)/t$ の形式で答えよ．ただし，$(r, s)/t$ $(r \in \Sigma \cup \{\varepsilon\}, s \in \Gamma, t \in \Gamma^*)$ は，入力から読み出す記号が r でスタックから取り出す記号が s のときに，スタックからその s を取り去り，t をスタックに押し込むことを意味する．t が複数の記号の場合は右側の記号から順にスタックに押し込む．$(a, Z)/0Z$ は，a を読み出すときにスタックから Z を取り去り，スタックに Z と 0 をこの順で押し込むことを意味する．t が ε の場合は，スタックに記号を押し込まない．r が ε の場合は，入力から記号を読み出さないで遷移を行う．入力アルファベットは $\Sigma = \Sigma_{ab}$，スタックアルファベットは $\Gamma = \{Z, 0\}$ とする．



(1-2) 決定性有限オートマトン (deterministic finite automaton) は $(Q, \Sigma, \delta, q_0, F)$ で表される．Q は状態の有限集合，Σ は入力アルファベット，δ は遷移関数，q_0 は開始状態，F は最終状態の集合である．入力アルファベットは $\Sigma = \Sigma_{ab}$ とする．このとき，(1-2-1) および (1-2-2) に答えよ．

(1-2-1) 言語 $L_2 = \{a^n b^n \mid n = 2\}$ を認識する決定性有限オートマトン，および，言語 $L_3 = \{a^n b^n \mid n = 3\}$ を認識する決定性有限オートマトンの状態遷移図をそれぞれ書け．なお，上図に示すように開始状態には太い矢印 (thick arrow) を付与し，最終状態は二重丸 (double circle) で表現すること．また，全ての状態において，全ての入力記号に対する遷移を表記すること．

(1-2-2) 言語 L_{ab} を認識する決定性有限オートマトンは存在しないことを背理法 (proof by contradiction) により証明したい．以下の空欄 (D) に適切な文章を記述し，証明を完成させよ．

証明：言語 L_{ab} を認識する決定性有限オートマトンが存在すると仮定し，そのオートマトンの状態数を k とする．文字列 $a^k b^k$ が与えられたとき，a を i 個 $(0 \leq i \leq k)$ 読み終えた時点での状態を p_i とすると，

(D)

よって仮定と矛盾するため，言語 L_{ab} を認識する決定性有限オートマトンは存在しない．　（証明終）

(1-3) 言語 L_{ab} を認識する決定性有限オートマトンは存在しないが，言語 $L'_{ab} = \{a^n b^m \mid n$ および m は正の整数 $\}$ を認識する決定性有限オートマトンは存在する．言語 L'_{ab} を認識する決定性有限オートマトンの状態遷移図を書け．なお，上図に示すように開始状態には太い矢印を付与し，最終状態は二重丸で表現すること．また，全ての状態において，全ての入力記号に対する遷移を表記すること．

(2) 文脈自由文法 (context-free grammar) は，変数 (variable) の有限集合 V, 終端記号 (terminal symbol) の有限集合 T, 生成規則 (production rule) の有限集合 P, および開始記号 (start symbol) S の組により定められる．変数は非終端記号 (nonterminal symbol) と呼ぶこともある．Chomsky 標準形 (Chomsky normal form) とは，文脈自由文法のうち，生成規則が，(i) $X \to YZ$, (ii) $X \to a$ のいずれかの形をしたものである．ただし，X, Y, Z は変数，a は終端記号である．

ここで，Chomsky 標準形で与えられる文脈自由文法 $G = (V, T, P, S)$ および文字列 $w = a_1 a_2 \cdots a_n$（ただし，各 a_i は T に属する）に対し，w が G によって生成されるか否かを，動的計画法 (dynamic programming) によって効率的に判定するアルゴリズムを考える．まず，整数 i, j のうち $1 \le i \le j \le n$ であるものに対し，集合

$$M[i,j] = \{X \in V \mid \text{文法 } G \text{ において，} w \text{ の連続した部分文字列 } a_i a_{i+1} \cdots a_j \text{ を変数 } X \text{ から導出できる}\}$$

を定める．以下に擬似コードで示したアルゴリズムでは各 $M[i,j]$ を $M[1,1], M[2,2], \ldots, M[n,n]$, $M[1,2], M[2,3], \ldots, M[n-1,n], M[1,3], \ldots$ の順に $M[1,n]$ まで計算している．最後の判定では，w が G によって生成されるとき，かつそのときに限り $S \in M[1,n]$ という性質を利用している．

```
入力 w = a_1 a_2 ... a_n に対し，集合 M[i,j] (1 ≤ i ≤ j ≤ n) の初期値をすべて空集合 ∅ とする．
for i = 1 to n do                    // 部分文字列の長さが 1 の場合の計算．
  (X → a_i) ∈ P であるすべての X ∈ V を M[i,i] に追加．
end for
for ℓ = 2 to n do                    // 他のすべての M_{i,j} の計算．ℓ は部分文字列の長さを表す．
  for i = 1 to n − ℓ + 1 do          // i は部分文字列の先頭位置を表す．
    j = i + ℓ − 1 とする．            // j は部分文字列の終了位置を表す．
    for k = i to j − 1 do            // k は部分文字列の分割位置の一つ前を表す．
      ある Y ∈ M[i,k], Z ∈ M[k+1,j] に対し，(X → YZ) ∈ P であるすべての X ∈ V を M[i,j] に追加．
    end for
  end for
end for
S ∈ M[1,n] であれば Yes を，そうでなければ No を出力して終了．          // 判定
```

文脈自由文法 $G_1 = (V_1, T_1, P_1, S_1)$ を，$V_1 = \{A, B, C, D, S\}, T_1 = \{a, b\}, S_1 = S$,

$$P_1 = \{S \to AB, A \to CA, A \to AA, A \to a, B \to DB, B \to b, C \to a, D \to b\},$$

と定める．例えば，文法 G_1 と文字列 bab に対し，上記のアルゴリズム実行した場合の計算は以下のように行われる．

- まず，$M[1,1], M[2,2], M[3,3]$ が決まる．
- 次に，部分文字列の長さが $\ell = 2$ の場合，先頭位置は $i = 1, 2$ の二つの場合があり，
  - $i = 1$ ならば $j = 2$ であり，$k = 1$ として，$Y \in M[1,1], Z \in M[2,2]$ を調べることで $M[1,2]$ が決まる．
  - $i = 2$ ならば $j = 3$ であり，$k = 2$ として，$Y \in M[2,2], Z \in M[3,3]$ を調べることで $M[2,3]$ が決まる．
- 最後に，部分文字列の長さが $\ell = 3$ の場合，先頭位置は $i = 1$ の場合だけであり，
  - $i = 1$ ならば $j = 3$ である．このとき，$k = 1$ として，$Y \in M[1,1], Z \in M[2,3]$ を調べ，$k = 2$ として，$Y \in M[1,2], Z \in M[3,3]$ を調べることで，$M[1,3]$ が決まる．

終了時における $M[i,j]$ の内容は以下の表のとおりである．この場合，$S \notin M[1,3]$ であるため No が出力される．

$M[i,j]$	$j=1$	$j=2$	$j=3$
$i=1$	$\{B,D\}$	$\emptyset$	$\emptyset$
$i=2$	—	$\{A,C\}$	$\{S\}$
$i=3$	—	—	$\{B,D\}$

以下の各小問に答えよ．

(2-1) 文法 G_1 および文字列 $w_1 = aaab$ に対し，上記のアルゴリズムを実行した場合，終了時における $M[i,j]$ の内容を，上記の例のように，表として示せ．

(2-2) 文脈自由文法 $G_2 = (V_2, T_2, P_2, S_2), V_2 = \{A, B, S\}, T_2 = \{a, b\}, S_2 = S,$

$$P_2 = \{S \to AB, S \to SA, S \to BS, S \to \varepsilon, A \to a, B \to b\}$$

を考える．ここで，ε は空文字列 (empty string) である．文法 G_2 は，上で説明した Chomsky 標準形の制約を満たしていないため，上記のアルゴリズムに対して正しく動作しない場合がある．つまり，ある入力文字列に対しては間違った判定をする．そのような入力文字列として長さ 2 以上のものを具体的に示し，正しく動作しないことを説明せよ．

5 【選択問題】ネットワーク　（情報工学 10/17）

配点: (1-1) 10 点，(1-2-1) 15 点，(1-2-2) 25 点，(1-3) 15 点，(1-4)15 点，
(2-1) 20 点，(2-2-1) 10 点，(2-2-2) 5 点，(2-2-3) 10 点

(1) ビット誤り (bit error) が発生することがある伝送路 (transmission line) を用いて，送信端末 (terminal) の応用プログラム (application program) がファイル (file) を受信端末の応用プログラムに転送するプロトコル (protocol) を考える．プロトコルでは，応用プログラムがファイルをデータ (data) に分割し，送信端末が分割されたデータをデータパケット (packet) を用いて転送する．以下の各小問に答えよ．なお，端末は受信したパケットのビット誤りを検出できるものとする．さらに，小問 (1-1) から (1-3) では，パケットは紛失 (loss) しないことを仮定する．

(1-1) ビット誤りが発生したデータパケットを再送 (retransmit) するプロトコルを考える．このプロトコルをプロトコル 1 と呼び，その送信端末の状態遷移図 (state transition diagram) を図 1 に示す．状態遷移では，発生したイベント (event) に基づきアクション (action) が実行され，次の状態に遷移する．イベントとアクションの組は図 1 の破線の枠内に示す通りに記載する．使用するイベント，アクションとそれらに付随する変数 (variable) を図 2 に示す．プロトコル 1 では，送信端末はシーケンス番号 (sequence number) に 0 を設定してデータパケットを送信する．一方，受信端末は，データパケットを受信すると，ビット誤りが無い場合は確認応答 (positive acknowledgement) パケットを返送し，ビット誤りを検出すると否定確認応答 (negative acknowledgement) パケットを返送する．以降，これら二つのパケットを応答パケットと総称する．なお，プロトコル 1 では，応答パケットのシーケンス番号には 0 を設定し，応答パケットにはビット誤りが発生しないものとする．図 1 の状態遷移図の空欄 （あ） ， （い） に，それぞれ一つのアクションを埋めよ．

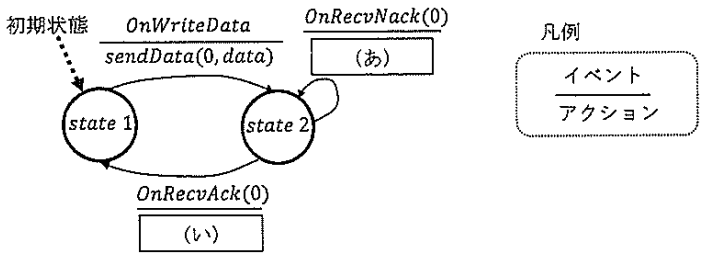


図 1: プロトコル 1 の送信端末の状態遷移図

変数　*data*
イベント　*OnWriteData*：応用プログラムが転送するデータを受領し，変数 *data* に格納．
　OnRecvAck(0 または 1)：シーケンス番号が 0 または 1 に設定された確認応答パケットを受信．
　OnRecvNack(0 または 1)：シーケンス番号が 0 または 1 に設定された否定確認応答パケットを受信．
アクション　A：何もしない．
　sendData(0 または 1, *data*)：シーケンス番号が 0 または 1 に設定され，変数 *data* の中身をペイロード (payload) として持つデータパケットを受信端末に送信する．

図 2: 送信端末の変数，イベント，アクション

(1-2) 次に，プロトコル 1 を改良し，応答パケットにビット誤りが発生したとき，データパケットを再送するプロトコルを考える．ここで，図 3 に示すイベントを追加する．(1-2-1) および (1-2-2) に答えよ．

(1-2-1) 送信端末の状態遷移図を図 4 とするプロトコルをプロトコル 2 と呼ぶ．しかし，プロトコル 2 では，送信端末の応用プログラムが転送したファイルの中身と，受信端末の応用プログラムが受領したファイルの中身が一致しないことがある．この問題が発生する条件を一つ示し，ファイルの中身が一致しない理由を説明せよ．

イベント　*OnRecvBiterr*：ビット誤りがある応答パケットを受信．

図 3: 追加するイベント

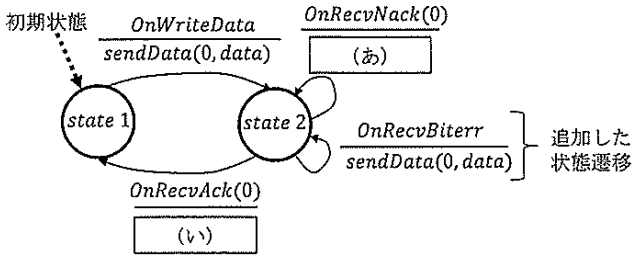


図 4: プロトコル 2 の送信端末の状態遷移図

(1-2-2) 応答パケットのビット誤りに対処できるように，送信端末がデータパケットに，0 と 1 のシーケンス番号を交互に設定し，最初のデータパケットには 0 を設定する方針でプロトコル 3 を設計する．受信端末は，データパケットを受信すると，ビット誤りの有無にかかわらず，確認応答パケットを返送する．データパケットにビット誤りが無いとき，次に受信することを期待するシーケンス番号 (next expected sequence number) を確認応答パケットに設定する．一方，ビット誤りを検出すると，受信することを期待していたシーケンス番号を確認応答パケットに設定する．図 5 の空欄 （う） ～ （き） に，それぞれ一つのイベントあるいはアクションを埋めることで，プロトコル 3 の送信端末の状態遷移図を完成させよ．

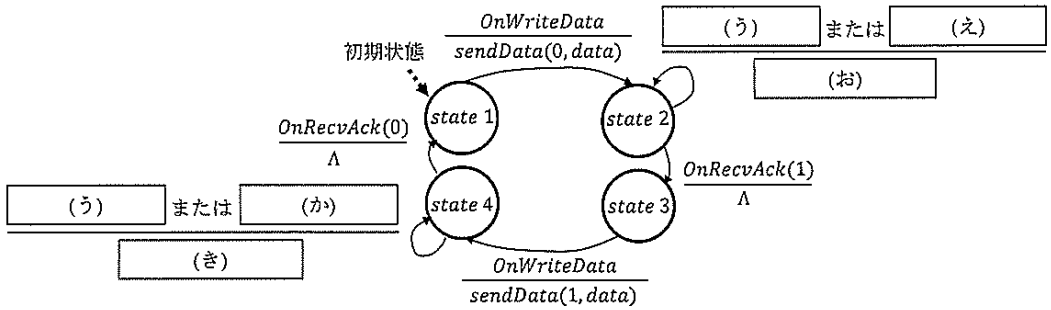


図 5: プロトコル 3 の送信端末の状態遷移図

(1-3) 伝送路の帯域 (bandwidth) は 1600 [ビット/秒]，伝搬遅延 (propagation delay) は 10 [ミリ秒]，送信端末と受信端末のプロトコルの処理遅延 (processing delay) は 0 [ミリ秒]，確認応答パケットは 8 ビット長と仮定する．このとき，24 ビット長のデータパケットを転送する場合の，プロトコル 3 の送信端末が，1 秒間に転送できる最大のデータパケット数を求めよ．あわせて，計算過程も示せ．

(1-4) 本小問では，伝送路でパケットが紛失することがあると仮定する．プロトコル 3 において，受信端末の動作を変えることなく，送信端末がデータパケットや確認応答パケットの紛失に対処する方法を一つ説明せよ．説明にあたっては，状態遷移図を示す必要はない．

(2) 伝送路における誤り検出 (error detection) に関する以下の各小問に答えよ．

(2-1) 以下の文章の空欄 （あ） ～ （お） を埋めよ．

ある情報が送信された後，伝送路上において電気的雑音 (noise) により受信された情報に誤りが発生する場合がある．これに対処するためには，誤り検出が必要となる．

誤り検出の例として，（あ） が挙げられる．（あ） とは，元の情報ビットに対し，1 の総数が奇数または偶数となるようなチェックビット (check bit) を付与する方法である．例えば，2^7 個の文字や記号を 7 ビットで表現する符号の場合，符号語 (codeword) 間の最小ハミング距離 (minimum Hamming distance) は 1 であるが，チェックビットを付与することによって符号語間の最小ハミング距離は （い） になる．一般に，最小ハミング距離が （い） の符号は，（う） ビットの誤りをすべて検出できる．

2 ビット以上の誤りや，あるいは符号語の連続したビットに発生する誤りである （え） にも対処できるように，巡回符号 (cyclic code) が広く用いられる．巡回符号は，（え） の検出に優れており，生成多項式 (generator polynomial) の次数が r であるとき，（お） ビット以下の （え） を必ず検出できる．

(2-2) 2 元巡回符号 (binary cyclic code) を用いた誤り検出について，(2-2-1)～(2-2-3) に答えよ．

(2-2-1) 生成多項式 $G(x)$ の 2 元巡回符号を考える．$G(x)$ の次数を r とし，符号長を n とする．このとき，情報ビットの長さ m は $n-r$ である．情報ビット “$a_{m-1}a_{m-2}\cdots a_1a_0$” を多項式

$$M(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_1x + a_0$$

で表現するとき，$M(x)$ から符号語多項式 (code polynomial) $F(x)$ を生成する方法を説明せよ．

(2-2-2) 生成多項式 $G(x)$ として

$$G(x) = x^4 + x + 1$$

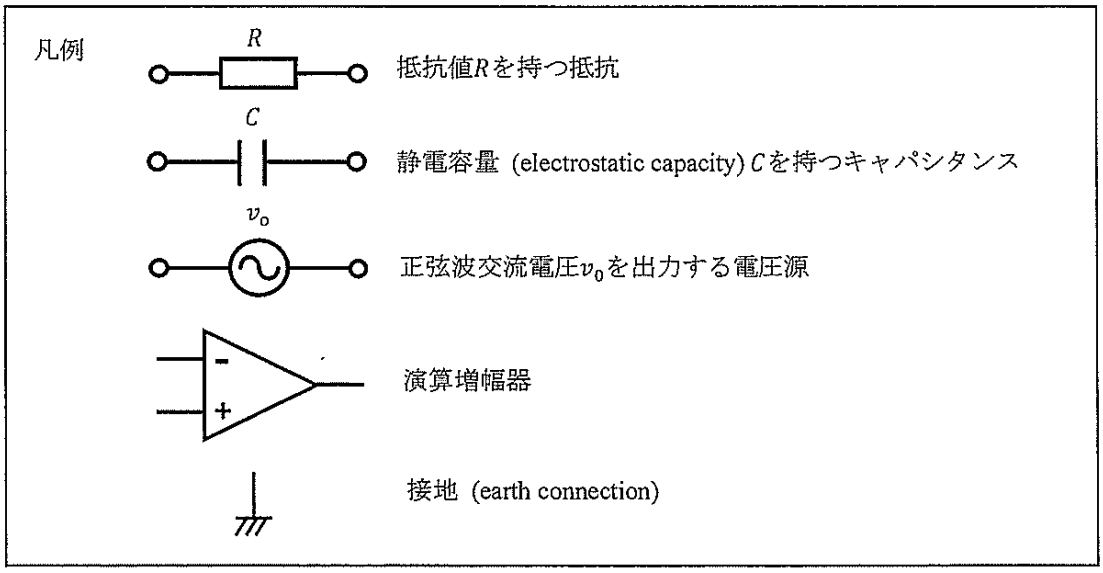
を用いる．符号長 n を 15 とするとき，情報ビット “00011010011” を送信する場合の符号語多項式 $F(x)$ を，(2-2-1) で説明した方法をもとに生成せよ．

(2-2-3) 受信側において，受信語 (received word) を表す多項式に基づいて誤りを検出する方法を一つ説明せよ．

6 【選択問題】電子回路と論理設計 （情報工学 13/17）

配点：(1-1) 10 点，(1-2) 20 点，(1-3) 30 点，(2-1) 10 点，(2-2) 30 点，(2-3) 15 点，(2-4) 10 点

(1) 演算増幅器 (operational amplifier)，抵抗 (resistance)，キャパシタンス (capacitance) から構成される回路 (circuit) について以下の各小問に答えよ．なお，図中の記号は以下の凡例に従うとする．また，正弦波交流電圧 (sinusoidal AC voltage) は複素数 (complex numbers) 表示されており，その絶対値は実効値 (effective value) を表すとし，演算増幅器の利得 (gain) 及び入力インピーダンス (input impedance) は無限大，出力インピーダンス (output impedance) は 0 であるとする．虚数単位 (imaginary unit) が必要な場合には j を用いること．



(1-1) 図 1 に示す非反転増幅器 (non-inverting amplifier) の利得 $A = V_{\mathrm{out}}/V_{\mathrm{in}}$ を求めよ．なお α は 0 または正の実数である．

(1-2) 図 2 に示す回路において，角周波数 (angular frequency) ωの正弦波交流電圧 v_{o} を印加した．回路の利得を $\beta = v_{\mathrm{fbk}}/v_{\mathrm{o}}$ としたとき，β の絶対値を最大とする角周波数 ω_{c} を R, C の式として示すとともに，$\omega = \omega_{\mathrm{c}}$ の時の入力電圧 v_o に対する出力電圧 v_{fbk} の位相差 (phase difference) を求めよ．

(1-3) 図 1 の回路に帰還回路 (feedback circuit) として図 2 の回路を追加した図 3 の回路を考える．今，α を 0 から連続的に増加させながら出力 V_{out} を観測したところ，ある α の時に発振 (oscillation) を開始した．この時の α 及び発振周波数 (oscillation frequency) を R, C の式として示せ．

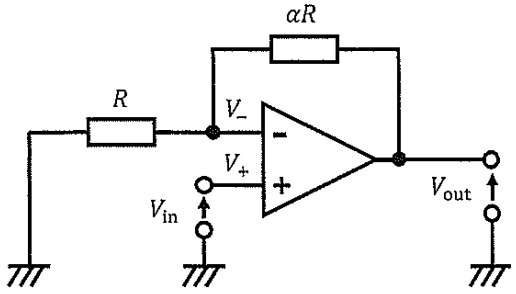


図 1

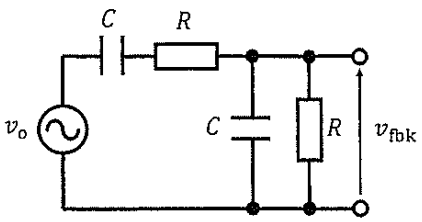


図 2

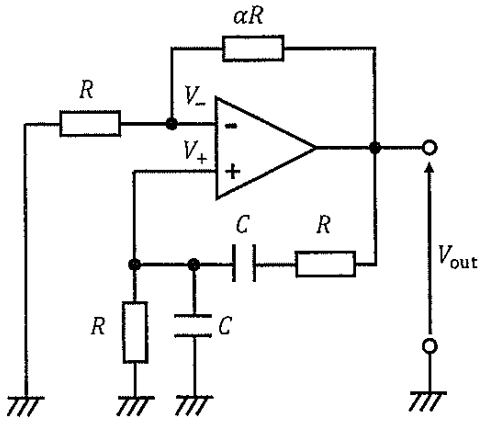


図 3

(2) 自動券売機 (automatic ticket machine) の制御回路 (control circuit) を同期式順序回路 (synchronous sequential circuit) として設計する．この自動券売機は 300 円のチケットのみを販売しており，100 円硬貨のみを受け付ける．100 円硬貨は 1 枚ずつ投入され，300 円分の硬貨が投入されると直ちにチケットが発券される．設計する制御回路は 1 ビットの入力 (input) x_0 と 1 ビットの出力 (output) y_0 を持ち，100 円硬貨が投入されたときに x_0=1，それ以外は x_0=0 が 1 クロックの間保持される．チケットを発券するときは y_0=1，それ以外は y_0=0 が出力される．また，表 1 のように A，B，C の 3 つの状態 (state) を持ち，状態変数 (state variable) k_0，k_1 を用いた状態割り当て (state assignment) が行われている．以下の各小問に答えよ．

表 1

状態	状態割当(k_0, k_1)	意味
A	(0, 0)	合計投入金額が 0 円
B	(0, 1)	合計投入金額が 100 円
C	(1, 0)	合計投入金額が 200 円

(2-1) この制御回路を Mealy 型順序回路 (Mealy sequential circuit) として設計するとき，状態遷移図 (state transition diagram) を示せ．状態遷移図には初期状態 (initial state) と入出力 (input and output) を明示すること．

(2-2) (2-1) の制御回路において，現在の状態 (current state) (k_0, k_1) に対する次状態 (next state) を (k_0^+, k_1^+) で表すとき，k_0^+，k_1^+，y_0 の最簡積和形 (最小積和形，minimal sum-of-products expression) の論理式 (logic expression) を導出せよ．

(2-3) この自動券売機にある機能を追加するため，制御回路に 1 ビットの入力 x_1 と 1 ビットの出力 y_1 を追加した．そして表 1 の状態割り当てのまま図 4 の制御回路を設計した．この制御回路の状態遷移図を示せ．状態遷移図には初期状態と入出力を明示すること．ここで，この制御回路は表 1 に示した状態からのみ遷移し，未定義の状態からの遷移は発生しないとする．また，図中の CLK はクロック信号を表す．

(2-4) 図 4 の制御回路はどのような機能が付いた自動券売機の制御回路であると考えられるか答えよ．またその理由も簡潔に述べよ．

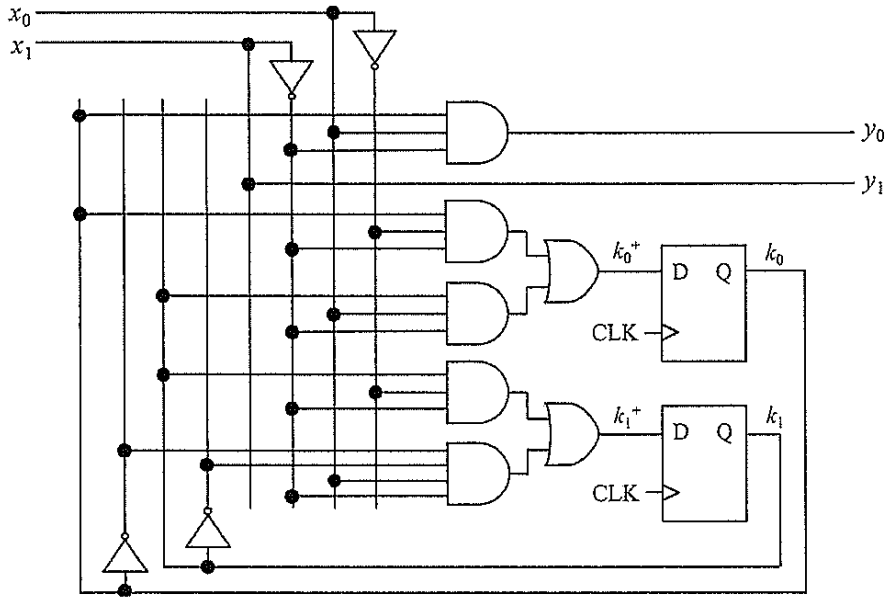


図 4

7 【選択問題】数学解析と信号処理 （情報工学 16/17）

配点: (1-1)30 点, (1-2)30 点, (2-1)20 点, (2-2)20 点, (2-3)25 点

以下の各問に答えよ.

(1) 以下の各小問に答えよ. ただし導出の過程も示すこと.

(1-1) 下記の関数のフーリエ級数 (Fourier series) を求めよ.

$$\begin{aligned} f(x) &= x^2 \\ f(x) &= f(x+2\pi) \qquad\qquad (-\pi \leq x < \pi) \end{aligned}$$

(1-2) (1-1) の結果を用いて下記の値 u を求めよ. ただし n は正の整数 (positive integer) とする.

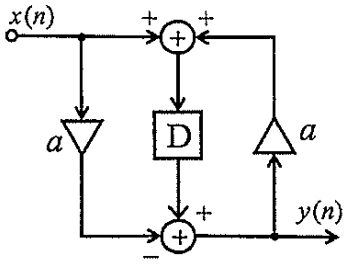
$$u = \sum_{n=1}^{\infty} \frac{1}{n^4}$$

(2) z 変換 (z-transform) された以下の伝達関数 (transfer function)

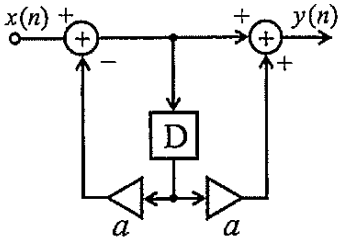
$$H(z) = \frac{1 - az}{z - a} \quad (a \text{ は実数で } -1 \leq a < 1)$$

について，以下の各小問に答えよ．ただし，導出の過程も示すこと．

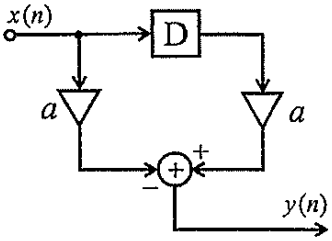
(2-1) $H(z)$ を実現するフィルタ回路 (filter circuit) を下記図 (あ)〜(う) から 1 つを選択し，さらに，入力と出力の関係を式で表せ．ただし，入力を $x(n)$，出力を $y(n)$ とする．



(あ)

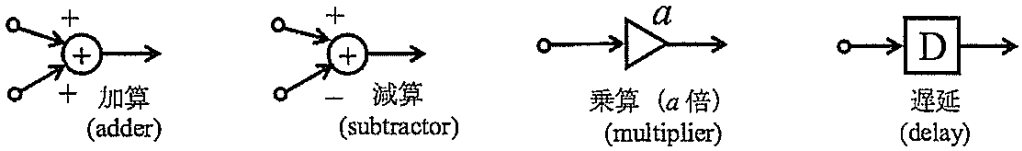


(い)



(う)

ここで，図中の回路の要素は以下の通りである．



(2-2) $a = 0, -1$ のそれぞれのとき，$H(z)$ を実現するフィルタ回路はどのような特性となるか，下記の選択肢から選んでその理由を説明せよ．

[選択肢]

加算，遅延，反転(inverter)，増幅(amplifier)， ハイパス(high-pass)，ローパス(low-pass)，そのまま通過(all-pass)

(2-3) $H(z)$ を実現するフィルタ回路について，振幅特性 (amplitude characteristics) と位相特性 (phase characteristics) を求めよ．

1 【必須問題】アルゴリズムとプログラミング　（情報工学 1/16）

配点：(1) 20 点，(2) 35 点，(3) 15 点，(4-1) 15 点，(4-2) 15 点，(5) 25 点

図 1 に示す ANSI-C 準拠である C 言語のプログラム（program）は所有している複数のくじ（lottery）のそれぞれが当選（win）しているかを調べて，当選しているくじ番号（lottery number）と等級（grade）をもれなく出力（output）するものである．くじ番号は 10000 未満の自然数（natural number）で定められており，いずれのくじ番号のくじもたかだか一つしか存在しない．所有しているくじ番号が，当選番号（winning number）と一致した場合に，その当選番号に対応する等級に当選したとする．当選番号は 10000 未満の自然数から重複なく選ばれたN個（Nは$3 \leq N \leq 100$の自然数）の数字で，等級は 1 等から 3 等まであり，1 等が 1 本，2 等が 1 本，3 等が$N-2$本である．

当選番号と等級は図 2 に示すような形式（format）のファイル win.txt で与えられ，1 行目に当選番号の総数N，2 行目以降のN行は全ての当選番号とその等級r（rは$1 \leq r \leq 3$の自然数）が書かれている．また，所有しているくじ番号は図 3 に示すような形式のファイル lots.txt で与えられ，所有しているくじ番号が 1 行目から各行に一つずつ書かれている．以下の各問に答えよ．

(1) 図 2 の win.txt，図 3 の lots.txt を与えてプログラムを実行することを考える．プログラムの 36 行目で関数 functionA が呼び出されたときに，プログラム 6〜13 行目の for 文処理において，i=1 および i=3 の時に，j に関する for 文が終了した時点で a[0]〜a[9]および b[0]〜b[9]の値が以下のようになった．8 行目の空欄(A)を配列 a に関する適切な条件式で埋めよ．

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
i=1	5308	900	7888	3500	8	4905	8698	1328	89	9003
i=3	900	3500	8	4905	5308	1328	89	7888	8698	9003

	b[0]	b[1]	b[2]	b[3]	b[4]	b[5]	b[6]	b[7]	b[8]	b[9]
i=1	3	3	2	3	3	1	3	3	3	3
i=3	3	3	3	1	3	3	3	2	3	3

(2) 36 行目で呼び出された関数 functionA の処理によって，配列 win および配列 grade はどのようになるか，説明せよ．またその処理の平均時間計算量（average case time complexity）を，変数 n を用いて，オーダ表記（order notation）で表わせ．その理由も答えよ．ただし，win.txt 内では，当選番号は無作為（at random）な順序で並んでいる．

(3) 39 行目で呼び出された関数 functionB はどのような処理をしているのか，配列 win，変数 lot，変数 n を用いて説明せよ．また，関数の戻り値（return value）についても言及すること．

(4) 4〜14 行目で定義されている関数 functionA を以下のように変更することで，36 行目で functionA を実行する時の平均時間計算量を少なくすることを考える．以下の各小問に答えよ．

```
void functionA(int a[], int b[], int t, int w){
 if(t<w){
  int i, j, x, tmp;
  i=t; j=w;
  x=a[t];

  while(1){
   while(a[i]<x) i++;
   while(a[j]>x) j--;
   if(i>=j) break;
   tmp=a[i]; a[i]=a[j]; a[j]=tmp; tmp=b[i]; b[i]=b[j]; b[j]=tmp;
   i++; j--;
  }
  functionA(a, b, (あ), (い));

  functionA(a, b, (う), (え));
 }
}
```

(4-1) 空欄(あ)〜(え)に入るものの組み合わせとして，適切なものを下の(ⅰ)〜(ⅳ)から一つ選び，答えよ．

	(あ)	(い)	(う)	(え)
(ⅰ)	i-1	t	w	j+1
(ⅱ)	t	j-1	i+1	w
(ⅲ)	t	i-1	j+1	w
(ⅳ)	j-1	t	w	i+1

(4-2) 関数 functionA の変更後の平均時間計算量を，変数 n を用いて，オーダ表記で表わせ．ただし，win.txt 内では，当選番号は無作為な順序で並んでいる．

(5) 図 1 のプログラムを，下線（ア）および下線（イ）で示した main 関数中の関数 functionA の引数 (argument) と if 文の条件式のみを変更し，1 等に当選している場合にのみ，当選しているくじ番号と等級を出力するようにする．当選番号と等級，および所有しているくじ番号は図 2 および図 3 と同じ形式で与えられる．39 行目の下線（イ）における判定を平均時間計算量$O(1)$で実現するためには，下線（ア）および下線（イ）をそれぞれどのように変更すればよいか，（ア）には適切な引数を，（イ）には適切な式をそれぞれ答えよ．

```
1  #include<stdio.h>
2  #define MAXN 100
3
4  void functionA(int a[], int b[], int t, int w){
5   int tmp, i, j;
6   for(i=t+1; i<=w; i++){
7    for(j=t; j<=w-i; j++){
8     if(      (A)      ){
9      tmp=a[j+1]; a[j+1]=a[j]; a[j]=tmp;
10     tmp=b[j+1]; b[j+1]=b[j]; b[j]=tmp;
11    }
12   }
13  }
14 }
15
16 int functionB(int a[], int x, int n){
17  int t, w, m;
18  t=0; w=n-1;
19  do{
20   m=(t+w)/2;
21   if(x<a[m]) w=m-1;
22   else t=m+1;
23  } while(t<=w);
24  if(w>=0 && x==a[w]) return w;
25  else return -1;
26 }
27
28 int main(){
29  int n, i, k;
30  FILE *fp;
31  int win[MAXN], grade[MAXN], lot;
32  fp=fopen("win.txt", "r");
33  fscanf(fp,"%d", &n);
34  for(i=0; i<n; i++) fscanf(fp, "%d %d", &win[i], &grade[i]);
35  fclose(fp);
36  functionA(win, grade, 0, n-1);
                  (ア)
37  fp=fopen("lots.txt", "r");
38  while(fscanf(fp, "%d", &lot)!=EOF){
39   if((k=functionB(win, lot, n))!=-1){
                  (イ)
40    printf("***winning number: %4d, grade: %d\n", lot, grade[k]);
41   }
42  }
43  fclose(fp);
44  return 0;
45 }
```

図 1 プログラム

```
10
5308    3
7888    2
900     3
8698    3
3500    3
8       3
4905    1
9003    3
1328    3
89      3
```

図 2 win.txt

```
9003
7888
356
28
2457
4905
43
29
3500
81
48
444
314
1028
777
```

図 3 lots.txt

2 【必須問題】計算機システムとシステムプログラム （情報工学 4/16）

配点： (1-1) 10 点，(1-2) 12 点，(1-3-1) 12 点，(1-3-2) 8 点，(1-4-1) 6 点，(1-4-2) 12 点
(2-1) 20 点，(2-2) 10 点，(2-3-1) 15 点，(2-3-2) 20 点

(1) 計算機 (computer) における整数 (integer) の表現 (representation) と算術演算 (arithmetic operation) に関する以下の各小問に答えよ．解答は，全て解答用紙の太線枠内に書くこと．

(1-1) 10 進数 (decimal number) の -15 を 8 [ビット] の符号絶対値表現 (signed magnitude representation) および 1 の補数表現 (one's complement representation) で表した場合のビット列 (bit string) を示せ．

(1-2) 8 [ビット] の 2 の補数表現 (two's complement representation) で表すことのできる正 (positive) の最大値 (maximum value) および負 (negative) の最小値 (minimum value) のビット列を示せ．また，それらを 10 進数の数値で示せ．

(1-3) 計算機で符号付き整数 (signed integer) の加算を行なう時，多くの場合，2 の補数表現が用いられる．2 の補数表現を用いた加算に関する以下の (1-3-1), (1-3-2) に答えよ．

(1-3-1) $43+(-5)$ の加算を 8 [ビット] の 2 の補数表現を用いて行う過程を示せ．最上位ビットからの桁上げ (carry) 出力の扱いも記すこと．

(1-3-2) 計算機上で符号付き整数の加算に 2 の補数表現を用いる利点を一つ示せ．

(1-4) 二つの整数 A, B を加算し，整数 S を得る演算を考える．A, B, S は同じビット長とする．A, B, S のビット列の最上位ビットをそれぞれ a, b, s とし，加算器の最上位ビットからの桁上げ出力を c とする．このとき，以下の (1-4-1), (1-4-2) に答えよ．

(1-4-1) A, B, S が符号無し整数の場合，オーバーフロー (overflow) の発生を判定する方法を，変数 a, b, s, c を用いて示せ．なお，必要の無い変数は使わなくて良い．

(1-4-2) A, B, S が 2 の補数表現で表された符号付き整数の場合，オーバーフローの発生を判定する方法を，変数 a, b, s, c を用いて示せ．なお，必要の無い変数は使わなくて良い．

(2) 単一プロセッサ (single processor) のマルチタスク (multitask) 環境における排他制御 (exclusive control) に関する以下の各小問に答えよ．解答は，全て解答用紙の太線枠内に書くこと．

(2-1) 排他制御を実現する際には，デッドロック状態 (deadlock) や飢餓状態 (starvation) になることを避ける必要がある．それぞれの状態について説明せよ．

(2-2) 以下に示すプロセス 1 とプロセス 2 は，右図に示すようなスタック (stack) の操作を実現する．プロセス 1 はスタックへのデータのプッシュ (push) を行い，プロセス 2 はスタックからのデータのポップ (pop) を行う．スタックには事前にデータ $d_1, d_2, \cdots, d_n$ が格納されている．top はスタックの先頭 (stack top) のアドレスを表す変数であり，二つのプロセスが共有している．top は事前に x に設定されている．stack() は引数 (argument) で指定されたアドレスのデータを表す．push_item と pop_item はデータを表す変数であり，push_item には事前に値が設定されている．スタックはプロセス 1 とプロセス 2 のみが共有しており，スタックの操作を実行するプロセスは他に存在しない．

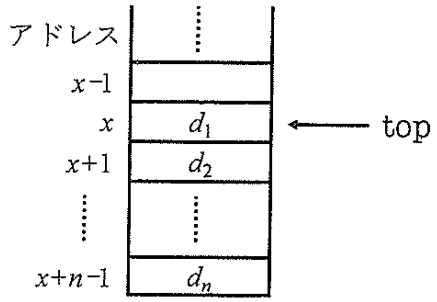


図: スタック

```
(ア)  top = top - 1;                    (ウ)  pop_item = stack(top);
(イ)  stack(top) = push_item;           (エ)  top = top + 1;
      プロセス 1                              プロセス 2
```

プロセス 1 とプロセス 2 が並行に動作した時に，二つのプロセスの命令実行とプロセス切り換えのタイミングによっては，プロセス 1 が実行するプッシュ操作とプロセス 2 が実行するポップ操作のいずれか，あるいは両方が正しく行われないことがある．そのような (ア)(イ)(ウ)(エ) の実行順序 (execution order) のうち，(ア) から開始されるものを一つ示せ．

(2-3) セマフォ (semaphore) を用いて，(2-2) におけるプロセス 1 とプロセス 2 が並行に動作した時においても，プロセス 1 が実行するプッシュ操作とプロセス 2 が実行するポップ操作が正しく行われるようにすることを考える．ここで，セマフォは 0 または 1 の値を取るものとする．P() はセマフォを引数とし，セマフォが 0 であれば P() を実行したプロセスを休止させ，1 であればセマフォを 1 から 0 にする不可分命令 (atomic operation, atomic transaction) である．V() はセマフォを引数とし，同じセマフォを引数として実行された P() によって休止されたプロセスがあれば起動し，なければセマフォを 0 から 1 にする不可分命令である．このとき，以下の (2-3-1), (2-3-2) に答えよ．

(2-3-1) プロセス 1 とプロセス 2 が並行に動作した時においても，プロセス 1 が実行するプッシュ操作とプロセス 2 が実行するポップ操作が正しく行われるように，P(), V(), 及びセマフォ a を用いて，プロセス 1 とプロセス 2 に適切な命令を追加したものを示せ．また，a の適切な初期値 (initial value) も示せ．

(2-3-2) 以下に示すプロセス 3 とプロセス 4 は，変数 m を共有している．プロセス 3 は m の値をインクリメント (increment) し続けるプロセスであり，プロセス 4 は m の値を表示し続けるプロセスである．m は事前に 1 に初期化されており，m を共有するプロセスは他に存在しない．

```
while (true) {                  while (true) {
 m = m + 1;                      print(m);
}                               }
プロセス 3                      プロセス 4
```

プロセス 3 とプロセス 4 が並行に動作した時に，プロセス 4 における m の値の表示が 1, 2, 3, … となるようにしたい．P(), V(), 及び二つのセマフォ b, c を用いて，プロセス 3 とプロセス 4 に適切な命令を追加したものを示せ．また，b, c の適切な初期値も示せ．

3 【選択問題】離散構造 （情報工学 6/16）

配点:(1-1-1) 10 点，(1-1-2) 20 点，(1-2-1) 10 点，(1-2-2) 10 点，(1-2-3) 10 点，(2-1-1) 15 点，(2-1-2) 10 点，(2-2-1) 10 点，(2-2-2) 10 点，(2-2-3) 20 点

(1) 述語論理（predicate logic）に関する以下の各小問に答えよ．ただし，論理式（logic formula）の記述には以下の記号を用いる．$\forall$, $\exists$ はそれぞれ全称作用素（universal quantifier），存在作用素（existential quantifier）であり，$\Leftrightarrow$, $\rightarrow$, $\wedge$, $\vee$, $\neg$ はそれぞれ等価（equivalence），含意（implication），論理積（conjunction，and），論理和（disjunction，or），否定（negation，not）を表す論理演算子である．また，論理式集合（a set of logical formulae）Γ から論理式 P が証明可能（provable）であるとき，$\Gamma \vdash P$ と書くこととする．特に断りのない限り，x を変数（variable）記号，p, q を述語（predicate）記号として用いる．

論理式 $E = (\forall x(p(x) \rightarrow q(x)) \wedge \exists x\ p(x)) \rightarrow \exists x\ q(x)$ が恒真（valid）であることを，2 通りの方法で証明する．以下の各小問に答えよ．

(1-1) 公理系（axioms）に基づいた証明を考える．ここでは，次の述語論理の公理系 S を用いる．

[公理系 S] P, Q, R で任意の論理式を表す．また，$P(x)$ は論理式 P 内に出現する可能性のある変数 x を明記した記法であり，$P(t)$ はその変数 x に項（term）t を代入（substitute）して得られる論理式を表す．S の公理（axiom）は次の $A1$〜$A4$ の 4 つである．

$A1$： $P \rightarrow (Q \rightarrow P)$

$A2$： $(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R))$

$A3$： $(\neg P \rightarrow \neg Q) \rightarrow ((\neg P \rightarrow Q) \rightarrow P)$

$A4$： $\forall x P(x) \rightarrow P(t)$　　ただし，t は $P(x)$ において変数 x に代入可能な項．

また，S の推論規則（inference rule）は次の $B1$，$B2$ である．

$B1$： P と $P \rightarrow Q$ から Q を導く．

$B2$： $P \rightarrow Q$ から $P \rightarrow \forall x Q$ を導く．ただし，P は自由変数 x を含まない．

本小問の証明では，以下の定理 $T1$〜$T3$ と演繹定理（deduction theorem）$D1$ は証明無しに用いてもよい．

$T1$： $(P \wedge Q) \vdash P$

$T2$： $(P \wedge Q) \vdash Q$

$T3$： $\vdash (P \rightarrow Q) \rightarrow (\exists x P \rightarrow \exists x Q)$

$D1$： P が閉論理式（closed formula）のとき，$\Gamma, P \vdash Q$ ならば $\Gamma \vdash (P \rightarrow Q)$

以下の (1-1-1) および (1-1-2) に答えよ．

(1-1-1) 公理系 S のもとで，次の定理 $T4$ を証明せよ．

$T4$： $\forall x P(x) \vdash P(x)$

(1-1-2) 公理系 S のもとで，論理式 E を証明せよ．ただし，定理 $T4$ も証明無しに用いてもよい．

(1-2) 論理式 E が恒真であることを示すため，まず，E の否定をスコーレム化（Skolemization）し，次いで，導出原理（resolution principle）を適用し，その式が充足不能であることを示す．以下の (1-2-1)〜(1-2-3) に答えよ．

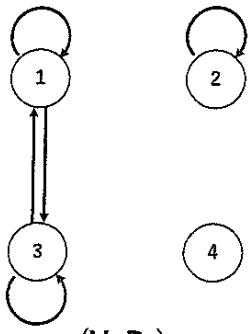
(1-2-1) $\neg E$ の冠頭標準形（prenex normal form）を示せ．冠頭標準形は，すべての限定作用素（quantifier）が先頭にある閉論理式である．ただし，閉論理式は連言標準形（和積標準形：conjunctive normal form）で示すこと．

(1-2-2) 上記の (1-2-1) で得た論理式のスコーレム連言標準形（Skolem conjunctive normal form）$\neg E'$ を求めよ．

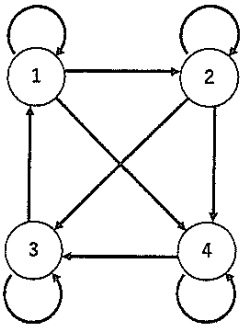
(1-2-3) 導出原理を用いて $\neg E'$ が充足不能であることを示せ．

(2) 空でない有限集合（finite set）上の二項関係（binary relation）に関し，以下の各小問に答えよ．

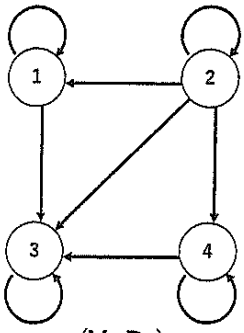
(2-1) 有向グラフ（directed graph）(V,R) $(R \subseteq V \times V)$ において，$R = \{(s,t) \mid s,t \in V; s$ から t への有向辺（directed edge）が存在 $\}$ であるとき，R は有限集合 V 上の二項関係を表す．以下の有向グラフで表される $V = \{1,2,3,4\}$ 上の二項関係 R_1，R_2 および R_3 について，以下の (2-1-1) および (2-1-2) に答えよ．



(V, R_1)



(V, R_2)



(V, R_3)

(2-1-1) R_1，R_2 および R_3 はそれぞれ

(a) 反射律（reflexive law）

(b) 対称律（symmetric law）

(c) 反対称律（antisymmetric law）

(d) 推移律（transitive law）

のうち，どの性質を満たすか．それぞれについて，記号 (a)~(d) の中から満たすものをすべて答えよ．

(2-1-2) R_1，R_2 および R_3 のうち，順序関係（partial order relation）であるものをすべて答えよ．

(2-2) 順序関係が定義された有限集合を順序集合（partially ordered set）とよび，順序関係 $\preceq$ が定義された有限集合 V を $(V,\preceq)$ で表す．任意の順序集合 $(V,\preceq)$ と 集合 A $(A \subseteq V)$ に対し，以下を定義する．

$upper(A) = \{t \mid \forall s \in A; s \preceq t\}$ を A の上界（upper bound）とよぶ．

$upper(A)$ に含まれる A の要素が存在すれば，それを A の最大元（maximum）とよぶ．

$lower(A) = \{s \mid \forall t \in A; s \preceq t\}$ を A の下界（lower bound）とよぶ．

$lower(A)$ に含まれる A の要素が存在すれば，それを A の最小元（minimum）とよぶ．

ある有限集合 X のべき集合（power set）$\mathcal{P}(X)$，およびその上での二項関係 $\preceq_{\mathcal{P}(X)} = \{(s,t) \mid s,t \in \mathcal{P}(X);\ s \subseteq t\}$ について，以下の (2-2-1)~(2-2-3) に答えよ．なお，$\cap$ および $\cup$ は，それぞれ二つの集合の共通集合（intersection）および和集合（union）を求める演算子である．

(2-2-1) $(\mathcal{P}(X), \preceq_{\mathcal{P}(X)})$ が順序集合であることを示せ．

(2-2-2) 任意の $s,t \in \mathcal{P}(X)$ に対し，$w \in upper(\{s,t\})$ であり，かつ要素数が最小の w を，s および t を用いて表せ．また，$z \in lower(\{s,t\})$ であり，かつ要素数が最大の z を，s および t を用いて表せ．

(2-2-3) 任意の $u \in \mathcal{P}(X)$ に対し，$c(u)$ は $\mathcal{P}(X)$ の要素のうち，$u \cup c(u)$ が $\mathcal{P}(X)$ の最大元，$u \cap c(u)$ が $\mathcal{P}(X)$ の最小元であるものとする．このとき，任意の $s,t \in \mathcal{P}(X)$ に対し，$c(s \cup t) = c(s) \cap c(t)$ であることを示せ．

4 【選択問題】計算理論　（情報工学 8/16）

配点：(1-1) 5 点, (1-2) 5 点, (1-3) 35 点, (1-4) 20 点, (2-1) 35 点, (2-2) 25 点

(1) アルファベット (alphabet) $\Sigma_{ab} = \{a, b\}$ で構成される回文 (palindrome) を考える．回文とは前から読んでも後ろから読んでも同じ文字列 (string) のことである．例えば，b, aa, aba, $baab$ は回文だが，ab, abb, $baaa$ は回文ではない．なお，文字列 w の長さを $|w|$ と表す．例えば，$|b| = 1$，$|abb| = 3$ である．また，空列 (empty string) ε ($|\varepsilon| = 0$) は回文である．

回文を受理するオートマトン (automaton) に関する以下の各小問に答えよ．

(1-1) 決定性有限オートマトン (deterministic finite automaton) は $(Q, \Sigma, \delta, q_0, F)$ で表される．Q は状態 (state) の有限集合，Σ は入力アルファベット (input alphabet)，δ は遷移関数 (transition function)，q_0 は開始状態 (start state)，F は最終状態 (final state) の集合である．言語 (language) $\{p \in \Sigma_{ab}^* \mid |p| = 2,\ p \text{ は回文}\}$ を受理する決定性有限オートマトンの状態遷移図 (state transition diagram) を書け．なお，下図に示すように開始状態には太い矢印 (thick arrow) を付与し，最終状態は二重丸 (double circle) で表現すること．下図では，開始状態は q_0，最終状態は q_2 である．

(1-2) 言語 $\{p \in \Sigma_{ab}^* \mid |p| = 3,\ p \text{ は回文}\}$ を受理する決定性有限オートマトンの状態遷移図を書け．なお，下図に示すように開始状態には太い矢印を付与し，最終状態は二重丸で表現すること．

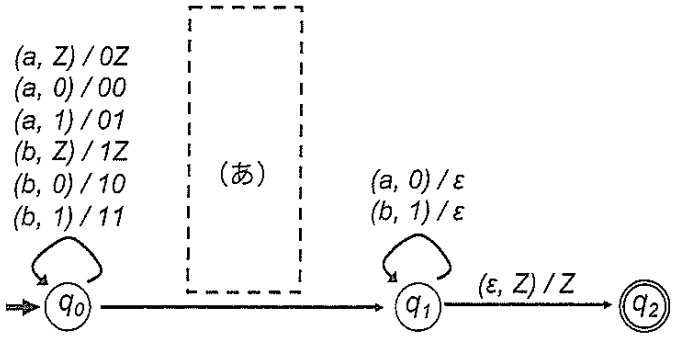
(1-3) 言語 $\{p \in \Sigma_{ab}^* \mid |p| \geq 0,\ p \text{ は回文}\}$ を受理する決定性有限オートマトンは存在しないこと，すなわち，この言語は正則言語 (regular language) ではないことを背理法 (proof by contradiction) により証明せよ．なお，以下の正則言語に対する反復補題 (pumping lemma) を用いること．

正則言語に対する反復補題

L を正則言語とすると，次の条件を満たす正の整数 n が存在する：$v \in L$，$|v| \geq n$ なら v は $v = xyz$ ($|xy| \leq n, |y| \geq 1$) と分解でき，任意の $k \geq 0$ に対して $xy^kz \in L$ である．

(1-4) プッシュダウンオートマトン (pushdown automaton) は $(Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ で表される．Q は状態の有限集合，Σ は入力アルファベット，Γ はスタックアルファベット (stack alphabet)，δ は遷移関数，q_0 は開始状態，Z はスタックの開始記号，F は最終状態の集合である．最終状態による受理 (acceptance by final state) を行うプッシュダウンオートマトンを利用すれば，言語 $\{p \in \Sigma_{ab}^* \mid |p| \geq 0,\ p \text{ は回文}\}$ を受理できる．以下はそのような非決定性プッシュダウンオートマトンの状態遷移図である．状態遷移図の「(あ)」に必要な動作を，$(r, s)/t$ の形式ですべて答えよ．

ただし，$(r, s)/t$ は，入力から r を読み出すときにスタックの先頭にある記号 s を取り去り，t をスタックに押し込むことを意味する．t が複数の記号の場合は右側の記号から順にスタックに押し込む．例えば，下図の $(a, Z)/0Z$ は，a を読み出すときにスタックの先頭から Z を取り去り，スタックに Z と 0 をこの順で押し込むことを意味する．t が ε の場合は，スタックに記号を押し込まない．r が ε の場合は，入力から記号を読み出さないで遷移を行う．スタックアルファベットは $\Gamma = \{Z, 0, 1\}$ とする．



(2) 文脈自由文法 (context-free grammar) は一般に $G = (V, T, P, S)$ で定められる．ここで V は変数 (variable; 非終端記号 nonterminal symbol) の集合，T は終端記号 (terminal symbol) の集合，P は生成規則 (production rule) の集合，S は出発記号 (start symbol) であり，$S \in V$ である．文脈自由文法 $G_1 = (V_1, T_1, P_1, S_1)$ を $V_1 = \{A\}$，$T_1 = \{\mathtt{a}, \mathtt{b}\}$，$P_1 = \{A \to \mathtt{a}A\mathtt{b}A,\ A \to \mathtt{b}A\mathtt{a}A,\ A \to \varepsilon\}$，$S_1 = A$ と定める．L_1 を G_1 により生成される言語 (language) とする．L_1 は同じ個数 (the same number) の a と b を含む文字列 (string) すべてからなる言語であることを証明したい．以下の各小問に答えよ．

【注】文字列とは終端記号の列である．任意の文字列 w に対し，記法 $|w|$ は w の中の終端記号すべての個数を，$|w|_{\mathtt{a}}$ は w の中の終端記号 a の個数を，$|w|_{\mathtt{b}}$ は w の中の終端記号 b の個数を，それぞれ表す．例えば $w = \mathtt{aaba}$ に対して，$|w| = 4$，$|w|_{\mathtt{a}} = 3$，$|w|_{\mathtt{b}} = 1$ である．ε は空列 (empty string) を表す．記法 $\alpha \Rightarrow \beta$ は文法 G_1 の生成規則を 1 回適用することで文形式 (sentential form) α から文形式 β が得られることを表す．

(2-1) L_1 に含まれている任意の文字列 w に対して w は同じ個数の a と b を含んでいること，すなわち必要条件 $\forall w \in T_1^*[w \in L_1 \text{ならば } |w|_{\mathtt{a}} = |w|_{\mathtt{b}}]$ を証明したい．空欄（ア）を埋めよ．

証明: 生成規則を適用する回数 k に関する数学的帰納法 (mathematical induction) で証明する．

- 基底段階 (base case): $k = 1$ の場合．1 回の生成規則の適用で文字列を得る導出 (derivation) は $A \Rightarrow \varepsilon$ のみである．$w = \varepsilon$ とすれば $|\varepsilon|_{\mathtt{a}} = |\varepsilon|_{\mathtt{b}} = 0$ なので，$|w|_{\mathtt{a}} = |w|_{\mathtt{b}}$ が成立する．
- 帰納段階 (inductive step): $k > 1$ の場合．$k-1$ 回以下の生成規則の適用で得られる文字列 v はいずれも $|v|_{\mathtt{a}} = |v|_{\mathtt{b}}$ が成立すると仮定する．すると以下の理由により $|w|_{\mathtt{a}} = |w|_{\mathtt{b}}$ が成立する．

理由：［　　（ア）　　］

(証明終)

(2-2) 同じ個数の a と b を含んでいる任意の文字列 w に対して w は L_1 に含まれていること，すなわち十分条件 $\forall w \in T_1^*[|w|_{\mathtt{a}} = |w|_{\mathtt{b}} \text{ならば } w \in L_1]$ を証明したい．空欄（イ）～（エ）を埋めよ．

証明: 文字列 w の長さ $|w|$ に関する帰納法で証明する．$|w|$ は 0 以上の偶数 (even number) なのは明らかである．

- 基底段階: $|w| = 0$ の場合．以下の理由により $w \in L_1$ が成立する．

理由：［　　（イ）　　］

- 帰納段階: $|w| > 0$ かつ偶数の場合．長さが偶数で $|w|$ 未満の文字列 v が $|v|_{\mathtt{a}} = |v|_{\mathtt{b}}$ を満たすならば，$v \in L_1$ であると仮定する．w の最初（左端）の終端記号により以下の場合分けをする．
  - a の場合．$|w|_{\mathtt{a}} = |w|_{\mathtt{b}}$ のとき，長さが $|w|$ 未満のある文字列 $v_1, v_2 \in L_1$ が存在して $w = \mathtt{a}v_1\mathtt{b}v_2$ と書ける事が示せる（証明略）．従って，導出 $A \Rightarrow$［　（ウ）　］$\Rightarrow \cdots \Rightarrow \mathtt{a}v_1\mathtt{b}v_2 = w$ により $w \in L_1$ が成立する．
  - b の場合．$|w|_{\mathtt{a}} = |w|_{\mathtt{b}}$ のとき，長さが $|w|$ 未満のある文字列 $v_1, v_2 \in L_1$ が存在して $w = \mathtt{b}v_1\mathtt{a}v_2$ と書ける事が示せる（証明略）．従って，導出 $A \Rightarrow$［　（エ）　］$\Rightarrow \cdots \Rightarrow \mathtt{b}v_1\mathtt{a}v_2 = w$ により $w \in L_1$ が成立する．

(証明終)

配点: (1-1-1) 6 点，(1-1-2) 6 点，(1-2-1) 15 点，(1-2-2) 6 点，(1-2-3) 12 点，(1-3) 15 点
(2-1) 20 点，(2-2) 15 点，(2-3-1) 10 点，(2-3-2) 10 点，(2-3-3) 10 点

(1) 次のような記憶がない情報源 (memoryless information source) S を考える．情報源 S の記号数 s は 2 以上とし，情報源記号は $a_1, a_2, \ldots, a_s$, 各 i $(1 \leq i \leq s)$ について記号 a_i の生起確率 (occurence probability) は 2^{-m_i} で表される．ここで，$m_1, m_2, \ldots, m_s$ は，$1 \leq m_1 \leq m_2 \leq \cdots \leq m_s$ を満たす整数である．このとき，各 i $(1 \leq i \leq s)$ について記号 a_i に対する符号語長 (code length) が m_i である瞬時に復号可能な符号化 (instantly decodable source coding) が存在することについて以下の各小問に答えよ．

(1-1) 特別な場合として，$s = 6$，$m_1 = 1$，$m_2 = 2$，$m_3 = m_4 = m_5 = m_6 = 4$ である情報源 S_0 を考える．

(1-1-1) この場合に題意を満たす符号化 (下線部) の復号木 (decoding tree) を一つ示せ．通信路記号は 0, 1 とせよ．なお，復号木は符号の木 (code tree) とも呼ばれている．

(1-1-2) 上で復号木を示した符号化の平均符号語長，および情報源 S_0 の 2 を底とするエントロピー (entropy) を求めよ．

(1-2) 一般の場合に題意を満たす符号化 (下線部) が存在することを，記号数 s に関する数学的帰納法 (mathematical induction) により証明したい．

(1-2-1) 証明の準備として，以下の文章の各空欄を埋めることにより，常に，$m_{s-1} = m_s$ であることを示せ．なお，空欄 （い） ～ （え） には，それぞれ「偶数」または「奇数」のいずれかの語を埋めよ．

$2^{m_s} \sum_{i=1}^{s} 2^{-m_i}$ = （あ） である．$m_s \geq 1$ なので，$2^{m_s} \sum_{i=1}^{s} 2^{-m_i}$ は （い） である．$m_{s-1} \neq m_s$ と仮定すれば，$2^{m_s} \sum_{i=1}^{s-1} 2^{-m_i}$ は （う） であり，これらの差 $2^{m_s} \sum_{i=1}^{s} 2^{-m_i} - 2^{m_s} \sum_{i=1}^{s-1} 2^{-m_i}$ は （え） となるが，差の値

$$2^{m_s} \sum_{i=1}^{s} 2^{-m_i} - 2^{m_s} \sum_{i=1}^{s-1} 2^{-m_i} = \boxed{\text{（お）}}$$

であり矛盾する．

(1-2-2) 証明の基底段階 (base case) を示せ．すなわち，$s = 2$ のときに題意を満たす符号化が存在することを示せ．

(1-2-3) 証明の帰納段階 (inductive step) を示せ．

(1-3) 以下の文章の各空欄を埋めよ．なお，空欄 （う） には人名 (姓 family name) を埋めよ．また，空欄 （お） には「がある」または「はない」のどちらかを埋めよ．

一般の場合に，情報源 S の 2 を底とするエントロピーは （あ） である．小問 (1-2) で存在を示した符号化の平均符号語長は （い） であり，（う） の （え） 定理から，S の n 次拡大 (n-th extension) に対する符号化を考えた場合に，一記号あたりの平均符号語長がより小さい符号化が存在する可能性 （お）．

(2)　図 1 に概要を示すディスタンスベクトル型ルーティングアルゴリズム (distance vector routing algorithm) を用いて最小コスト経路 (minimum cost route) を決めることを考える．これについて以下の各小問に答えよ．

記号

$\mathcal{N}$　全ノードの集合．

$c(i,j)$　ノード i からノード j へのリンクのコスト．$c(i,j) \geq 0$，$c(i,i) = 0$ とする．また，ノード i からノード j へリンクがない場合は $c(i,j) = \infty$ とする．

$d_i(j)$　ノード i において，その時点でノード i が知るノード i からノード j までの最小コスト経路のコストを保持する変数．

D_i　ノード i の持つディスタンスベクトル．$\mathcal{N} = \{x, y, z\}$ の場合，D_i は以下の通りとする．

$D_i = [d_i(x)$ の値$, d_i(y)$ の値$, d_i(z)$ の値$]$

初期状態

- 各ノード i において $c(i,j)$ は既知である $(\forall j \in \mathcal{N})$.
- 各ノード i において $d_i(j) = c(i,j)$ である $(\forall j \in \mathcal{N})$.

動作

全ノードは，各ステップで以下の動作 I，動作 II のそれぞれを同期的 (synchronously) に実行し，そのステップを繰り返し実行する．

動作 I.　各ノード i は，隣接 (adjacent) ノードにディスタンスベクトル D_i を送信し，自身に送信されたディスタンスベクトルを受信する．

動作 II.　各ノード i は，受信した全てのディスタンスベクトルにもとづいて，自身のディスタンスベクトル D_i を更新する．

経路の収束 (convergence)

前ステップと現ステップの間で D_i $(\forall i \in \mathcal{N})$ の値に変化がなくなることを経路が収束すると定義する．最小コスト経路は，収束したときの各ノードが持つ情報から導出できる．

図 1: ディスタンスベクトル型ルーティングアルゴリズムの概要

(2-1)　図 1 の動作 II の下線部における D_i の更新方法を数式で記述せよ．記述に必要な記号を新たに定義しても構わないが，定義した記号の説明も解答に含めよ．

(2-2)　図 2 に示すネットワークを考える．リンクのコストは，それぞれ，$c(x,y) = c(y,x) = 2$，$c(x,z) = c(z,x) = 8$，$c(y,z) = c(z,y) = 3$ である．また，動作 I では送信したディスタンスベクトルは紛失なく届くものとする．このとき，図 1 の初期状態の直後の更新をステップ 1 として，経路が収束するまでの各ステップについて，動作 I においてノード x が受信したディスタンスベクトル D_y と D_z，および，動作 II において更新した後のノード x のディスタンスベクトル D_x を記述せよ．ただし，D_i $(i \in \{x,y,z\})$ は，$D_i = [d_i(x)$ の値$, d_i(y)$ の値$, d_i(z)$ の値$]$ の表記法で記述せよ．

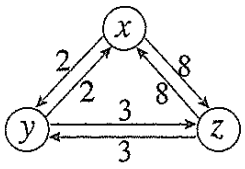


図 2: ネットワーク

(2-3)　ディスタンスベクトル型ルーティングアルゴリズムでは，リンクが切断した際に経路の収束までに非常に長い時間を要するという問題 (count-to-infinity 問題) がある．この問題について以下の (2-3-1)，(2-3-2)，(2-3-3) に答えよ．ただし，ここでは，リンクが切断された場合，ディスタンスベクトルは切断されたリンクを介して到達しないものとする．

(2-3-1)　図 3 のネットワークを考える．リンクのコストは，$c(x,y) = c(y,x) = c(y,z) = c(z,y) = 1$ である．このネットワーク上で経路が収束している状態でのノード y のディスタンスベクトル D_y は，以下の通りである．

$$D_y = [d_y(x) \text{ の値},\ d_y(y) \text{ の値},\ d_y(z) \text{ の値}] = [1, 0, 1]$$

経路が収束した状態でノード x とノード y 間のリンクが切断した $(c(x,y) = c(y,x) = \infty)$ とする．リンク切断が発生した直後の更新をステップ 1 として，図 1 の動作にしたがって経路を更新したときのステップ 1 からステップ 3 までの各ステップ終了時の $d_y(x)$ の値を示せ．

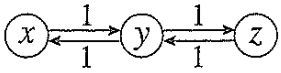


図 3: ネットワーク

(2-3-2)　(2-3-1) の $d_y(x)$ をふまえて，ディスタンスベクトル型ルーティングアルゴリズムで，count-to-infinity 問題が発生する理由を説明せよ．

(2-3-3)　リンクステート型ルーティングアルゴリズム (link state routing algorithm) では，count-to-infinity 問題が発生しない理由を説明せよ．

6 【選択問題】電子回路と論理設計 （情報工学 13/16）

配点：(1-1) 20 点, (1-2) 25 点, (2-1) 15 点, (2-2) 20 点, (3-1) 15 点, (3-2) 12 点, (3-3) 18 点

(1) 図 1 に示す MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) を使った 2 入力 (input)1 出力 (output) の CMOS 型回路 (CMOS circuit) では以下を満たすとする．

- $V_{DD}(>0)$ は電源電圧 (source voltage) である．
- C_1，C_2 は負荷容量 (load capacitance) である．
- x，y は入力，z は出力である．
- x，y は V_{DD} もしくは 0 の値しか取らない．
- nMOS はゲート・ソース間電圧 (gate-source voltage)$V_{GS} \geq 0.5V_{DD}$ のときオンになり，$V_{GS} < 0.5V_{DD}$ のときオフとなる．
- pMOS はゲート・ソース間電圧 $V_{GS} \geq 0.5V_{DD}$ のときオフになり，$V_{GS} < 0.5V_{DD}$ のときオンとなる．
- すべての MOSFET のドレイン (drain)・ソース (source) 間の抵抗値 (resistance) はオンにすると瞬時に R_{MOS} となり，オフにすると瞬時に ∞ になる．
- 配線 (wire) の負荷容量，抵抗値は 0 とする．

以下の各小問に答えよ．

(1-1) 図 1 の CMOS 型回路は，ある論理ゲート (logical gate) を実現したものである．図 1 の CMOS 型回路に相当する論理ゲートを図 2 の 1〜5 から 1 つ選択せよ．

(1-2) 図 1 の CMOS 型回路において，x が 0 で y が V_{DD} になって十分に長い時間が経った後に，x が V_{DD} に変化してから z における出力電圧 (output voltage) が $0.5V_{DD}$ になるまでの時間 T を V_{DD}，R_{MOS}，C_1，C_2 を用いて示せ．ただし，V_{DD}，R_{MOS}，C_1，C_2 の中で不要な変数は用いなくてよい．

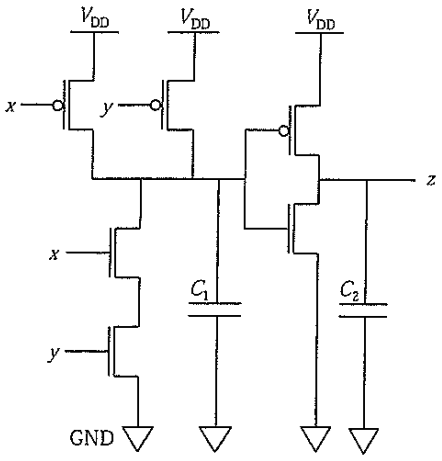


図 1: 2 入力 1 出力 CMOS 型回路

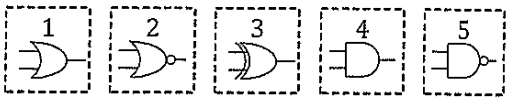


図 2: 論理ゲート

(2) $K(0 \leqq K \leqq 3)$ ビットの左論理シフト (logical left shift) を行う組み合わせ回路 (combinational logic circuit) を設計する．図 3 に設計する回路の回路図を示す．この回路は以下を満たすとする．

- $X = (x_3, x_2, x_1, x_0)$ は 4bit 符号なし 2 進数 (unsigned binary number) であり，入力である．
- $Y = (y_3, y_2, y_1, y_0)$ は 4bit 符号なし 2 進数であり，出力である．
- x_0，y_0 を最下位ビット (LSB: Least Significant Bit) とする．
- シフト量 K は p_3，p_2，p_1，p_0 の 4 個のスイッチによって選択可能であり，p_K をオン $(= 1)$ にすると K ビットシフトする．ただし，複数のスイッチがオンになった場合は，オンになった p_K のうち最大の K をシフト量とする．
- セレクタ 1 は，選択信号 s_0 が 0 の時は A_0 を，1 の時は B_0 を出力する回路である．
- セレクタ 2 は，選択信号 s_1 が 0 の時は A_1 を，1 の時は B_1 を出力する回路である．
- 左 1 ビット論理シフタと左 2 ビット論理シフタでは，シフトによって空いたビットに 0 を格納する．
- 論理回路 1 は K ビットシフトを行うために，スイッチからの入力 p_3，p_2，p_1，p_0 をセレクタ 1，2 への選択信号 s_1，s_0 に変換する回路である．

以下の各小問に答えよ．

(2-1) $X = (0, 0, 0, 1)$ に対して，スイッチを $(p_3, p_2, p_1, p_0) = (0, 0, 1, 1)$ とすると，$Y = (0, 0, 1, 0)$ となる．$X = (0, 0, 0, 1)$ に対して，スイッチを $(p_3, p_2, p_1, p_0) = (0, 1, 1, 0)$ とした時の Y を示せ．

(2-2) 図 3 の論理回路 1 の出力 s_1，s_0 を，入力 p_3，p_2，p_1，p_0 を用いた最簡積和形 (最小積和形, minimal sum-of-products expression) の論理式 (logical expression) で示せ．

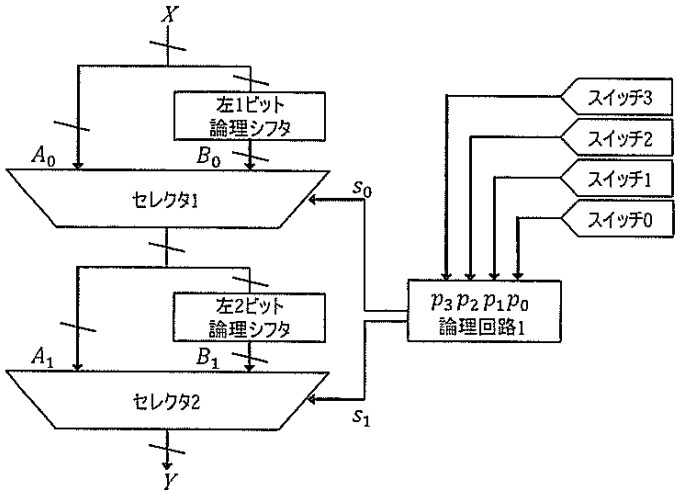


図 3: 回路図

(3) 0〜3 のいずれかの値を保持する 4 進のアップダウンカウンタ (up/down counter) を設計する．作成するカウンタは以下を満たすとする．

- Moore 型の順序機械 (Moore machine) とする．
- 2 個のエッジトリガ型の D フリップフロップ (edge-triggered D flip-flop) を用いる．
- 各フリップフロップの入力をそれぞれ D_1，D_0，出力を Q_1，Q_0 とする．
- この順序機械の状態 (state) は，カウンタ値が 0 の時には S_0，1 の時には S_1，2 の時には S_2，3 の時には S_3 が割り当てられている．
- 制御信号 (x_1, x_0) とクロック clk を入力，カウンタ値 (q_1, q_0) を出力とする．
- カウンタ値 (q_1, q_0) を 2 進数として q_0 を最下位ビットで表現する．
- カウンタ値の初期値 (initial value) は 0 とする．
- Q_1 と q_1，Q_0 と q_0 がそれぞれ接続されている．
- $(x_1, x_0) = (0, 0)$ のとき，現在の値を次の時刻でも保持する．
- $(x_1, x_0) = (0, 1)$ のとき，現在の値に 1 を加算 (add) した値を次の時刻で保持する．ただし，現在の値が 3 のときは次の時刻で 0 を保持する．
- $(x_1, x_0) = (1, 0)$ のとき，現在の値から 1 を減算 (subtract) した値を次の時刻で保持する．ただし，現在の値が 0 のときは次の時刻で 3 を保持する．
- $(x_1, x_0) = (1, 1)$ は禁止入力である．

以下の各小問に答えよ．

(3-1) カウンタの状態遷移図 (state transition diagram) を示せ．ただし，太い矢印が指し示している状態が開始状態 (start state) であるとする．

(3-2) (3-1) で作成した状態遷移図を基に，カウンタの状態遷移表 (state transition table) を作成する．表 1 に状態遷移表を示す．表 1 の (A)〜(L) に入る状態をそれぞれ S_0〜S_3 で示せ．

(3-3) (3-2) で作成した状態遷移表を基に，各フリップフロップの入力 D_1，D_0 を，カウンタへの入力 x_1，x_0 および各フリップフロップの出力 Q_1，Q_0 を用いた最簡積和形の論理式で示せ．

表 1: 状態遷移表

現在の状態	次の状態		
	$(x_1, x_0) = (0, 0)$	$(x_1, x_0) = (0, 1)$	$(x_1, x_0) = (1, 0)$
S_0	(A)	(B)	(C)
S_1	(D)	(E)	(F)
S_2	(G)	(H)	(I)
S_3	(J)	(K)	(L)

7 【選択問題】数学解析と信号処理 （情報工学 16/16）

配点: (1-1)25 点, (1-2)20 点, (1-3)30 点, (2-1)15 点, (2-2)15 点, (2-3)20 点

以下の各問に答えよ.

(1) 以下の各小問に答えよ.

(1-1) 以下の通り定義されるガンマ関数 (Gamma function) を考える.

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$$

ただし, z は実部が正 (positive real part) である複素数 (complex number) とする. このとき $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ であることを証明せよ.

(1-2) $t > 0$, s を複素変数 (complex variable) とする.

$$f(t) = \frac{1}{\sqrt{t}}$$

であるとき, $f(t)$ のラプラス変換 (Laplace transform) $\mathcal{L}[f(t)]$ が $\sqrt{\frac{\pi}{s}}$ であることを証明せよ.

(1-3) 以下の式の逆ラプラス変換 (inverse Laplace transform) $\mathcal{L}^{-1}[G(s)]$ を求めよ.

$$G(s) = \frac{1}{\sqrt{s}(s-1)}$$

(2) 離散時間 (discrete-time) の線形時不変システム (linear time-invariant system) について, 以下の各小問に答えよ. ただし, n は整数 (integer), z は複素変数 (complex variable) である.

(2-1) 以下の離散時間信号 (discrete-time signal) $x(n)$ の z 変換 (z-transform) $X(z)$ を求めよ. ここで $\delta(n)$ は単位インパルス (unit impulse) を意味する.

$$x(n) = \delta(n) - \delta(n-1) + 2\delta(n-2)$$

(2-2) あるシステムに離散時間信号 $x(n)$ を印加したときの応答 (response) が以下の $y(n)$ で与えられた. このシステムの伝達関数 (transfer function) $H(z)$ を求めよ.

$$y(n) = x(n) + 2x(n-1) - 3x(n-2)$$

(2-3) 以下の伝達関数 $H(z)$ を有するシステムの振幅特性 (amplitude characteristics) と位相特性 (phase characteristics) を求めよ.

$$H(z) = 1 + 2z^{-1} + z^{-2}$$

1 【必須問題】アルゴリズムとプログラミング

配点：(1-1) 15 点，(1-2) 10 点，(1-3) 30 点，(2-1) 20 点，(2-2-1) 30 点，(2-2-2) 20 点

図 1 に示す ANSI-C 準拠である C 言語のプログラム (program) は任意の 2 人が同じ組織 (organization) に所属するか判定し，その結果を出力 (output) するものである．人 (構成員 (member) と呼ぶ) はN人 (Nは正の整数 (positive integer)) 存在し，各構成員には直属の上司 (direct supervisor) である構成員が 1 人以下 (1 or less) 存在する．組織は一つ以上存在し，各組織は，各構成員を節点 (ノード (node))，その直属の上司を親 (parent) とする木 (tree) で表される．同じ組織に所属する構成員は，その組織の最上位の上司を根 (root) とする一つの木を構成する．

各構成員にはそれぞれ 0～N-1 の整数 (integer) が構成員番号として重複なく付与されている．配列 (array) p は構成員番号 i である構成員の直属の上司の構成員番号を要素 (element) として p[i]に格納し，直属の上司が存在しない場合は自身の構成員番号を格納する．このデータ構造を用い，関数 same は 2 人の構成員番号を引数とし，同じ組織に所属するかどうかを標準出力に出力する．input.txt，pair.txt という図 2 および図 3 にそれぞれ示すようなフォーマットのファイルが存在するものとし，図 1 のプログラムでそれらを読み込み実行する．input.txt の 1 行目には構成員の総数N (Nは N_MAX 以下の正の整数)，2 行目以降の各行には，構成員とその直属の上司の構成員番号のペア (pair) がこの順で書かれている．pair.txt の各行には，同じ組織に所属するか判定したい構成員のペアの構成員番号が書かれている．以下の各問に答えよ．

(1) 図 1 のプログラムは，図 2 の input.txt と図 3 の pair.txt を読み込み実行する．以下の各小問に答えよ．

(1-1) 21～29 行目で読み込まれる全ての木を示せ．ただし図 4 にならい，丸でノードを，丸の中の数字で構成員番号を，線で枝 (edge) を表すこと．

(1-2) find(x)が意味する内容を，x を用いて説明せよ．

(1-3) 14 行目の空欄 A に当てはまる式を，15 行目の空欄 B に当てはまる条件式をそれぞれ書け．

(2) 非常に大きな数の構成員に対し，多数の無作為に選ばれた構成員のペアのそれぞれが同じ組織に所属するか判定したい．そのため，このようなデータを含む input.txt および pair.txt を図 1 のプログラムで読み込み実行する．ただし，図 1 のプログラム 2 行目の N_MAX の値を適切に変更するものとする．以下の各小問に答えよ．

(2-1) 関数 same を実行する際の，1 回当たりの平均時間計算量 (average time complexity) をオーダ表記 (order notation) で表せ．また理由も答えよ．ただし，ノードの平均の深さ (average depth) をhとする．

(2-2) ここで，関数 find の 9 行目を変更し，最上位の上司を格納するように配列 p を更新することで，実行時間を短縮できる場合がある．以下に答えよ．

(2-2-1) 変更後の 9 行目を一文で書け．

(2-2-2) 関数 same を十分大きな回数実行した場合に漸近する，1 回当たりの平均時間計算量をオーダ表記で表せ．また理由も答えよ．

```
1  #include <stdio.h>
2  #define N_MAX 1000
3  int p[N_MAX];
4
5  int find(int x) {
6     if (p[x] == x)
7        return x;
8     else
9        return find(p[x]);
10 }
11 void same(int x, int y) {
12    int n, m;
13    n = find(x);
14    m = [ A ];
15    if ( [ B ] )    /*同じ組織に所属する*/
16       printf("%d & %d are in the same organization.\n", x, y);
17    else
18       printf("%d & %d are in different organizations.\n", x, y);
19 }
20 int main(void) {
21     FILE *fp;
22     int n, i, j, sv;
23     fp = fopen("input.txt", "r");
24     fscanf(fp, "%d", &n);
25     for (i = 0; i < n; i++)
26         p[i] = i;
27     while (fscanf(fp, "%d %d", &i, &sv) != EOF)
28         p[i] = sv;
29     fclose(fp);
30     fp = fopen("pair.txt", "r");
31     while (fscanf(fp, "%d %d", &i, &j) != EOF)
32         same(i, j);
33     fclose(fp);
34     return 0;
35 }
```

図1 プログラム

10	
2	0
3	1
4	3
6	3
7	1
8	4
9	8

図2 input.txt

9	7
6	4
0	2

図3 pair.txt

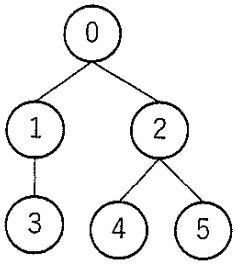


図4 木の表記例

2 【必須問題】計算機システムとシステムプログラム （情報工学 3/13）

配点： (1-1) 8 点，(1-2) 10 点，(1-3-1) 18 点，(1-3-2) 29 点，
(2-1) 18 点，(2-2-1) 27 点，(2-2-2) 10 点，(2-2-3) 5 点

(1) キャッシュメモリ (cache memory) および仮想記憶 (virtual memory) を含む，計算機 (computer) の記憶システム (memory system) に関する以下の各小問に答えよ．解答は全て解答用紙の太線枠内に書くこと．

(1-1) キャッシュメモリや仮想記憶では，(a)「一度アクセス (access) されたアドレス (address) は近いうちに再アクセスされる可能性が高い」，(b)「一度アクセスされたアドレスに近接するアドレスは，近いうちにアクセスされる可能性が高い」というメモリアクセスの性質が一般に利用されている．これらの性質の名称を記せ．

(1-2) (a) キャッシュメモリを導入することによる効果，(b) 仮想記憶を導入することによる効果を，それぞれ一つずつ，簡潔に記せ．

(1-3) キャッシュメモリおよび仮想記憶を有し，以下の箇条書きに示す仕様 (specification) を持つ記憶システムを考える．なお，論理アドレス (logical address) から物理アドレス (physical address) への変換が，CPU とキャッシュメモリの間で行われる場合と，キャッシュメモリと主記憶 (main memory) の間で行われる場合の，二通りの方式がある．前者は物理アドレスキャッシュと呼ばれ，物理アドレスを用いてキャッシュメモリへのアクセスが行われる．後者は論理アドレスキャッシュと呼ばれ，論理アドレスを用いてキャッシュメモリへのアクセスが行われる．ここでは，これら双方の方式について考える．この記憶システムに関し，(1-3-1)，(1-3-2) に答えよ．

- アドレスは 1 [バイト] ([byte]) 毎に付与される．
- キャッシュメモリのマッピング (mapping) 方式は，直接マッピング (direct mapping) 方式．
- キャッシュメモリの 1 [ブロック] ([block]) は 2 [バイト]．
- キャッシュメモリ容量は 8 [バイト]．但し，タグ (tag) 等に必要な容量は，これには含まれない．
- 仮想記憶の制御方式は，ページング (paging) 方式．
- 仮想記憶の 1 [ページ] ([page]) は 8 [バイト]．
- 主記憶容量は 32 [バイト]．但し，実際のシステムでは主記憶上にページングの対象とならない領域が存在するが，ここでは，便宜上，主記憶の全ての領域がページングの対象となるものとする．
- 仮想記憶容量は 256 [バイト]．

(1-3-1) 以下の値 (a)～(f) を記せ．

(a) 論理アドレスを表すのに必要な最小ビット長 (minimum bit length) [ビット] ([bit]).

(b) キャッシュメモリ内のブロック数．

(c) ページテーブル (page table) のエントリー (entry) 数．

(d) 主記憶内におけるページ番号を表わすのに必要な最小ビット長 [ビット]．

(e) 物理アドレスキャッシュにおける，キャッシュメモリの各タグの最小ビット長 [ビット]．

(f) 論理アドレスキャッシュにおける，キャッシュメモリの各タグの最小ビット長 [ビット]．

(1-3-2) 物理アドレスキャッシュにおいて，サイクル (cycle) 0～7 に，以下の表に示す順に論理アドレスが参照されたとする．初期状態において，キャッシュメモリ内のブロックは空であり，主記憶内のページ枠 (page frame) 0～3 には，それぞれ，仮想記憶内のページ 0，1，2，31 が割り当てられているとする．主記憶，仮想記憶，共に，アドレス 0 からページ 0 が始まり，連続してページが配置されるものとする．0～7 の各サイクルでの，(a) 参照論理アドレスに対応する主記憶内のブロック番号，(b) アクセスされるキャッシュメモリ内のブロック番号，(c) アクセスされるキャッシュメモリブロックのタグの値を記せ．また，(d) サイクル 0～7 におけるキャッシュヒット率を記せ．

サイクル	0	1	2	3	4	5	6	7
参照論理アドレス	0	1	2	3	4	8	21	255

(2) ファイルシステム (file system) に関する以下の各小問に答えよ．解答は全て解答用紙の太線枠内に書くこと．

(2-1) 以下の文中の空欄 [(ア)] ～ [(カ)] それぞれに最も適切な語を選択肢から一つ選び，その記号を答えよ．ただし，同じ選択肢を複数回用いてはならない．

ファイルシステムは，ファイルをファイル装置 (file device) に割り付ける (allocate) 等のファイル管理機能を提供する．ファイルは，ブロック (block) と呼ばれるファイル装置上の最小構成単位 (minimum configuration unit) でファイル装置に物理的に格納されており，ファイルへのアクセス (access) もブロック単位で行われる．ファイルへのアクセス方式には，ファイルを構成する先頭ブロックから末尾に向かってアクセスする [(ア)] と，ファイルを構成するブロックのいずれにも任意順にアクセスできる [(イ)] がある．

[(イ)] の場合のファイル割り付け方式を考える．ファイル装置上の連続したブロックに格納する連続ファイル割り付け (contiguous file allocation) を用いると，ファイル生成時に [(ウ)] を見積もる必要があるという問題や，ファイル装置上の未使用領域が [(エ)] した状態になるといった問題が生じる．これらの問題を回避するため，各ブロックへのポインタ配列 (array of pointers) を索引ブロックに格納しておく [(オ)] や，各ブロックに次のブロックへのポインタ（アドレス）を持たせておく [(カ)] 等が用いられる．

選択肢

(A) インデックスファイル割り付け（索引ブロックを用いた割り付け, indexed file allocation), (B) 直接アクセス (direct access), (C) 非断片化 (defragmentation), (D) プロセス割り付け (process allocation), (E) 最大ファイルサイズ (maximum file size), (F) 順アクセス（逐次アクセス，sequential access), (G) 断片化 (fragmentation), (H) アクセス時間 (access time), (I) リンクファイル割り付け（連結リスト割り付け, linked file allocation)

(2-2) ブロックサイズ (block size) が b [バイト], ブロックのアドレスの長さが a [バイト] で構築されたファイルシステムに，以下の三つの方式それぞれによってファイルを割り付けることを考える．

(i) 連続ファイル割り付け

(ii) リンクファイル割り付け

(iii) インデックスファイル割り付け

なお，方式 (ii) において，各ブロックには，次のブロックのアドレスを格納する容量が必要となることに注意せよ．また，方式 (iii) において，各ファイルの索引は一つの索引ブロックに収まるものとする．解答に際しては，床関数 (floor function) floor(x)（実数 x 以下の最大の整数を返す関数）や，天井関数 (ceiling function) ceil(x)（実数 x 以上の最小の整数を返す関数）を用いて良い．次の (2-2-1)～(2-2-3) に答えよ．

(2-2-1) ファイルサイズ s [バイト] のファイルを，方式 (i)～(iii) のそれぞれによってファイルシステムに割り付けた場合を考え，以下の (a)～(c) を求めよ．

なお，(a), (b) を求める際には，次の二点に注意せよ．

・各ブロックへの事前アクセスは無く，初めてアクセスするものとする．

・同一ブロックへ複数回アクセスする場合であっても，アクセスするブロック数は 1 である．

(a) ファイルの n ($1 \leq n \leq s$) [バイト] 目にアクセスする場合に，アクセスするブロック数

(b) ファイルの先頭から n ($1 \leq n \leq s$) [バイト] 目まで順にアクセスする場合に，アクセスするブロック数

(c) ファイルを格納するのに必要となるファイル装置上の最小サイズ p（ブロックサイズの整数倍）[バイト]

(2-2-2) ファイル装置上の最小サイズ p に対するファイルサイズ s の割合を利用効率 e ($= s/p$) として定義する．また，ブロックサイズに対するファイルサイズの比を r ($= s/b$) として定義する．(2-2-1) の結果を踏まえて，方式 (i) と (iii) のそれぞれについて，r に対する利用効率 e の変化を，$0 < r \leq 3$ の範囲で図示せよ．

(2-2-3) (2-2-2) の結果を踏まえて，r が大きくなるにつれて，方式 (i) の利用効率に対する方式 (iii) の利用効率の比がどのように変化するか，簡潔に述べよ．

3 【選択問題】離散構造 （情報工学 5/13）

配点:(1-1)~(1-4) 各 6 点，(1-5-1) 16 点，(1-5-2) 25 点，(2-1-1) 15 点，(2-1-2) 15 点，(2-2-1) 15 点，(2-2-2) 15 点

(1) 一階述語論理（first-order predicate logic）に関する以下の各小問に答えよ．ただし，論理式（logic formula）の記述には以下の記号を用いる．$\forall$，$\exists$ はそれぞれ全称作用素（universal quantifier），存在作用素（existential quantifier）であり，$\Leftrightarrow$，$\rightarrow$，$\wedge$，$\vee$，$\neg$ はそれぞれ等価（equivalence），含意（implication），論理積（conjunction，and），論理和（disjunction，or），否定（negation，not）を表す論理演算子である．また，必要に応じて $\bigwedge_{1\leq i\leq n} T_i$（あるいは $\bigvee_{1\leq i\leq n} T_i$）の表記を用いる．これは論理式 T_i の i を 1 から n（n は正整数）まで順次変えながら論理積（あるいは論理和）で結合した式を意味している．

本問では，要素（element）の列であるリスト（list）を扱う．先頭の要素が H であり，残りの要素により構成されるリストが T であるとき，このリストを $[H|T]$ で表す．また，空のリストを $[\,]$ で表し，要素数が 1 のリスト $[H|[\,]]$ は簡略化して $[H]$ と表す．以降，X, Y, Z はリストを表す変数とし，A はリストの要素を表す変数とする．リスト X とリスト Y をこの順に接合したリストがリスト Z であることを表す述語（predicate）を $append(X, Y, Z)$ とし，リスト X の要素の順を逆にしたリストが Y であることを表す述語を $reverse(X, Y)$ とする．以下の各小問に答えよ．

(1-1) 「任意のリスト X に対して，空のリスト $[\,]$ とリスト X をこの順に接合したリストがリスト X である」ことを表す閉論理式（closed formula）を示せ．

(1-2) 「$append(X, Y, Z)$ が成り立つ任意のリスト X，Y，Z に対して，リスト X の先頭に任意の要素 A を追加したリストとリスト Y とをこの順に接合したリストは，リスト Z の先頭に要素 A を追加したリストである」ことを表す閉論理式を示せ．

(1-3) 「空リストを逆順にしたものは空リストである」ことを表す論理式を述語 *reverse* を用いて示せ．

(1-4) 述語 *reverse* に対して成り立つ以下の論理式について，空欄 α を埋めよ．

$$\forall A\ \forall X\ \forall Y\ \forall Z((reverse(X, Y) \wedge append(Y, [A], Z)) \rightarrow reverse(\boxed{\quad \alpha \quad}, Z))$$

(1-5) 論理プログラミングでは以下の論理式の恒真性（validity）を問うことで解を探索する．

$$\left(\bigwedge_{1\leq i\leq m} (\forall x_1 \cdots \forall x_h((P_{i1} \wedge \cdots \wedge P_{in_i}) \rightarrow Q_i))\right) \rightarrow \exists x_1 \cdots \exists x_h(R_1 \wedge \cdots \wedge R_k) \tag{1}$$

ただし，すべての P_{i1}，$\cdots$，P_{in_i}，Q_i，R_j $(1 \leq i \leq m, 1 \leq j \leq k)$ は原子論理式（atomic formula）であり，x_1，$\cdots$，x_h はこれらの原子論理式中に現れる変数で，以降の問では $h \geq 1$ とする．また，$n_1, \cdots, n_m \geq 0$ である．この論理式の恒真性を調べるには，この論理式の否定である式 (2) の充足不能性（unsatisfiability）を調べればよい．

(1-5-1) 式 (1) の否定をとると式 (2) となる．空欄 β および空欄 γ を埋めよ．ただし，解答には含意（$\rightarrow$）は含めないこと．

$$\forall x_1 \cdots \forall x_h(\boxed{\quad \beta \quad} \wedge \boxed{\quad \gamma \quad}) \tag{2}$$

(1-5-2) 式 (1) の否定が式 (2) であるという事実と小問 (1-1)~(1-4) で得られた論理式を利用して，導出原理（resolution principle）により，$reverse([a|[b]], W)$ の解が存在することと，そのときの W への代入を示せ．解の探索過程も記すこと．ただし，a，b はリストの要素を表す定数，W はリストを表す変数とする．

(2) 直立した m 本の棒（rod）1, 2, ..., m と，中心に穴があいた n 枚の円盤（disk）がある．円盤は大きさがすべて異なり，小さい順に 1, 2, ..., n で表す．各円盤 i はその穴によって，必ずいずれかの棒に挿されている．i から j $(1 \leq i \leq j \leq n)$ までの $j-i+1$ 枚の円盤が棒 k $(1 \leq k \leq m)$ に挿されているとき，かつそのときに限り真（true）となる述語（predicate）を $S_k(i,j)$ で表す．

$S_1(1,n)$ が成り立つ状態（state）から，$S_m(1,n)$ が成り立つ状態に移行するために，円盤を 1 枚ずつ移動させる手順（移動手順）を考える．ただし，いかなる状態においても，ある円盤の上にそれより大きい円盤があってはならず，ある円盤はその上に円盤がないときに限り移動できる．図 1 は，$m=3$ および $n=3$ の場合の，$S_1(1,n)$ が成り立つ状態（左図），その状態から円盤 1 を棒 2 に移動した状態（中図），$S_m(1,n)$ が成り立つ状態（右図），を表している．

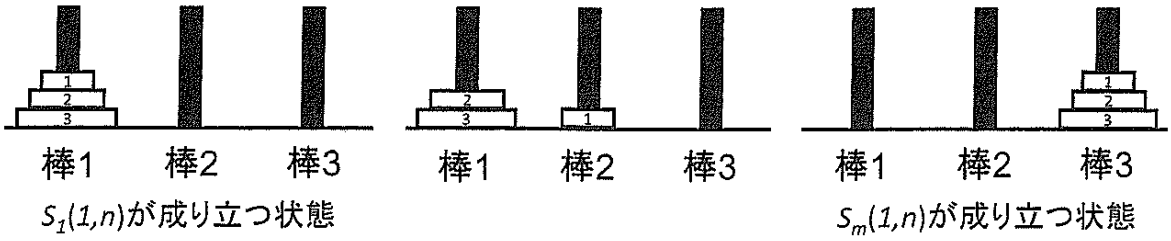


図 1: $m=3$ および $n=3$ の場合の状態の例

以下の各小問に答えよ．

(2-1) $m=3$ とする．このとき，$S_1(1,n)$ が成り立つ状態から，$S_3(1,n)$ が成り立つ状態へ移行する移動手順のうち，円盤の移動回数が最小のものを F_n で表し，F_n の円盤の移動回数を f_n で表す．以下の (2-1-1) および (2-1-2) に答えよ．

(2-1-1) 漸化式（recurrence relation）を用いた求め方により，f_n を n の式で表せ．

(2-1-2) F_n において，各円盤 i $(1 \leq i \leq n)$ が棒 1 から初めて移動されるとき，どの棒に移動されるかを答えよ．

(2-2) $m=4$ とし，ある円盤 d $(1 \leq d \leq n)$ に着目する．$S_1(1,n)$ が成り立つ状態から，$S_3(1,d)$ が成り立つ状態を少なくとも 1 つ経て，$S_4(1,n)$ が成り立つ状態に移行する移動手順を考える．それらの移動手順のうち，いかなる状態においても以下の (条件 1) および (条件 2) を満たし，かつ，円盤の移動回数が最小であるものを G_n で表し，G_n の円盤の移動回数を g_n で表す．

（条件 1）円盤 d およびそれより小さいどの円盤も棒 2 に挿されていてはならない．

（条件 2）円盤 d より大きいどの円盤も棒 3 に挿されていてはならない．

以下の (2-2-1) および (2-2-2) に答えよ．

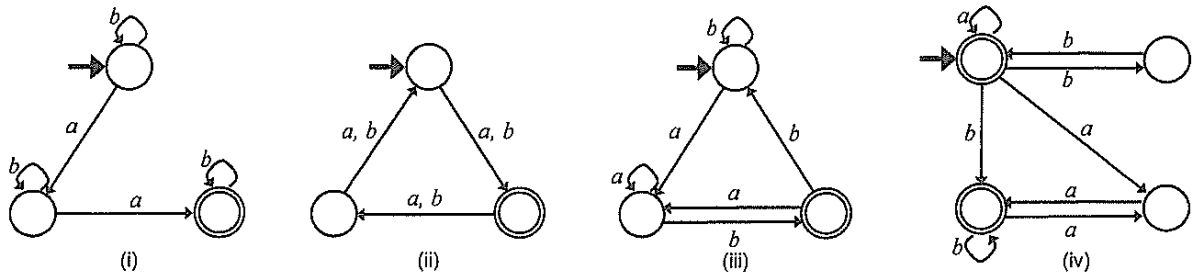
(2-2-1) g_n を n と d の式で表せ．求め方も述べよ．

(2-2-2) $f_n = g_n$ となる d の値を n で表せ．求め方も述べよ．

（配点：(1-1) 20 点, (1-2-1) 20 点, (1-2-2) 20 点, (2-1) 15 点,
(2-2) 5 点, (2-3) 10 点, (2-4) 10 点, (2-5) 25 点）

(1) オートマトン (automaton) に関する以下の各小問に答えよ．

(1-1) 下の (i), (ii), (iii), (iv) は，有限オートマトン (finite automaton) の状態遷移図 (state transition diagram) を表している．太い矢印 (thick arrow) が指し示している状態が開始状態 (start state)，二重丸で表されている状態が最終状態 (final state) である．これらのオートマトンそれぞれについて，そのオートマトンが受理する言語 (language) を表している正則表現 (regular expression) を，1〜8 の中から一つ選べ．

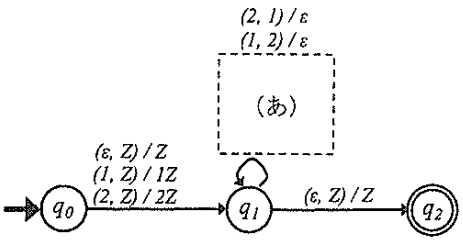


1. $(a+b)a^*b^*$　2. $b^*a+aa^*(a+b)$　3. $(a+bb)^*(aa+b)^*$　4. $b^*+a^*+b^*a^*$
5. $(a+b)^*ab$　6. $b^*a^*b(a+b)^*$　7. $b^*ab^*ab^*$　8. $((a+b)(a+b)(a+b))^*(a+b)$

(1-2)10 進数 (decimal) で表現された正の整数 (positive integer) のうち，3 で割り切れるものを受理するオートマトンを考える．なお，10 進数で表現された正の整数を 3 で割った余りは，その正の整数を構成する各桁の数の和を 3 で割った余りと等しくなることを利用して良い．例えば，122 における各桁の数の和は 1+2+2=5 であるので，122 を 3 で割った余りは 2 となる．(1-2-1) および (1-2-2) に答えよ．

(1-2-1) 決定性有限オートマトン (deterministic finite automaton) は $(Q, \Sigma, \delta, p_0, F)$ で表される．ただし，Q は状態 (state) の有限集合，Σ は入力アルファベット (input alphabet)，δ は遷移関数 (transition function)，p_0 は開始状態，F は最終状態の集合である．決定性有限オートマトン $A_D = (\{p_0, p_1, p_2\}, \Sigma_D, \delta_D, p_0, \{p_0\})$ を用いて実現する場合，A_D の状態遷移図を書け．開始状態には太い矢印を付与すること．最終状態は二重丸で表現すること．簡単化のため，入力アルファベット Σ_D は Σ_D={1,2} とする．例えば，1221 や 222111 は受理される語 (word) であるが，1211 や 2222 は受理されない語である．

(1-2-2) プッシュダウンオートマトン (pushdown automaton) は $(Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ で表される．ただし，Q は状態の有限集合，Σ は入力アルファベット，Γ はスタックアルファベット (stack alphabet)，δ は遷移関数，q_0 は開始状態，Z はスタックの開始記号，F は最終状態の集合である．最終状態による受理 (acceptance by final state) を行うプッシュダウンオートマトン $A_P = (\{q_0, q_1, q_2\}, \Sigma_P, \Gamma_P, \delta_P, q_0, Z, \{q_2\})$ を用いて実現する場合，以下の状態遷移図の「(あ)」の部分に必要な動作を全て答えよ．なお，「(あ)」の上部にすでに記述されている $(2,1)/\varepsilon$ および $(1,2)/\varepsilon$ は解答用紙に記述する必要はない．ε は空列を表す．スタックアルファベット Γ_P は $\Gamma_P = \{Z, 1, 2\}$ とする．簡単化のため，入力アルファベット Σ_P は Σ_P={1,2} とする．



(2) 文脈自由文法 (context-free grammar) は一般に $G = (V, T, P, S)$ で定められる．ただし V は変数 (variable; 非終端記号 nonterminal symbol) の集合，T は終端記号 (terminal symbol) の集合，P は生成規則 (production rule) の集合，S は出発記号 (start symbol) であり，$S \in V$ である．文脈自由文法 G_1 を以下の通り定め，G_1 により生成される言語を L_1 で表す．以下の各小問に答えよ．

$G_1 = (V_1, T_1, P_1, S_1)$

- $V_1 = \{A\}$
- $T_1 = \{(,)\}$
- $P_1 = \{A \to AA,\ A \to (A),\ A \to ()\}$
- $S_1 = A$

なお，L_1 に含まれる列 (sequence; 語 word) は，左括弧 (left parenthesis) と右括弧 (right parenthesis) のバランス (balance) がとれているものである．列のバランスがとれているとは，その列の中の列 () の削除を 1 回以上可能な限り繰り返すと，最終的に空列 ε が得られるとき，およびその時のみをいう．例えば列 ((((())))) や列 (())()() はバランスがとれているが，列)((()) や列 (())(() はバランスがとれていない．

(2-1) L_1 に含まれる長さ (length) が 6 以下の列を全て記せ．

(2-2) 列 (())()() に対する構文木 (parse tree) で，互いに異なるもの全てを図示せよ．

(2-3) 列 ()()()() に対する構文木で互いに異なるものは全部で何個あるか．数を答えよ．求め方は示さなくてよい．

(2-4) 列 ()()()()() に対する構文木で互いに異なるものは全部で何個あるか．数を答えよ．求め方は示さなくてよい．

(2-5) バランスのとれている任意の列が L_1 に含まれることを，列の長さに関する帰納法 (induction) で証明したい．証明を完成させるために，空欄（ア）～（エ）を適切な内容で埋めよ．

ただし，文形式 (sentential form) α に対して生成規則を 1 回適用して文形式 β が得られるとき，$\alpha \Rightarrow \beta$ と表記する．また，文形式 α に対して生成規則を 0 回以上適用して文形式 β が得られるとき，$\alpha \overset{*}{\Rightarrow} \beta$ と表記する．

証明: バランスのとれている列の長さは正の偶数 (even number) である．

- 長さ 2 のバランスがとれている列，すなわち列 () の場合，導出 (derivation) $A \Rightarrow$ [　（ア）　] により，この列は L_1 に含まれる．
- 長さ k 以下（ただし k は 2 以上の偶数）のバランスがとれている任意の列は L_1 に含まれると仮定する．
  長さ $k+2$ のバランスがとれている任意の列を x とおく．バランスがとれている二つの列 v と w に対し，$x = vw$ の形に分解 (decomposition) できる場合と，できない場合の 2 通りがある．
  - $x = vw$ の形に分解できる場合：
    v, w はそれぞれ長さが k 以下のバランスがとれている列なので，帰納法の仮定から v, w はともに L_1 に含まれる．すなわち $A \overset{*}{\Rightarrow} v$ かつ $A \overset{*}{\Rightarrow} w$ である．従って，導出 $A \Rightarrow$ [　（イ）　] $= x$ により，列 x は L_1 に含まれる．
  - $x = vw$ の形に分解できない場合：
    このとき，長さ k のバランスのとれている列 y が存在して，$x =$ [　（ウ）　] の形をしている．y の長さは k なので，帰納法の仮定から y は L_1 に含まれる．すなわち $A \overset{*}{\Rightarrow} y$ である．従って，導出 $A \Rightarrow$ [　（エ）　] $= x$ により，列 x は L_1 に含まれる．

(証明終)

配点: (1-1) (あ)〜(う) 各 2 点，(え)〜(け) 各 4 点，(1-2) 10 点，(1-3) 10 点
(2-1) (a)〜(c) 各 6 点，(d) 12 点，(2-2) 25 点，(2-3) 20 点

(1) 2 元ハミング符号 (binary Hamming code) に関する以下の文章について，各小問に答えよ．

次の行列を検査行列 (parity check matrix) とする 2 元ハミング符号を考える．

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

この符号の符号長 (code length) は [(あ)] であり，情報記号数 (the number of information symbols) は [(い)] である．したがって，符号化率 (code rate) は [(う)] となる．次の行列はこの符号の生成行列 (generator matrix) である．

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \boxed{\text{(え)}} \\ 0 & 1 & 0 & 0 & \\ 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 1 & \end{pmatrix}$$

一般に 2 元ハミング符号では，最小距離 (minimum distance) d は [(お)] であり，冗長記号数 (the number of redundancy symbols) を m とするとき，その符号長は [(か)] となる．また，その検査行列は，どの列ベクトル (列要素) (column vector) も零ベクトルではなく，各列ベクトルは全て相異なるという性質をもつ．したがって，冗長記号数が同じと仮定すると，2元ハミング符号より符号化率が高い 2 元ブロック符号の最小距離は [(お)] より小さくなる(i) ことがわかる．

さて，2 元対称通信路 (binary symmetric channel) で，2 元ハミング符号に対して限界距離 (bounded distance) $\lfloor (d-1)/2 \rfloor$ の復号 (decoding) を行う場合，受信語中のビット誤りの個数が [(き)] 個以下の場合は正しく復号 (訂正) できるが，それより多い場合は必ず誤って復号する(ii)．なお，限界距離復号を行う方法として，受信語と [(く)] 行列から求められる [(け)] を用いる方法がよく知られている．

注．限界距離 t の復号法は，受信語から距離 t 以内に符号語が存在すればその符号語に復号し，そうでなければ復号に失敗する．

(1-1) 空欄 (あ) 〜 (け) を埋めよ．

(1-2) 下線部 (i) が成り立つ理由を述べよ．

(1-3) 下線部 (ii) が成り立つ理由について，検査行列の性質 (二重下線部) を用いて述べよ．

(2) イーサネット (Ethernet) で使用される多重アクセス制御方式 (multiple access) の一つである CSMA/CD (Carrier Sense Multiple Access / Collision Detection) 方式に関する説明文を読み，以下の各小問に答えよ．

> ブロードキャスト型ネットワーク (broadcast network) において，複数のホスト (host) が同時にフレーム (frame) を送出すると，フレームが回線上で衝突する．CSMA/CD 方式においては，各ホストはフレームを送出する前に搬送波検知 (carrier sense) を行い，他のホストがすでにフレームを送出していることが検知された場合にはフレーム送出を遅らせ，そうでない場合は直ちにフレームを送出する．また，フレーム送出中にも衝突検出 (collision detection) を行い，フレーム送出中に衝突を検出すると送出を中止する．

(2-1) 以下は，イーサネットで使用される CSMA/CD 方式においてフレームの衝突を検出した後に用いられるバックオフアルゴリズム (backoff algorithm) に関する説明文である．文中 (a) から (c) の空欄を適切な用語または数式で埋めよ．また，下線部 (d) の処理を行う理由を述べよ．

> 衝突を検出してフレーム送出を中止したホストは，時間 t だけ待った後に当該フレームの再送を試みる．t は各ホストで [(a)] に選択される．また，連続して衝突する回数が増えるに従って t の平均が大きくなるようにする(d)．
>
> CSMA/CD 方式において用いられる [(b)] バックオフアルゴリズムでは，連続して衝突する回数が増えるに従って再送待ち時間 (retransmission time) を指数的に増大させる．当該フレームの n 回目の再送においては，$0 \leq r \leq X$ を満たす整数 r を [(a)] に選択し，最大往復伝播遅延時間 (slot time) を D としたとき $t =$ [(c)] 後にフレームを再送する．このとき，X は以下のように決定される．なお，15 回を超える再送は行わない．
>
> $$X = \begin{cases} 2^n - 1 & (1 \leq n \leq 10) \\ 2^{10} - 1 & (10 < n \leq 15) \end{cases}$$

(2-2) イーサネットにおいて 10BASE-5 と呼ばれる規格では，回線容量が 10 [Mbps] であり，同軸ケーブル (coaxial cable) を使用している．また，最小フレーム長は 64 [byte]，最大フレーム長は 1518 [byte] である．

10BASE-5 においては，ホスト間の最大距離は 2.5 [km] (リピータ使用時) と規定されている．衝突検出の観点から，この規定が適切であることを最小フレーム長との関係に基づいて説明せよ．ただし，同軸ケーブルの信号伝播速度は 2×10^5 [km/s] とする．

(2-3) CSMA/CD 方式とは別の多重アクセス制御方式として Pure ALOHA 方式がある．Pure ALOHA 方式においては，CSMA/CD 方式のような搬送波検知や衝突検出を行わず，送信側ホストはフレームをすぐに送出する．一方，受信側ホストはフレームを受信すると ACK (確認応答) を送信側ホストに通知する．送信側ホストは，ある時間以内に ACK を受け取らなければフレームを再送する．

回線の混雑度合いが低い時と高い時のそれぞれにおける，Pure ALOHA 方式を用いた場合の回線のスループット (throughput, 単位時間当たりの伝送成功フレーム数) について，CSMA/CD 方式と比較してその高低を理由とともに説明せよ．ただし，フレーム長は全て等しいものとする．また，搬送波検知にかかる時間は無視できるものとする．

6 【選択問題】電子回路と論理設計 （情報工学 11/13）

配点：(1-1) 20 点，(1-2) 20 点，(1-3) 20 点，(1-4) 20 点，(2-1) 20 点，(2-2) 25 点

(1) 図 1 に示す手順で 2 数の差 (difference) の絶対値 (absolute value) を計算する組合せ論理回路 (combinational logic circuit)を考える．ただし，入力を 2 の補数表現 (two's complement expression) された 3bit の整数 $A=(a_2, a_1, a_0)$ および $B=(b_2, b_1, b_0)$，出力を 3bit の符号なし整数 $F=(f_2, f_1, f_0)$ とする．a_2, b_2, f_2 はそれぞれ A, B, F の MSB (most significant bit) を表している．まず，演算結果がオーバーフローしないように，3bit の $A=(a_2, a_1, a_0)$, $B=(b_2, b_1, b_0)$ を，それぞれ同値を表現する 2 の補数表現された 4bit の $A'=(a_3, a_2, a_1, a_0)$，$B'=(b_3, b_2, b_1, b_0)$ に拡張する．次に，A' から B' を減じた結果 $T=(t_3, t_2, t_1, t_0)$ を計算する．t_3 は T の MSB である．Tが非負であれば求めたい値が t_2, t_1, t_0に格納されているが，Tが負であれば値の補正を行う．以下の各小問に答えよ．

(1-1) $A=(0,1,1)$, $B=(0,1,0)$の時の A', B', T, Fをそれぞれ求めよ．また，$A=(1,0,1)$, $B=(0,1,0)$の時の A', B', T, Fをそれぞれ求めよ．

(1-2) A'，B' の a_3および b_3を，a_i, b_i (i=0, 1, 2) および 0, 1 を用いた論理式 (logical expression) で表せ．

(1-3) Tの t_i ($i=0, 1, 2, 3$) は，図 2 に示す 4 個の全加算器 (Full Adder: FA) に論理ゲートを追加して計算することができる．全加算器 FA_iの入力 x_i, y_i, z_i ($i=0, 1, 2, 3$) を a_i, b_i ($i=0, 1, 2, 3$), s_j, c_j (j=0, 1, 2 ただし $j<i$) および 0, 1 を用いた論理式で表せ．s_j, c_jは，それぞれ各全加算器の和出力および桁上げ出力である．ただし，$t_i=s_i$ ($i=0, 1, 2, 3$)となるようにせよ．

(1-4) t_3, t_2, t_1, t_0 を変数とする f_2, f_1, f_0 の最簡積和形(最小積和形, minimal sum-of-products expression) の論理式を導出せよ．

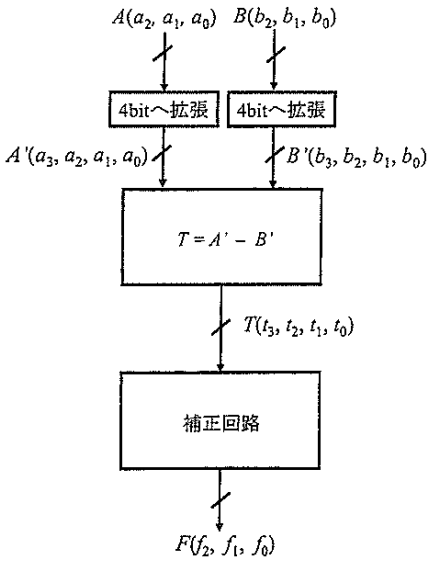


図 1

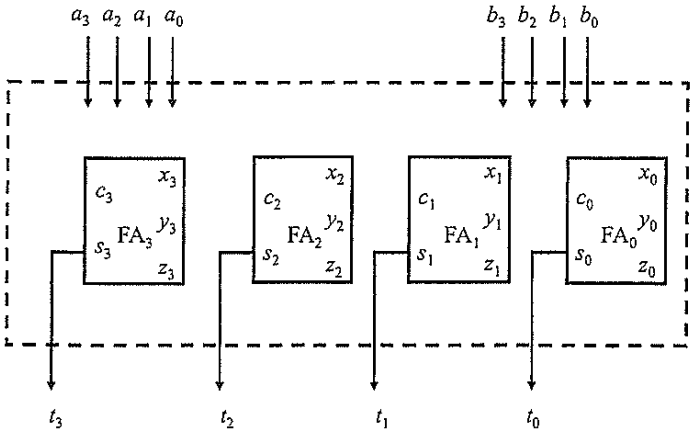


図 2

(2) 図 3 に示すような MOSFET を使った 2 入力 1 出力の CMOS 型 NOR 回路において，以下の小問に答えよ．ただし，$V_{DD}(>0)$ は電源電圧，C_L は負荷容量である．また，x_0，x_1 を入力，y を出力とし，x_0，x_1 は V_{DD} もしくは 0 の値しか取らないとする．

(2-1) 図 3 の A～D において，pMOS と nMOS のどちらが入るかをそれぞれ示せ．

(2-2) すべての MOSFET のドレイン（drain）・ソース（source）間の抵抗値（resistance）はオンにするときには瞬時に R_{MOS} となり，オフするときには瞬時に∞になると仮定する．図 3 の CMOS 型 NOR 回路において，十分に長い時間 V_{DD} であった入力 x_0，x_1 が 0 に変化してから，y における出力電圧が 0.5 V_{DD} になるまでの時間を式で示せ．

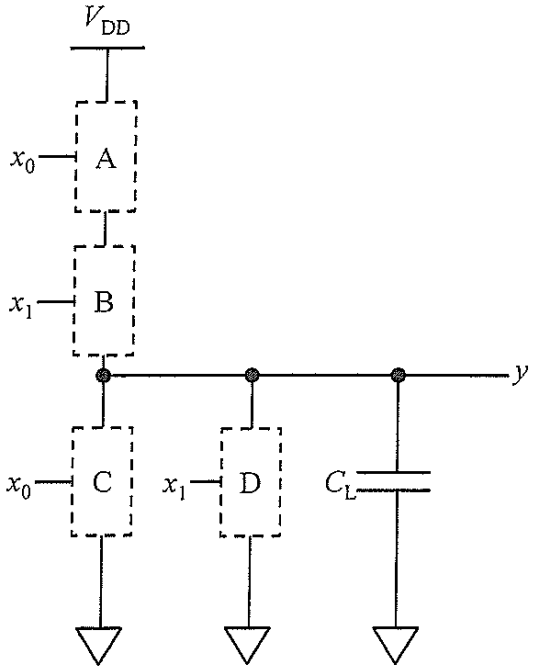


図 3

配点: (1-1)20 点,(1-2)40 点,(2-1)20 点,(2-2-1)25 点,(2-2-2)20 点

(1) 関数 $f(t)$ のラプラス変換 (Laplace transform) を $\mathcal{L}[f(t)] = F(s)$ と定義する.

以下の各小問に答えよ.

(1-1) $\frac{d}{ds}\mathcal{L}[f(t)] = -\mathcal{L}[tf(t)]$ であることを証明せよ.

(1-2) 上記の結果および $\mathcal{L}[f(t)'] = s\mathcal{L}[f(t)] - f(0)$ であることを用いて，次の微分方程式 (differential equation) を満たす $f(t)$ を求めよ.

$$tf(t)'' - (6t-1)f(t)' + 3(3t-1)f(t) = 0$$

ただし $f(t)' = \frac{df(t)}{dt}$, $f(t)'' = \frac{d^2f(t)}{dt^2}$ であり，$f''(0) = 9$ とする.

(2) 関数 $g(u)$ のフーリエ変換 (Fourier transform) を $\mathcal{F}[g(u)](\xi) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} g(u)e^{-i\xi u}du$ と定義する．ただし i は虚数単位（imaginary unit）である.

以下の各小問に答えよ.

(2-1) 下記の $g(u)$ について $\mathcal{F}[g(u)](\xi)$ を示せ.

$$g(u) = \begin{cases} u+2 & (-2 \le u < 0) \\ -u+2 & (0 \le u \le 2) \\ 0 & (otherwise) \end{cases}$$

(2-2) 下記の信号波形 (signal wave) を考える.

$$x(t) = y(t) = \begin{cases} 1 & (-1 \le t \le 1) \\ 0 & (otherwise) \end{cases}$$

(2-2-1) 畳み込み積分 (convolution integral) $x(t)*y(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)d\tau$ を求めよ.

(2-2-2) 畳み込み積分のフーリエ変換公式 (Fourier transform formula of convolution integral) $\mathcal{F}[x(t)*y(t)](\xi) = \sqrt{2\pi}\mathcal{F}[x(t)](\xi)\cdot\mathcal{F}[y(t)](\xi)$ について，(2) の下線の定義を用いて左辺と右辺をそれぞれ計算し，公式が成立することを証明せよ.

1 【必須問題】アルゴリズムとプログラミング　（情報工学 1/15）

配点：(1) 32 点，(2) 30 点，(3) 20 点，(4-1) 20 点，(4-2) 23 点

図 1 に示す ANSI-C 準拠である C 言語プログラムは，以下に挙げる 4 個の制約を満たす重み付き無向グラフ（weighted undirected graph）G における 2 頂点（vertex）間の最短経路長（shortest path length）を出力するプログラムである．

・頂点数を n として，頂点を 0～$n-1$ の非負整数（non-negative integer）で識別できること．
・すべての辺（edge）の重みが非負整数であること．
・グラフ G は連結（connected）であること．すなわち，任意の 2 頂点間に経路が存在する．
・無限大（infinity）とみなす値を INF として，任意の 2 頂点間の最短経路長が INF 未満であること．

ここで，始点（source）s から終点（destination）d への最短経路とは，s から d への経路上の辺の重みの和（sum）が最小（minimum）となる経路であり，s と d が同一頂点である場合，その最短経路長は 0 である．本プログラムは，グラフ G の頂点数 n を変数（variable）n に格納し，頂点 i から頂点 j への辺の重み $w_{i,j}$ を，配列（array）w の要素（element）w[i*n+j] に格納している．なお，存在しない辺に対しては重みの値を INF とし，任意の頂点 i に対して $w_{i,i}=0$ とする．図 2 に，図 1 で扱うグラフを示す．以下の各問に答えよ．

(1) 22～34 行目の while 文の処理において 1 巡目および 2 巡目の終了時における Len[0]～Len[3] の値を，解答用紙の太線内の空欄を埋めることにより答えよ．

(2) 関数 compute が実現しているアルゴリズムの時間計算量（time complexity）のオーダ表記（order notation）を，頂点数 n を用いて答えよ．その理由も簡潔に説明せよ．

(3) 最短経路長だけでなく，最短経路の一つを出力するように図 1 のプログラムを変更したい．例えば，始点 0 から終点 3 への最短経路であれば，以下のように出力したい．

```
Shortest Path Length from 0 to 3 is 3
- 0 - 1 - 3
```

そこで，5，17，27 および 47 行目の注釈を解除（uncomment）し，下記に示す関数 printpath の定義を 12 行目の位置に挿入した．関数 printpath を呼び出して上記のような出力を得るためには，［（ア）］および［（イ）］をどのように記述すればよいか，適切な文をそれぞれ答えよ．

```
void printpath(int v) {
  if (v != -1) { printpath(Prev[v]); printf("- %d ", v); }
}
```

(4) 図 1 の main 関数を変更することにより，頂点数を $n=4$ に固定したときの一般のグラフにおけるすべての頂点対間（all-pairs of vertices）の最短経路長を出力したい．その実現のために，44 行目以降に以下の変更のみを許す．

・関数 compute の呼び出しを追加・削除し，その実引数（actual argument）を変更してよい．
・関数 printf の呼び出しを追加・削除し，文字列や配列 Len の要素に限り，それらを出力してよい．

なお，処理対象のグラフに合わせて 38～41 行目の値は変更されているものとする．また，最短経路は出力しなくてよい．以下の各小問に答えよ．

(4-1) すべての頂点対間の最短経路長を出力するためには，関数 compute をどのように呼び出せばよいか，簡潔に説明せよ．呼び出し回数 T の値に言及し，できるだけ T の少ない方法を示すこと．

(4-2) 22 行目を以下のように変更した．変更後の関数 compute をどのように呼び出せば，すべての頂点対間の最短経路長を出力できるか，簡潔に説明せよ．呼び出し回数 T の値に言及し，できるだけ T の少ない方法を示すこと．

```
while (allvisited(visited, n)==0) {
```
変更前

→

```
while (visited[d]==0) {
```
変更後

```
 1  #include <stdio.h>
 2  #define INF  255
 3  #define MAXN 16
 4
 5  int Len[MAXN];     /*  int Prev[MAXN];  */
 6
 7  int allvisited(int *a, int n) {
 8    int i, r = 1;
 9    for (i = 0; i < n; i++) { r *= a[i]; }
10    return r;
11  }
12
13  void compute(int *w, int n, int s, int d) {
14    int i, j, next, min, visited[MAXN];
15
16    for (i = 0; i < n; i++) {
17      Len[i] = INF; visited[i] = 0;    /*  Prev[i] = -1; */
18    }
19
20    i = s; Len[i] = 0; visited[i] = 1;
21
22    while (allvisited(visited, n)==0) {
23      min = INF; next = d;
24      for (j = 0; j < n; j++) {
25        if (visited[j]==1) continue;
26        if (Len[j] > Len[i] + w[i*n+j]) {
27          Len[j] = Len[i] + w[i*n+j];    /*  [  (ア)  ]  */
28        }
29        if (min > Len[j]) {
30          min = Len[j]; next = j;
31        }
32      }
33      i = next; visited[i] = 1;
34    }
35  }
36
37  int main() {
38    int w[] = { 0,   2,   5,   INF,
39                2,   0,   1,   1,
40                5,   1,   0,   2,
41                INF, 1,   2,   0   };
42    int n = 4;
43
44    compute(w, n, 0, 3);
45    printf("Shortest Path Length from 0 to 3 is %d\n", Len[3]);
46
47    /*  [  (イ)  ]  printf("\n"); */
48
49    return 0;
50  }
```

図 1：2 頂点間の最短経路長を出力するプログラム

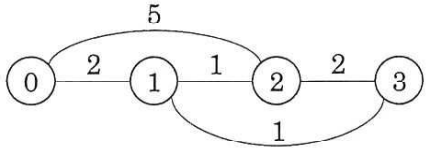


図 2：図 1 で扱うグラフ

2 【必須問題】計算機システムとシステムプログラム （情報工学 3/15）

配点： (1-1-1) 16 点，(1-1-2) 6 点，(1-1-3) 8 点，(1-2-1) 10 点，(1-2-2) 9 点，(1-2-3) 12 点
(2-1) 14 点，(2-2-1) 20 点，(2-2-2) 30 点

(1) 計算機 (computer) における数の表現と演算に関する以下の各小問に答えよ．解答は全て解答用紙の太線内に書くこと．

(1-1) 計算機において整数を扱う場合，符号なし整数 (unsigned integer) と符号付き整数 (signed integer) があり，符号付き整数の表現方法には，符号絶対値表現 (signed magnitude representation)，1 の補数 (ones' complement) 表現，2 の補数 (two's complement) 表現がある．この時，以下に答えよ．

(1-1-1) 整数を 10 ビットで表す場合，符号なし整数，符号絶対値表現，1 の補数表現，2 の補数表現それぞれで表すことができる数の最小値，最大値を 10 進数で示せ．また，それらに対応するビット列を示せ．

(1-1-2) 10 進数 -52 を 10 ビットの符号絶対値表現，1 の補数表現，2 の補数表現それぞれで表したビット列を示せ．

(1-1-3) 1010110011 が 10 ビットの符号なし整数，符号絶対値表現，1 の補数表現，2 の補数表現それぞれで表されたビット列である時，このビット列が表す数値を 10 進数で示せ．

(1-2) 整数の加算を行う回路に，桁上げ伝搬加算器 (リプル桁上げ加算器，ripple carry adder)，桁上げ先見加算器 (carry look-ahead adder) がある．4 ビットのこれらの加算器の構成を，それぞれ，図 1，図 2 に示す．ここで，FA_i ($i \in \{0,1,2,3\}$) は全加算器 (full adder) であり，s_i が加算出力，c_{i+1} が桁上げ (carry) 出力である．また，$\mathrm{FA'}_i$ は，FA_i から桁上げ出力 c_{i+1} を取り除き，出力 $g_i = a_i b_i$ および出力 $p_i = a_i + b_i$ を付加した回路である．この時，以下に答えよ．

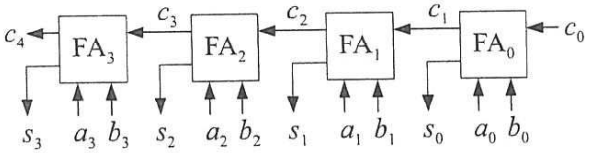


図 1．桁上げ伝搬加算器の構成

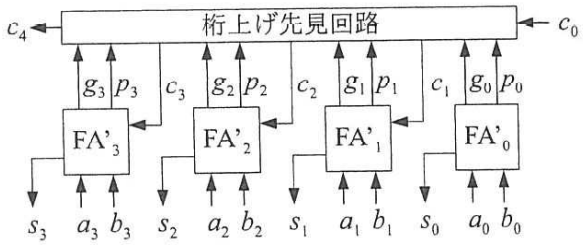


図 2．桁上げ先見加算器の構成

(1-2-1) FA_i の s_i, c_{i+1} の論理式 (logic expression) を書け．ただし，a_i, b_i, c_i を変数とする最簡積和形 (minimal sum-of-products expression) で書くこと．

(1-2-2) 図 2 の c_3 の論理式を書け．ただし，$c_0, g_0, \cdots, g_3, p_0, \cdots, p_3$ を変数とする最簡積和形で書くこと．導出過程も示すこと．

(1-2-3) 全ての入力信号 a_i, b_i, c_i が揃ってから，FA_i の出力信号 s_i, c_{i+1} が決定されるまでの遅延を，それぞれ，6 [ns], 4 [ns]，$\mathrm{FA'}_i$ の出力信号 g_i, p_i, s_i が決定されるまでの遅延を，それぞれ，2 [ns], 2 [ns], 6 [ns] とする．また全ての入力 $c_0, g_0, \cdots, g_3, p_0, \cdots, p_3$ が揃ってから，桁上げ先見回路の出力 c_1, c_2, c_3, c_4 が決定されるまでの遅延を，それぞれ，4 [ns], 5 [ns], 6 [ns], 7 [ns] とする．
この時，全ての入力信号 $a_0, \cdots, a_3, b_0, \cdots, b_3, c_0$ が揃ってからそれぞれの加算器の出力が決定されるまでの最大遅延 (maximum delay) を求めよ．また，それぞれのクリティカルパス (最大遅延経路，critical path) の一つを，信号名を用いて $a_0 \rightarrow c_1 \rightarrow s_1$ の様に示せ．

(2) プロセス管理とスケジューリング (scheduling) に関する以下の各小問に答えよ．解答は全て解答用紙の太線内に書くこと．

(2-1) 以下の文中の空欄 [（ア）] ～ [（キ）] それぞれに最も適切な語を選択肢から一つ選び，その記号を答えよ．ただし，同じ選択肢を複数回用いてはならない．

単一プロセッサ (uniprocessor) を時分割制御するなどして，ユーザに複数プログラムが同時実行されているように見せかける多重化方式を [（ア）] という．例えば，入出力処理を多く含む [（イ）] 型プロセスを実行する場合，プロセッサによる処理から入出力処理へと切り替わる際に，そのプロセスが [（ウ）] されて実行中状態から [（エ）] へと遷移し，実行可能状態にある別のプロセスが [（オ）] されて，実行中状態へと遷移する．また，実行中状態のプロセスと比べて，処理時間が短いなどの理由で優先度の高いプロセスが実行可能キュー (ready queue) にある場合は，実行中状態のプロセスを [（カ）] して強制的に実行可能状態に遷移させ，優先度の高いプロセスを実行する．このように，プロセスを適切にスケジューリングすることで，[（キ）] やスループットの増加を図っている．

選択肢

(a) マルチプログラミング (multi-programming), (b) 並列処理 (parallel processing), (c) ブロック（イベント待ち, block), (d) 横取り (preemption), (e) 実行中状態 (running state), (f) 実行可能状態 (ready state), (g) 待ち状態 (waiting state; ブロック状態, blocked state), (h) ディスパッチ (dispatch), (i) ウェイクアップ (wake-up), (j) プロセッサバウンド (processor bound; CPU バウンド, CPU bound), (k) I/O バウンド (I/O bound), (l) プロセッサ利用率 (processor utilization rate), (m) ターンアラウンド時間 (turn-around time)

(2-2) 表 1 にプロセス P1, P2, P3, P4, P5, P6 が生成される時刻，およびそれぞれのプロセスの処理時間を示す．なお，プロセス P3 は，表 2 に示す通り，入出力装置による処理を含む三つのサブプロセス P3-A, P3-B, P3-C を，順次 (sequentially) 実行する I/O バウンド型プロセスであり，それ以外のプロセス P1, P2, P4, P5, P6 には入出力処理は含まれない．下記の仮定に留意し，次の (2-2-1) および (2-2-2) に答えよ．

仮定

1. 単一プロセッサを利用する．
2. あるプロセスが生成または横取りされた時刻から，そのプロセスが実行可能キューに格納される時刻までに要する処理時間を 0 とする．
3. あるプロセスが終了した，または横取りされた時刻から，次のプロセスがプロセッサにディスパッチされる時刻までに要する処理時間（プロセス切替えに関する処理時間）を 0 とする．

(2-2-1) 四つのプロセス P1, P2, P4, P5 を実行することを考える．横取り無し (non-preemption) として，以下の二つのスケジューリングアルゴリズム (algorithm) (a) および (b) を用いた場合のプロセス P1, P2, P4, P5 のターンアラウンド時間とその平均をそれぞれ求めよ．

(a) 到着順 (first come first service (served)) 方式

(b) 最小処理時間順 (shortest processing time first) 方式

(2-2-2) 次に，(2-2-1) とは異なる組み合わせの四つのプロセス P1, P3, P4, P6 を実行することを考える．横取り有りとして，表 3 にある優先度順による二つのスケジューリングアルゴリズム (c) および (d) を用いた場合の (i) プロセス P1, P3, P4, P6 のターンアラウンド時間とその平均，および (ii) 全プロセス終了までのプロセッサ平均利用率[%]をそれぞれ求めよ．なお，最小残余時間 (shortest remaining time) は，プロセッサの処理残余時間と入出力装置の処理残余時間の和として考えるものとする．

表 1. プロセスの生成時刻と処理時間

プロセス	生成時刻	処理時間
P1	0	20
P2	5	40
P3	10	40
P4	25	30
P5	40	10
P6	45	10

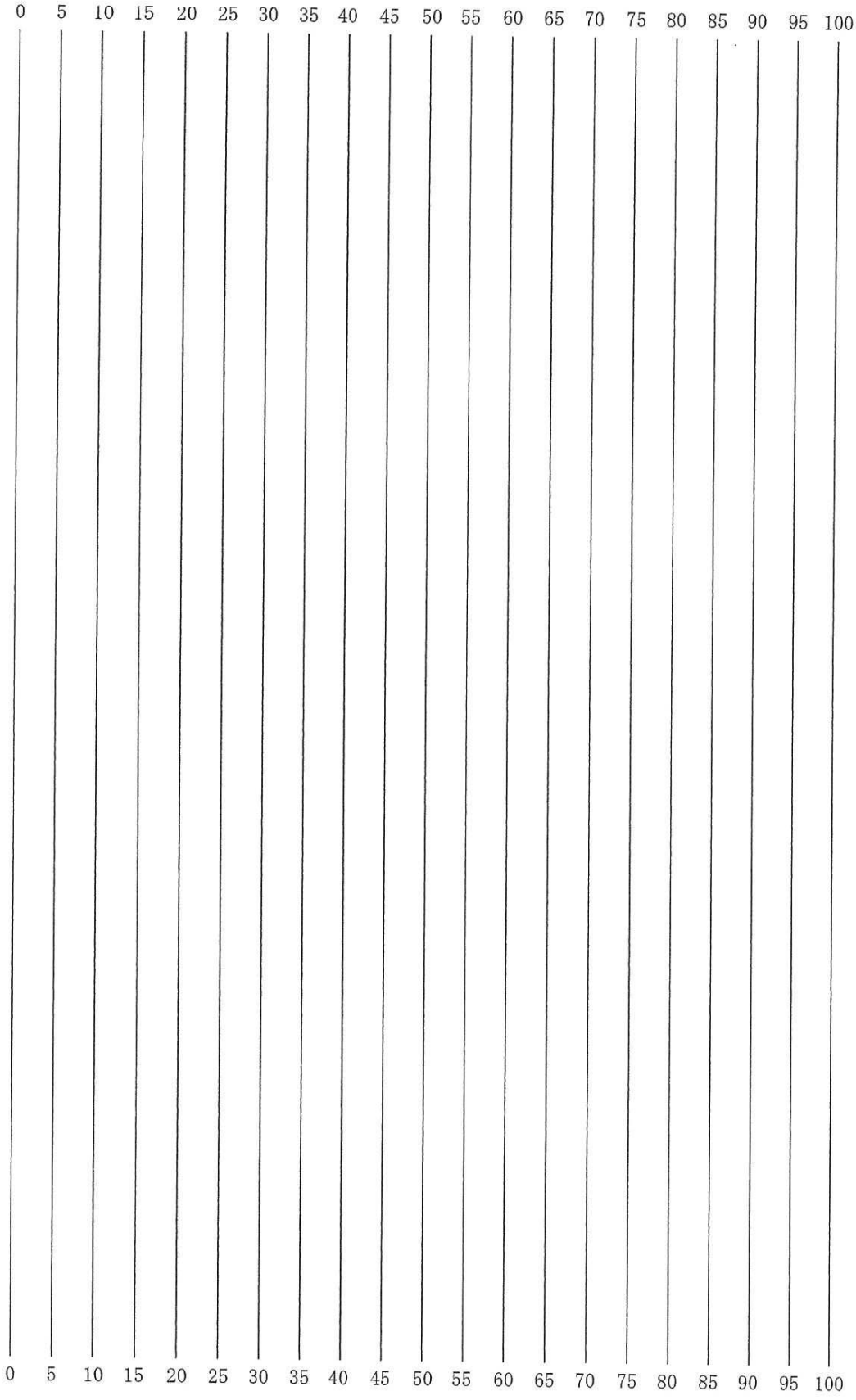
表 2. P3 のサブプロセスの利用資源と処理時間

サブプロセス	利用資源	処理時間
P3-A	プロセッサ	10
P3-B	入出力装置	20
P3-C	プロセッサ	10

表 3. 横取り有りの各スケジューリングアルゴリズムにおける優先度順

方式	第一優先	第二優先	第三優先
(c)	最小残余時間順	到着順	
(d)	プロセッサバウンド型(CPU バウンド型)プロセスよりも I/O バウンド型プロセスを優先	最小残余時間順	到着順

（計算用紙）


0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

3 【選択問題】離散構造　　（情報工学 7/15）

配点:(1-1)~(1-2) 各 10 点，(1-3) 20 点，(2-1)~(2-2) 各 10 点，(2-3-1) 10 点，(2-3-2) 15 点，(3-1)~(3-4) 各 10 点

(1) 命題論理（propositional logic）に関する以下の各小問に答えよ．$\rightarrow$ は含意（implication）を表す論理演算子とする．また，公理系（system of axioms）X において，論理式集合（set of logical formulae）Γ から論理式（logical formula）P が証明可能（provable）であるとき，$\Gamma \vdash_X P$ と書くこととする．

命題論理の公理系 S および公理系 D を次のように定める．
[公理系 S] P, Q, R を任意の論理式を表わす変数とする．S の公理（axiom）は次の $A1$ および $A2$ の 2 つである．
$A1$: $P \rightarrow (Q \rightarrow P)$
$A2$: $(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R))$
また，S の推論規則は次の $B1$ である．
$B1$: P と $P \rightarrow Q$ から Q を得る．

[公理系 D] P, Q を任意の論理式を表わす変数とし，Γ を論理式の集合とする．D の公理は次の $D1$ のみである．
$D1$: $\Gamma, P \vdash_D Q$ ならば $\Gamma \vdash_D (P \rightarrow Q)$
また，D の推論規則は公理系 S の $B1$ と同じとする．

公理系 S と公理系 D が等価であることを示すには，
(a) 公理系 S のもとで $\boxed{\alpha}$ を証明し，
(b) 公理系 D のもとで $\boxed{\beta}$ を証明すればよい．

(1-1) 上記の空欄 α および β を埋めよ．

(1-2) 公理 $D1$ は一般に何とよばれているか答えよ．

(1-3) 上記の (b) に該当する証明を示せ．

(2) 述語論理（predicate logic）において，$\Leftrightarrow$, $\rightarrow$, $\wedge$, $\vee$, $\neg$ を，それぞれ等価 (equivalence)，含意 (implication)，論理積 (conjunction, and)，論理和 (disjunction, or)，否定 (negation, not) を表す論理演算子とする．また，空でない任意の有向グラフ（directed graph）$G = (V, E)$（ただし V, E $(E \subseteq V \times V)$ はそれぞれ頂点（vertex）および有向辺（directed edge）の集合）において，以下の述語を定義する．

$n(x, y) \quad \Leftrightarrow \quad x \neq y \wedge x \in V \wedge y \in V$ であるとき真 (true)，そうでないとき偽 (false)
$e(x, y) \quad \Leftrightarrow \quad (x, y) \in E$ であるとき真，そうでないとき偽

以下の各小問に答えよ．

(2-1) $\exists y \, \forall x \, e(x, y)$ を真とする G において，$\forall x \, \exists y \, e(x, y)$ が真であることを説明せよ．

(2-2) 頂点数が 4 である G のうち，$\exists z \, \forall x \, \forall y \, (\, n(x, z) \wedge n(y, z) \rightarrow e(x, z) \wedge \neg e(z, x) \wedge \neg e(x, y) \wedge \neg e(y, x) \,)$ を真とするものを図示せよ．

(2-3) V のある部分集合 Z が，以下の（条件 X）を満たすものとする．

（条件 X）集合 Z に属さない V の各頂点は，Z に属する頂点からの有向辺を少なくとも 1 本持つ

また，以下の述語を定義する．

$Z(x) \quad \Leftrightarrow \quad x \in Z$ であるとき真，そうでないとき偽

以下の問 (2-3-1) および問 (2-3-2) に答えよ．

(2-3-1) （条件 X）を表す閉論理式（closed formula）を A とする．A を，$Z(x)$ および $e(x, y)$ を用いて記述せよ．

(2-3-2) $V=\{v_1,v_2,v_3,v_4\}$, $E=\{(v_1,v_2),(v_2,v_3),(v_3,v_4),(v_4,v_1)\}$ とする．v_2 および v_4 が集合 Z に含まれなければ，$Z=\{v_1,v_3\}$ となることを，A の変数に値を代入して導き出せる命題論理式を用い，導出原理（resolution principle）により示せ．論駁過程（refutation process）も示すこと．

(3) X,Y を空でない有限集合（finite set）とし，f,g をそれぞれ X から Y への写像（map）とする．X から直積 $Y\times Y$ への写像 h を，$x\in X$ について $h(x)=(f(x),g(x))$ で定義するとき，以下の各命題の真偽を答えよ．ただし真のときは証明を示し，偽のときは反例を示せ．

(3-1) h が全射（surjective）ならば，f が全射かつ g が全射である

(3-2) f が全射かつ g が全射ならば，h が全射である

(3-3) h が単射（injective）ならば，f が単射または g が単射である

(3-4) f が単射または g が単射ならば，h が単射である

4 【選択問題】計算理論　（情報工学 9/15）

配点：(1-1-1) 15 点, (1-1-2) 15 点, (1-2-1) 10 点, (1-2-2) 20 点, (2-1) 20 点, (2-2) 20 点, (2-3) 25 点

(1) 以下のようなプログラム Q を考える．

1. プログラム Q は整数 (integer) $x \in \mathbf{N}$ を引数 (argument) として実行される．$\mathbf{N}$ は整数の集合を表す．
2. プログラム Q は，関数 (function) a,b,c,d を呼び出すことがある．関数 a〜d の中からは，関数 a〜d を呼び出すことはない．

アルファベット (alphabet) を $\Sigma = \{a, b, c, d\}$ とし，プログラム Q の実行において関数 a〜d がどのような順序で呼び出されるかを，呼び出される順にこれらの関数名を並べた Σ 上の文字列 (string) として表す．また，引数が x の場合におけるこの文字列を $P(x)$ と表記する．例えば，引数が x_0 のとき，プログラム Q は関数 a を呼び出し，次に関数 b を呼び出し，その後，終了するのであれば，$P(x_0) = ab$ となる．

(1-1) Σ 上の言語 L_1 を，以下が成り立つような言語とする．

任意の $x \in \mathbf{N}$ について
$P(x) \in L_1 \iff x$ が引数のときにプログラム Q は関数 a を 1 回以上呼び出す

下の (1-1-1) と (1-1-2) に答えよ．

(1-1-1) 下の遷移表 (transition table) で表される決定性有限オートマトン (deterministic finite automaton) を M_1 とする．M_1 が言語 L_1 を受理するように，遷移表の空欄 (ア)〜(ク) を状態 p または q で埋めよ．ただし，表の左端の列において，→で初期状態 (initial state) を，$*$ で受理状態 (accepting state) を表している．

		入力記号 a	b	c	d
状態	$\rightarrow p$	(ア)	(イ)	(ウ)	(エ)
	$*q$	(オ)	(カ)	(キ)	(ク)

(1-1-2) 上記プログラム Q について下の式が成り立つとき，プログラム Q について成り立つ性質を簡単に説明せよ．ただし，$\overline{L_1}$ は L_1 の補集合 (complement) を，$\emptyset$ は空集合 (empty set) を表す．

$$\{P(x) \mid x \in \mathbf{N}\} \cap \overline{L_1} \neq \emptyset$$

(1-2) 正則表現 (regular expression) $(ab + cd)^*$ で表される言語を L_2 とする．下の (1-2-1) と (1-2-2) に答えよ．

(1-2-1) $P(x) \in L_2$ であるような x を引数としてプログラム Q を実行した場合，関数 a〜d がどのような順序で呼び出されるか，簡単に説明せよ．

(1-2-2) 言語 $\overline{L_2}$ (すなわち L_2 の補集合) を受理する決定性有限オートマトン M_2 を求め，遷移表の形で示せ．初期状態を→で，受理状態を $*$ で明示すること．また，どのようにして求めたのか簡単に説明せよ．

(2) 下の枠内に示す文脈自由言語 (context-free language) に対する反復補題 (pumping lemma) とその証明の概要 (proof outline) を読んで，以下の各小問に答えよ．

(2-1) 任意に整数 (integer) k (≥ 0) が与えられる．構文木 (parse tree) の高さが $k+1$ である文字列 (string) z に関して，z の長さの最大値 (maximum) を表す式を k を用いて示し，理由を簡単に説明せよ．構文木の高さとは，根 (root) から各葉 (leaf) への経路 (path) の長さの最大値と定める．根の子 (child) がすべて葉である構文木の高さは 1 である．

(2-2) 証明の概要で「A_0，A_1，A_2，...，A_k に同一の変数が 2 回以上現れる」と述べている．k と $|V|$ （V の要素数）がどのような関係にあれば同一の変数が 2 回以上現れるかを不等式 (inequality) で示し，理由を簡単に説明せよ．

(2-3) 空欄 (ア) を式で埋め，理由を簡単に説明せよ．

反復補題: 言語 L を任意の文脈自由言語とする．L に対してある定数 (constant) n が存在し，L に属する長さ n 以上の任意の文字列 (string) z は以下の条件を満たす $z = uvwxy$ の形に分解 (decompose) できる．

1. $|vwx| \leq n$ である．
2. $vx \neq \varepsilon$ である．ただし ε は空文字列 (empty string) である．
3. 任意の $i \geq 0$ に対し $uv^iwx^iy \in L$ である．

ただし文字列 s に対して，$|s|$ は s の長さを表すものとする．

証明の概要: 言語 L を生成する文脈自由文法 (context-free grammar) を $G = (V, T, P, S)$ とする．なお V は変数 (variable，あるいは非終端記号 non-terminal symbol) の集合，T は終端記号 (terminal symbol) の集合，P は生成規則 (production rule) の集合，S は開始記号 (start symbol) で $S \in V$ である．一般性を失うことなく，G はチョムスキー標準形 (Chomsky normal form) であると仮定する．チョムスキー標準形とは，生成規則の右辺が二つの変数（$A \to BC$ の形），あるいは一つの終端記号（$A \to \mathrm{a}$ の形）に限定したものをいう．

任意の $z \in L$ に対する構文木 (parse tree) において，根 (root) から葉 (leaf) への経路 (path) のうちで最も長いものを考える（構文木の根は開始記号 S であり，葉は終端記号である）．ただし最も長い経路が複数ある場合は，それらの中で任意のものを選ぶ．図 2-1 に示す通り，その経路に現れる変数を根から順に $A_0(=S)$，A_1，A_2，...，A_k とし，A_k の子 (child) は葉で終端記号である．なお A_0，A_1，A_2，...，$A_k \in V$ である．

$n =$ [(ア)] とし，任意に $z \in L$ (ただし $|z| \geq n$) を選ぶと k の値は十分に大きく，A_0，A_1，A_2，...，A_k に同一の変数が 2 回以上現れる．そのような変数を $A \in V$ とし，図 2-2 で示す通り $A = A_p = A_q$ $(p < q)$ とする．図 2-3 に示す通り $S \overset{*}{\Rightarrow} uAy \overset{*}{\Rightarrow} uvAxy \overset{*}{\Rightarrow} uvwxy = z$ は文字列 z の導出 (derivation) であり，$A \overset{*}{\Rightarrow} vAx$ かつ $A \overset{*}{\Rightarrow} w$ が成り立つ．

以上より，導出 $S \overset{*}{\Rightarrow} uAy \overset{*}{\Rightarrow} uv^iAx^iy \overset{*}{\Rightarrow} uv^iwx^iy$ を構成できるので，$uv^iwx^iy \in L$ が成り立つ． □

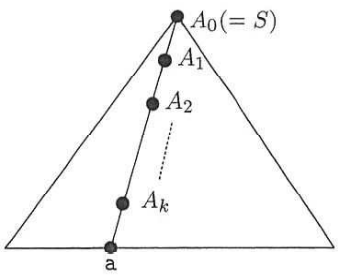


図 2-1

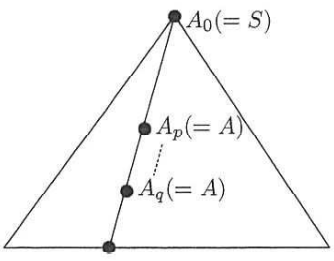


図 2-2

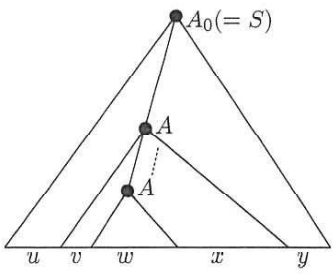


図 2-3

5 【選択問題】ネットワーク　（情報工学 11/15）

配点: (1-1-1) 5 点, (1-1-2) 14 点, (1-2) 21 点
(2-1)18 点, (2-2-1)10 点, (2-2-2)10 点, (2-2-3)15 点, (2-3)32 点

(1) 記憶がない情報源 (memoryless information source) の 2 元ハフマン符号化 (binary Huffman coding) について考える.

(1-1) 情報源記号が a, b, 各記号の生起確率 (occurrence probability) が $p(a) = 0.8$, $p(b) = 0.2$ である情報源 S_0 を考える. 以下では通信路記号を 0, 1 とし, 2 元ハフマン符号化の復号木 (decoding tree) における通信路記号の割り当ては一通りだけを示せばよい. なお, 復号木は符号の木 (code tree) とも呼ばれている.

(1-1-1) 情報源 S_0 の 2 元ハフマン符号化の復号木および平均符号語長 (average code length) を書け.

(1-1-2) 情報源 S_0 の記号を 2 個ずつまとめて出力する情報源, すなわち S_0 の 2 次拡大 (second extension) S_0^2 を考える. 記号列 aa, ab, ba, bb それぞれの生起確率を求めよ. また, S_0^2 の 2 元ハフマン符号化のすべての復号木と, 情報源記号 1 個あたりの平均符号語長を求めよ.

(1-2) 一般の記憶がない情報源 S の n 次拡大 (n-th extension) S^n について考える. 以下では, S^n の 2 元ハフマン符号化の, 情報源記号 1 個あたりの平均符号語長を l_n とする. 以下の文章の空欄 (あ) ～ (う) を埋めよ.

> 情報源 S の 2 を底とするエントロピー (entropy) を $H(S)$ とするとき, 拡大情報源 S^n のエントロピーは (あ) である. また, シャノンの情報源符号化定理 (Shannon's source coding theorem, シャノンの第一定理ともいう) から, $H(S) \leq l_n < H(S) +$ (い) が成り立つ.
> この式を用いると, 拡大情報源の情報源記号 1 個あたりの平均符号語長に関する以下の議論が成立する. 今, ある正の整数 n_0 について, l_{n_0} が $H(S)$ に等しくないとする. すなわち, ある $\delta > 0$ を用いて $l_{n_0} = H(S) + \delta$ と表せるとする. このとき, $n \geq$ (う) を満たすように n を選べば, l_n は l_{n_0} より必ず小さくなる.

(2) トランスポート層プロトコル (transport layer protocol) についての以下の各小問に答えよ.

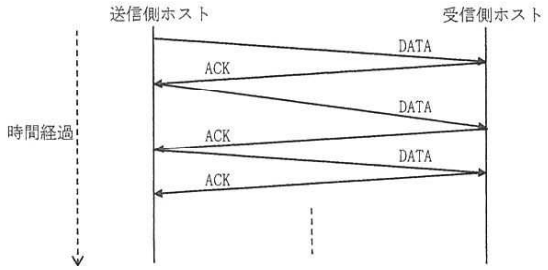
(2-1) トランスポート層プロトコルが用いるフロー制御 (flow control), ふくそう制御 (congestion control), および再送制御 (retransmission control) に関する以下の文章の空欄 (あ) ～ (か) に当てはまる最も適切な語句を選択肢から選び, その記号を答えよ. なお, 異なる空欄には異なる語句が当てはまる.

> フロー制御は, 送信側ホストと受信側ホストの (あ) や (い) などの違いを考慮し, (う) を決定するための制御である. ふくそう制御は, ネットワークが (え) になることで, ネットワーク内で発生するセグメントの廃棄や (お) の増加を防止し, ふくそう状態から回復するために, (う) を決定するための制御である. 再送制御は, 再送タイマ (retransmission timer) 等を利用し, ネットワーク内でのセグメントの (か) を検出し, 再送を行うための制御である.

選択肢

(a) セグメントサイズ (segment size)
(b) データ転送速度 (data transmission speed)
(c) 過負荷 (overload)
(d) 転送遅延 (transmission latency)
(e) 経路 (route)
(f) ループ (loop)
(g) IP アドレス (IP address)
(h) 喪失 (loss)
(i) メモリサイズ (memory size)
(j) ポート番号 (port number)
(k) スループット (throughput)
(l) 処理速度 (processing speed)

(2-2) 以下の図は，代表的なフロー制御手法であるストップアンドウェイトフロー制御 (stop and wait flow control) を用いたデータ転送の様子を表している．図中の矢印はセグメントの送信を意味し，DATA はデータセグメント (data segment) を，ACK は確認応答セグメント (acknowledgement segment) をそれぞれ表す．なお，ここでは，セグメントの喪失は発生しないものとする．



(2-2-1) データセグメントが P [バイト] のデータと H [バイト] のヘッダから構成されるものとする．また，データセグメントをネットワーク層に受け渡してから，対応する確認応答セグメントをネットワーク層から受け取るまでの時間である，ラウンドトリップ時間 (round trip time) の平均値を T [秒] とする．上図における，データ転送の平均スループット (average throughput) [ビット/秒] を答えよ．

(2-2-2) (2-2-1) の結果を用いて，データ転送のスループットの観点から，ストップアンドウェイトフロー制御の問題点を説明せよ．

(2-2-3) ストップアンドウェイトフロー制御に比べてデータ転送のスループットを向上することができるフロー制御の一つである，スライディングウインドウフロー制御 (sliding window flow control) を用いたデータ転送の様子を，上図と同様の方法で示せ．なお，ウィンドウサイズを 3 [セグメント] とし，データ転送の様子のみ解答すること．

(2-3) データセグメントの送信時に作動させる再送タイマの設定値が大きすぎる場合，および小さすぎる場合に発生する問題点をそれぞれ述べよ．また，再送タイマの設定値を適切に決定するために，TCP (Transmission Control Protocol) の一般的な実装 (implementation) において用いられている手法を説明せよ．

6 【選択問題】電子回路と論理設計　（情報工学 13/15）

配点：(1) 20 点，(2) 20 点，(3) 20 点，(4) 20 点，(5) 20 点，(6) 15 点，(7) 10 点

順序回路（sequential circuit）について，以下の各問に答えよ．

(1) 図 1 の状態遷移図（state transition diagram）と等価な状態遷移出力表（state transition table with output）を作成せよ．ただし，入力は x，出力は y，状態は S_0 から S_5 で表すこととする．

(2) (1) の順序回路の状態を表 1 のように割り当てた場合，x, Q_2, Q_1, Q_0, を変数とする y, Q_2^+, Q_1^+, Q_0^+ の最簡積和形（最小積和形, minimal sum-of-products expression）の論理式（logic expression）を導出せよ．ただし，現在の状態（current state）を（Q_2, Q_1, Q_0），次状態（next state）を（Q_2^+, Q_1^+, Q_0^+）で表すものとする．

(3) (2) で求めた最簡積和形の論理式を用いて，図 2 の CL0，CL1，CL2，CLY 内の組合せ回路（combinational circuit）をそれぞれ最小のゲート数で実現する．このときの CL0, CL1, CL2, CLY 内のゲート数（gate count）をそれぞれ求めよ．ただし，論理ゲートは，2 入力 NAND ゲートおよび 3 入力 NAND ゲートのみが使用できるものとする．最小であることの証明は不要である．

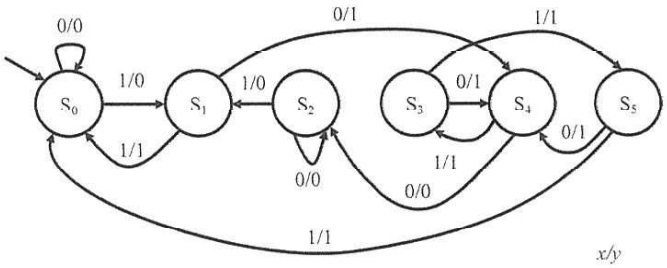


図 1

表 1

状態	状態割り当て Q_2	Q_1	Q_0
S_0	0	0	0
S_1	0	0	1
S_2	0	1	0
S_3	0	1	1
S_4	1	0	0
S_5	1	0	1

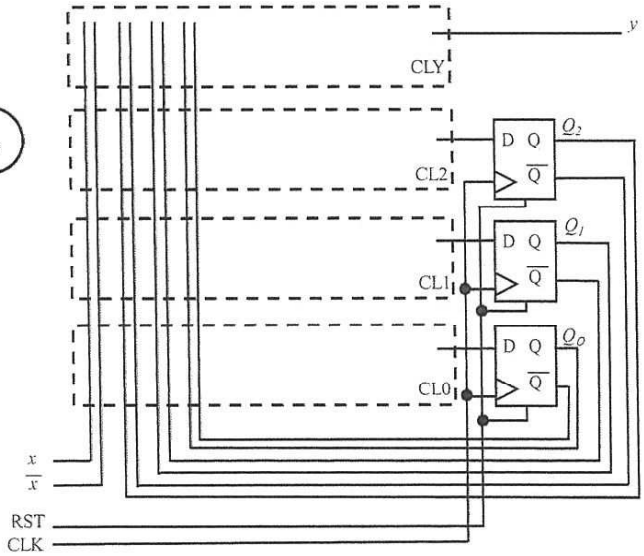


図 2

(4) 図 1 の状態遷移図から共通化できる状態をまとめ，状態数最小化（state minimization）を行う．共通化した状態を新たに S_0', S_1', S_2', S_3' とした場合の状態割り当てを表 2 に示す．A から F を埋めて表 2 を完成させよ．ただし，□ の中にはいる状態は，S_0 から S_5 のいずれかもしくは状態がないことを表す記号 (-) である．

(5) (4) の順序回路の状態を表 2 のように割り当てた場合，x, Q_1, Q_0 を変数とする y, Q_1^+, Q_0^+の最簡積和形の論理式を導出せよ．ただし，現在の状態を (Q_1, Q_0)，次状態を (Q_1^+, Q_0^+) で表すものとする．

(6) 2 入力 NAND ゲートのハードウェアコスト（hardware cost）を 2，3 入力 NAND ゲートのハードウェアコストを 3，D フリップフロップのハードウェアコストを 12 とする．状態数最小化前の (2) の順序回路と状態数最小化後の (5) の順序回路のハードウェアコストをそれぞれ求めよ．ただし，順序回路のハードウェアコストは，2 入力 NAND ゲート，3 入力 NAND ゲート，D フリップフロップのハードウェアコストの総和で求められる．

(7) 順序回路においてはその状態数が減らせても，ハードウェアコストが必ずしも減るわけではない．その理由について，組合せ回路と記憶回路（memory element）のハードウェアコストの観点から簡潔に説明せよ．

表 2

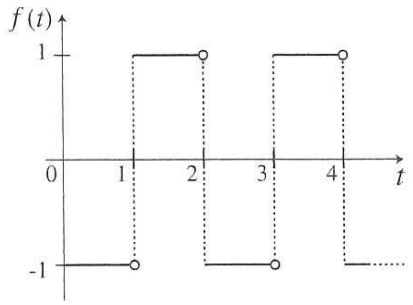
新状態	共通化された状態	状態割り当て Q_1	Q_0
S_0'	{ S_0 , A }	0	0
S_1'	{ S_1 , B }	0	1
S_2'	{ C , D }	1	0
S_3'	{ E , F }	1	1

7 【選択問題】数学解析と信号処理 （情報工学 15/15）

配点: (1-1)30 点, (1-2)30 点, (2-1)20 点, (2-2)45 点

以下の各問に答えよ.

(1) 関数 $f(t) = (-1)^{n+1} \quad \{n \leq t < n+1 \quad (n = 0, 1, 2...)\}$ は下記のように図示される．この関数 $f(t)$ について以下の小問に答えよ．ただし，関数 f は $[0, +\infty)$ で定義される．



(1-1) $\int_0^\infty e^{-st} f(t)dt$ を級数 (series) の形式で求めよ．ただし s は複素変数 (complex variable) である.

(1-2) (1-1) の結果を用いて $f(t)$ のラプラス変換 (Laplace transform)$\mathcal{L}[f(t)]$ が収束する条件 (convergence condition) と，この条件が満たされるときの $\mathcal{L}[f(t)]$ を求めよ.

(2) 次の正弦波信号 (sine wave signal) $u(t)$ について，以下の小問に答えよ．ただし，$V_m > 0$ とする.

$$u(t) = V_m \sin \frac{2\pi}{T} t$$

(2-1) $u(t)$ を全波整流 (full-wave rectification) した波形 (waveform)$v(t)$ を式で示し，図示せよ．なお，全波整流とは交流電流 (alternating current) の正・負両波とも整流 (rectification) し，流れの向きを同じにすることである.

(2-2) $v(t)$ をフーリエ級数展開 (Fourier series expansion) せよ.