令和 4 年 7 月 30 日（土）9:00～12:00

大阪大学大学院情報科学研究科

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

令和 5 年度　博士前期課程　入試問題

（A）情報工学

【注意事項】

- 問題数は必須問題 2 題（問題 1～2），選択問題 5 題（問題 3～7），合計 7 題である．
  必須問題は 2 題すべて解答すること．また，選択問題は 2 題を選択して解答すること．
- 問題用紙は表紙を含めて 13 ページである．
- 解答用紙は全部で 4 枚ある．
  1 枚目（赤色）の解答用紙には問題 1（必須問題）の解答を
  2 枚目（青色）の解答用紙には問題 2（必須問題）の解答を
  3 枚目（白色）の解答用紙には問題 3～7（選択問題）から選択した 1 題の解答を
  4 枚目（白色）の解答用紙には問題 3～7（選択問題）から選択したもう 1 題の解答を
  それぞれ記入すること．
  解答用紙は間違えると採点されないことがあるので注意すること．
- 解答用紙は 4 枚すべてを回収するので，すべての解答用紙に受験番号を記入すること．
- 解答用紙の「試験科目」の欄には解答した問題の科目名（「アルゴリズムとプログラミング」など）を記入すること．
  また，選択問題調査票には，選択した問題の番号（3～7 から二つ）に○をつけること．
- 解答欄が不足した場合は裏面を使用すること．その際，表面末尾に「裏面に続く」と明記しておくこと．解答用紙の追加は認めない．
- 解答用紙には，日本語または英語で解答すること．

1 【必須問題】アルゴリズムとプログラミング （情報工学 1/12）

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

ハッシュ関数 (hash function) を用いてキー (key) を配列 (array) に挿入 (insert) することを考える．キーは，非負整数 (non-negative integer) とする．また，キーの削除 (delete) は考えないものとする．処理が異なる二つの方式，方式 1 と方式 2 を，図 1 に示す通り ANSI-C 準拠の C 言語で記述した．方式 1 と方式 2 のキーの挿入処理は，それぞれ関数 (function) `insert1` と `insert2`，キーの探索 (search) 処理はそれぞれ関数 `search1` と `search2` である．ハッシュ関数は，関数 `hash` である．プログラムに対する入力 (input) は，図 2 に示すようなファイル `input.txt` で与える．`input.txt` の 1 行目は変数 (variable) `delta` に設定する値，2 行目は挿入するキーの個数，3 行目以降の各行は挿入するキーである．`input.txt` で与えるキーの個数は記号定数 (symbolic constant) `MAX_KEYS` で規定される値以下とし，与えられたキーに重複はないものとする．方式 1 と方式 2 について，以下の各問に答えよ．解答は全て解答用紙の太枠内に書くこと．

(1) 方式 1 について，以下の各小問に答えよ．

(1-1) `input.txt` が図 2 の内容であるとき，図 1 の 61〜62 行目の for ループが終了した時点の配列 `table1` の内容を答えよ．

(1-2) `input.txt` で指定する `delta` の値とキーによっては，図 1 の 61〜62 行目の for ループが終了しない．記号定数 `TABLE1_SIZE` 個の任意のキーの挿入に対してこの for ループが終了するような `delta` の値のうち，0 以上 10 以下のものを全て列挙せよ．

(2) 方式 2 について，以下の各小問に答えよ．

(2-1) `input.txt` が図 2 の内容であるとき，図 1 の 64〜65 行目の for ループが終了した時点の配列 `table2_1` と `table2_2` の内容を答えよ．なお，排他的論理和 (exclusive or) の計算結果については表 1 の値を用いて良い．

(2-2) `input.txt` が図 3 の内容であり，空欄 [(A)] にあるキーが指定されたとき，図 1 の 64〜65 行目の for ループが終了しない．空欄 [(A)] に当てはまる最小のキーを答えよ．

(3) `search1` と `search2` を図 4 の通りに記述した．これらの関数は，例えば，図 1 の関数 `main` 中でキー `10` を探索する場合，それぞれ，以下のように呼び出すものとする．

```
search1(table1, TABLE1_SIZE, delta, 10)
search2(table2_1, table2_2, TABLE2_SIZE, delta, 10)
```

以下の各小問に答えよ．

(3-1) `search2` は，指定されたキーが挿入されている場合はキーが挿入されている配列の要素のポインタを返し，キーが挿入されていない場合は `NULL` を返すものとする．図 4 の空欄 [(B)] 〜 [(H)] に適切な式 (expression) を埋めよ．

(3-2) `table1`，`table2_1` および `table2_2` には，それぞれ方式 1 と方式 2 で同じキーが挿入されているものとする．このときの方式 1 に対する方式 2 のキーの探索効率における利点を，二つの方式の探索時の配列への参照回数を比較しながら簡潔に説明せよ．

(次ページへ続く)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define MAX_KEYS 20
5  #define TABLE1_SIZE 20
6  #define TABLE2_SIZE 10
7  #define EMPTY -1
8
9  int* search1(int *array, int size, int delta, int key);
10 int* search2(int *array1, int *array2, int size, int delta, int key);
11
12 int hash(int x, int n) { return x % n; }
13
14 void insert1(int *array, int size, int delta, int key) {
15   int v = key;
16   int h = hash(v, size);
17   while(array[h] != EMPTY) {
18     v += delta; h = hash(v, size);
19   }
20   array[h] = key;
21 }
22
23 int insert2_internal(int *array, int key, int h) {
24   if(array[h] == EMPTY) {
25     array[h] = key;
26     return EMPTY;
27   } else {
28     int tmp = array[h]; array[h] = key;
29     return tmp;
30   }
31 }
32
33 void insert2(int *array1, int *array2, int size, int delta, int key) {
34   int x = key;
35   while(1) {
36     if((x = insert2_internal(array1, x, hash(x, size))) == EMPTY)
37       return;
38     if((x = insert2_internal(array2, x, hash(x ^ delta, size))) == EMPTY)
39       return;
40   }
41 }
42
43 int main() {
44   FILE *fp;
45   int i; int m; int delta;
46   int table1[TABLE1_SIZE];
47   int table2_1[TABLE2_SIZE]; int table2_2[TABLE2_SIZE];
48   int keys[MAX_KEYS];
49
50   for(i = 0; i < TABLE1_SIZE; ++i) table1[i] = EMPTY;
51   for(i = 0; i < TABLE2_SIZE; ++i) {
52     table2_1[i] = EMPTY; table2_2[i] = EMPTY;
53   }
54
55   fp = fopen("input.txt", "r");
56   fscanf(fp, "%d\n", &delta);
57   fscanf(fp, "%d\n", &m);
58   for(i = 0; i < m; ++i) fscanf(fp, "%d\n", &keys[i]);
59   fclose(fp);
60
61   for(i = 0; i < m; ++i)
62     insert1(table1, TABLE1_SIZE, delta, keys[i]);
63
64   for(i = 0; i < m; ++i)
65     insert2(table2_1, table2_2, TABLE2_SIZE, delta, keys[i]);
66
67   return 0;
68 }
```

図 1. ハッシュ関数を用いてキーを配列に挿入するプログラム

（次ページへ続く）

```
3
5
8
22
25
2
18
```

図 2. input.txt の例 1

```
3
6
8
22
25
2
18
(A)
```

図 3. input.txt の例 2

```
int* search1(int *array, int size, int delta, int key) {
  int v = key;
  int h = hash(v, size);
  while(array[h] != EMPTY) {
    if(array[h] == key) return &array[h];
    v += delta; h = hash(v, size);
  }
  return NULL;
}

int* search2(int *array1, int *array2, int size, int delta, int key) {
  int h = hash( (B) , size);
  if( (C) )
    return (D) ;
  h = hash( (E) , size);
  if( (F) )
    return (G) ;
  return (H) ;
}
```

図 4. 関数 search1 と search2

表 1. 排他的論理和の計算例

y	y ^ 3
0	3
1	2
2	1
3	0
4	7
5	6
6	5
7	4
8	11
9	10
10	9
11	8
12	15
13	14
14	13
15	12
16	19
17	18
18	17
19	16
20	23
21	22
22	21
23	20
24	27
25	26
26	25
27	24
28	31
29	30
30	29
31	28

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 4/12）

配点： (1) 40, (2-1) 10, (2-2) 25, (2-3) 25, (2-4) 25

次のようなページング（paging）による仮想記憶システム（virtual memory system）を想定する．

- 仮想メモリ（virtual memory）：1 [ギガバイト]（Gbyte）
- 物理メモリ（physical memory）：128 [メガバイト]（Mbyte）
- ページサイズ（page size）：4 [キロバイト]（Kbyte）

この仮想記憶システムは，ライトバック（write back）方式を用いて，物理メモリに収まらない仮想ページ（virtual page）を，仮想メモリの内容を格納するために十分な容量を有する 2 次記憶装置（secondary storage）に待避させる．この仮想記憶システムでは，アドレス（address）は 1 [バイト]（byte）ごとに付与され，仮想メモリのアドレスを指定するための仮想アドレス（virtual address）には最低 v [ビット]（bit），物理メモリのアドレスを指定するための物理アドレス（physical address）には最低 p [ビット]，ページ内オフセット（page offset）には最低 f [ビット]必要である．仮想アドレス v [ビット]のうち，下位 f [ビット]がページ内オフセット，残りの上位 $v-f$ [ビット]が仮想ページ番号（virtual page number）となっている．仮想ページは 2^n 個存在している．以下の各問に答えよ．解答は全て解答用紙の太枠線内に書くこと．

ただし，仮想記憶システム本体のプログラム（program）やアドレス変換テーブル（address translation table）が使用する物理メモリや 2 次記憶装置は，上述の仮想メモリや物理メモリ，2 次記憶装置とは別である．また，メモリ圧縮（memory compression）は使用しないものとする．1 [ギガバイト] = 2^{30} [バイト]，1 [メガバイト] = 2^{20} [バイト]，1 [キロバイト] = 2^{10} [バイト]とする．0x に続く表記は 16 進数（hexadecimal）を示す．なお，特に指定がなければ数値は 10 進数で答えること．

(1) v, p, f, n に入る値を求めよ．

(2) この仮想記憶システムにおいて，仮想アドレスは，それに含まれる仮想ページ番号により，表 1 のようなアドレス変換テーブルのエントリに対応づけられる．仮想ページが物理メモリに読み込まれているとき，仮想アドレスから物理アドレスへの変換は，仮想アドレスの下位 f [ビット]を物理アドレスの下位 f [ビット]とし，仮想ページに対応づけられた物理ページ番号（physical page number）を物理アドレスの上位 $p-f$ [ビット]として行う．

アドレス変換テーブルの列 X の各エントリは 1 [ビット]で，そのエントリに対応する仮想ページが物理メモリに読み込まれていれば 1，そうでなければ 0 である．アドレス変換テーブルの列 Y の各エントリも 1 [ビット]で，そのエントリに対応する仮想ページが新しく 2 次記憶装置から物理メモリに読み込まれると 0 にセットされ，その仮想ページに書き込みがあったときに 1 にセットされる．以下の各小問に答えよ．

（次ページへ続く）

表 1　アドレス変換テーブル

仮想ページ番号	X	Y	物理ページ番号
…	…	…	…
0xF	1	0	0x14
…	…	…	…
0xF0	1	1	0x7C
…	…	…	…
0xF00	1	1	0x2
…	…	…	…

(2-1)　物理ページ番号を表現するのに最低必要なビット数を求めよ．

(2-2)　アドレス変換テーブルが表 1 のようであるとき，仮想アドレス 0xF009 を含む仮想ページが物理メモリ上に存在している．この仮想アドレス 0xF009 に対応する物理アドレスを 16 進数で求めよ．仮想アドレス v [ビット]において，下位 f [ビット]がページ内オフセット，残りの上位 $v-f$ [ビット]が仮想ページ番号であることを用いること．

(2-3)　アドレス変換テーブルの列 Y はこの仮想記憶システムの動作の高速化にどのように貢献しているか，その仕組みを簡潔に述べよ．

(2-4)　この仮想記憶システムにおいて，ページサイズを 8 [キロバイト]に変更する．このとき f, n は変化する．このときのアドレス変換テーブル全体を格納するのに必要な容量をキロバイト単位で求めよ．ただし，仮想ページ番号はアドレス変換テーブルのサイズに含まれないものとする．

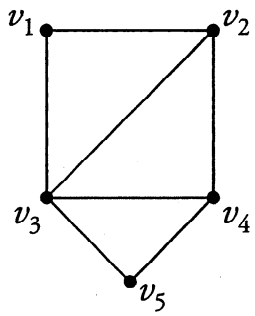
配点：(1) 10，(2-1) 10，(2-2) 15，(2-3) 30，(3-1) 10, (3-2) 25, (4) 25

グラフ（graph）$G=(V,E)$ は，n 個の頂点（vertex）の集合 $V=\{v_1,v_2,...,v_n\}$ と，頂点のペアにより定義される辺（edge）の集合 E により構成される無向グラフ（undirected graph）である．また，グラフ G は同じ頂点を結ぶ辺を持たず，かつ，任意の 2 つの頂点間を結ぶ辺は高々一つであるとする．

頂点 v_i と v_j の間に辺が存在するとき，頂点 v_i と v_j は隣接する（adjacent）と呼ぶ．次の規則で定義される (i,j) 成分 a_{ij} を持つ $n\times n$ 行列 A_G をグラフ G の隣接行列（adjacency matrix）と呼ぶ．

$$a_{ij}=\begin{cases}1 & (v_i\text{と }v_j\text{の間に辺が存在するとき})\\ 0 & (\text{その他のとき})\end{cases}$$

また，隣接する頂点の系列 $v_{x_0}\to v_{x_1}\to ...\to v_{x_h}\to ...\to v_{x_m}$ $(0\le h\le m,\ 1\le x_h\le n)$ を，長さ m の歩道（walk）と呼ぶ．歩道は同じ頂点を複数含んでも良い．例えば，下図グラフ G_1 において，$v_1\to v_2\to v_4\to v_2\to v_3$ は長さ 4 の歩道である．



G_1

以下の各問に答えよ．

(1) 上図グラフ G_1 の隣接行列 A_{G_1} を示せ．

(2) グラフ G における，頂点 v_i から v_j への長さ k $(k\ge 1)$ の歩道の総数を，$f_G(k,i,j)$ で表すこととする．

(2-1) 上図グラフ G_1 について考える．グラフ G_1 において，$f_{G_1}(3,3,2)$ の値を答えよ．

(2-2) 上図グラフ G_1 において，$\sum_{1\le k\le 3} f_{G_1}(k,4,3)$ の値を答えよ．

(2-3) グラフ G における隣接行列 A_G の k 乗を ${A_G}^k$ で表す．行列 ${A_G}^k$ の (i,j) 成分を $a_{ij}^{(k)}$ と表現すると，$a_{ij}^{(k)}=f_G(k,i,j)$ となることを証明せよ．

(3) 頂点数が n の完全グラフ（complete graph）を K_n とする．

(3-1) 完全グラフ K_n の辺の数を n を用いて示せ．

(3-2) 一般に，$n\ge 3$ のグラフ G が K_3 を含むかどうかは，隣接行列を用いて判定することができる．隣接行列 A_G の (i,j) 成分を a_{ij}，${A_G}^2$ の (i,j) 成分を b_{ij} としたときに，a_{ij} と b_{ij} を用いて，グラフ G が K_3 を含むかどうかを判定する方法を理由とともに説明せよ．

(4) $n\ge 2$ のグラフ G が連結である（connected）ことは，行列 [α] がそのいずれの成分にも 0 を持たないことを調べることで確認できる．I_n を $n\times n$ の単位行列（identity matrix）としたときに，隣接行列 A_G，I_n，n を用いて空欄 α を埋めよ．

配点：(1) 60, (2-1) 20, (2-2) 20, (3) 25

以下の各問に答えよ．

(1) 以下の図 (a)~(d) は，最終状態による受理を行う (acceptance by final state) プッシュダウンオートマトン (pushdown automaton) である．各プッシュダウンオートマトンの開始状態 (start state) には太い矢印 (thick arrow) が付与されており，最終状態は二重丸 (double circle) で表現されている．また，入力アルファベット (input alphabet) は $\{a, b\}$，スタックアルファベット (stack alphabet) は $\{0, 1, z\}$，スタックの開始記号 (start symbol) は z である．図 (a)~(d) において，遷移に付与されているラベル $(r, s)/t$ は，入力から読み出す記号が r でスタックの先頭の記号が s のときに，その s をスタックから取り去り，記号列 t を右側の記号から順にスタックに押し込むことを意味する．例えば，$(a, z)/0z$ 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．

図 (a)~(d) の各プッシュダウンオートマトンについて，受理される言語 (language) が正則表現 (regular expression) で表せるかどうかを答えよ．また，表せる場合には受理される言語の正則表現も書け．表せない場合は，受理される言語を簡潔に説明せよ．

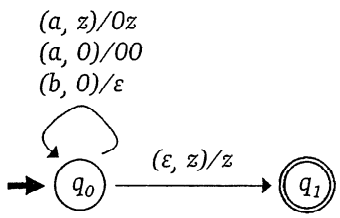


図 (a)

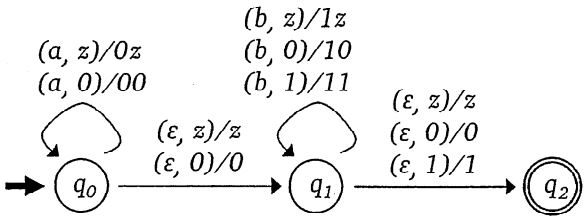


図 (b)

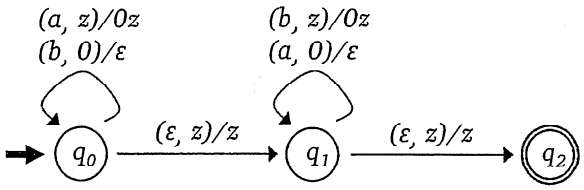


図 (c)

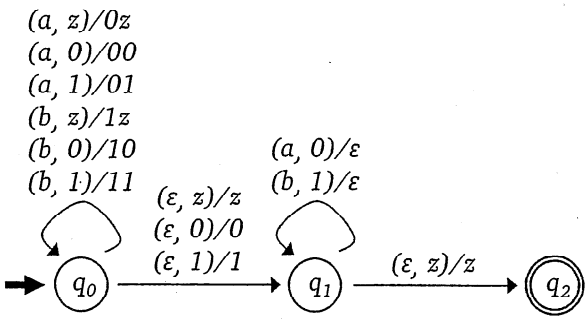


図 (d)

(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 を以下のように定める．

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

以下の各小問に答えよ．

(2-1) 文脈自由文法 G_1 はあいまい (ambiguous) である．$a + a * a$ の導出木 (derivation tree) を全て記述せよ．

（次ページに続く）

(2-2) 文脈自由文法 G_1 と同じ言語を受理するあいまいでない文脈自由文法 G_2 は，以下のように定義できる．

$G_2 = (\{S,\ X,\ Y\},\ \{+,\ *,\ a\},\ P_2,\ S)$,

$P_2 = \{S \to S + X,\ S \to$ [(e)], $X \to$ [(f)], $X \to$ [(g)], $Y \to$ [(h)] $\}$

上記定義の (e)~(h) について，適切な式を書け．

(3) アルファベット $\{(,\ ),\ [,\ ]\}$ から成る列において，バランスが取れている列とは，空列 (empty sequence) もしくは，その列の中の長さ 2 の部分列 () または [] の削除を一回以上可能な限り繰り返すと，最終的に空列が得られる列を意味する．例えば列 (([])) や列 [()]([]) はバランスが取れているが，列)([] や列 [(]) はバランスが取れていない．

以下の文脈自由文法 G_3 が全てのバランスが取れている列を生成するように，(i) にあてはまる適切な生成規則を書け．ε-規則 (ε-production) を用いても良い．生成規則の個数は 4 以下とする．

$G_3 = (\{S\},\ \{(,\ ),\ [,\ ]\},\ P_3,\ S)$,

$P_3 = \{$ [(i)] $\}$

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

TCP（Transmission Control Protocol）に関する以下の各問に答えよ．以降ではセグメント（segment）サイズ（size）を MSS [ビット]，セグメントの送信を始めてから確認応答（acknowledgement）を受け取るまでの時間を R [秒] とし，輻輳ウィンドウサイズ（congestion window size）を W [セグメント] と表記する．

(1) TCP の送信レート（transmission rate）[ビット毎秒] の最大値を，MSS，R，W を用いて表せ．

(2) TCP においてセグメント損失（segment loss）を検出する方法を 2 つ述べよ．

(3) 輻輳回避フェーズ（congestion avoidance phase）において，セグメント損失を検出した時は輻輳ウィンドウサイズ W を半減させ，セグメント損失を検出しなかった時は輻輳ウィンドウサイズ W を R 毎に 1 増加させるものとする．輻輳ウィンドウサイズ W が W_{L} の時にセグメント損失を検出したとして以下の各小問に答えよ．なお，スロースタートフェーズ（slow start phase）への遷移は考えなくて良い．

(3-1) セグメント損失の検出により輻輳ウィンドウサイズを $W_{\mathrm{L}}/2$ としてから，輻輳ウィンドウサイズが再び W_{L} となるまでの時間 τ を求めよ．ただし W_{L} は偶数（even number）と仮定する．また，輻輳ウィンドウサイズが再び W_{L} となるまでセグメント損失は発生しないものとする．

(3-2) (3-1) で求めた τ の間に送信されるセグメント数が

$$\frac{3}{8}(W_{\mathrm{L}})^2$$

となることを示せ．

(3-3) TCP の輻輳回避フェーズにおける平均送信レートが

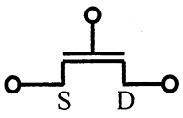
$$\sqrt{\frac{3}{2}} \times \frac{\mathrm{MSS}}{\mathrm{R} \times \sqrt{p}}$$

で近似されることを，(3-1) および (3-2) を用いて示せ．ただし p はセグメント損失率（segment loss rate）であり，セグメント損失を検出してから，次にセグメント損失を検出するまでの間に送信されるセグメント数の期待値は $1/p$ とする．

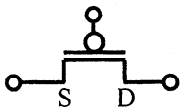
配点：(1-1) 20, (1-2) 20, (1-3) 20, (2-1) 30, (2-2) 35

以下の各問に答えよ．なお，図中の記号は以下の凡例に従うとする．

凡例



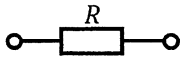
Sをソース側, Dをドレイン側とする
n型MOS (Metal Oxide Semiconductor)トランジスタ



Sをソース側, Dをドレイン側とする
p型MOSトランジスタ

電源(power supply)

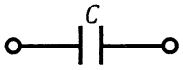

接地(earth connection)



抵抗値(resistance) Rを持つ抵抗


交流電圧源(alternating current voltage source)

インダクタンス(inductance) Lを持つインダクタ



静電容量(capacitance) Cを持つキャパシタ


スイッチ

(1) CMOS (Complementary MOS)に基づくディジタル回路(以後 CMOS 回路と呼ぶ)に関する以下の各小問に答えよ．電源の電圧値は 1 V である．以下を仮定して答えよ．

仮定 1: MOS トランジスタの特性について以下が成立する．ゲート，ソース，ドレインは端子名である．

- ゲート-ソース間の電圧の絶対値が 0.1 V 未満のとき，ドレイン-ソース間を流れる電流がゼロになる．
- ゲート-ソース間の電圧の絶対値が 0.1 V 以上のとき，ドレイン-ソース間の抵抗値が 10 kΩになる．
- ゲート-ソース間およびゲート-ドレイン間を流れる電流はゼロである．
- ゲート-ドレイン間の静電容量はゼロである．

仮定 2: 配線の静電容量や抵抗値はともにゼロである．

(1-1) 図 1 に示す論理ゲートを実現する CMOS 回路を図示せよ．n 型 MOS トランジスタ, p 型 MOS トランジスタの使用個数をともに 3 として図示すること．

（次のページに続く）

(1-2) 図 2 の CMOS 回路において, A, B の電圧をそれぞれV_A, V_Bとする. V_A, V_B に定電圧を CMOS 回路に与えて定常状態(steady state)になったときの Y の電圧を, (V_A, V_B) = (0 V, 0 V), (0 V, 1 V), (1 V, 0 V), (1 V, 1 V) の 4 通りの場合に対してそれぞれ書け.

(1-3) 図 2 の CMOS 回路において, A, B の電圧をそれぞれV_A, V_Bとする. 出力 Y の負荷容量(load capacitance)が 0.02 pF となるようキャパシタを Y と接地の間に取り付け, V_AとV_Bをともに0 Vにした. CMOS 回路が定常状態になった後, V_AとV_Bが同時に0 Vから1 Vに変化するステップ電圧(step voltage)を与えた. V_A, V_Bがともに0 Vから1 Vに変化してから Y の電圧が 0.5 Vになるまでの時間を書け. $\log_e 2 = 0.69$ とせよ.

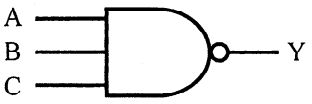


図 1

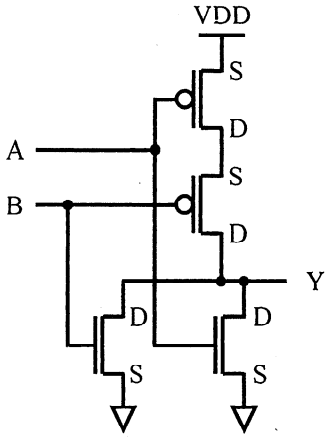


図 2

(2) 図 3 に示す回路について, 以下の各小問に答えよ. スイッチ S1, S2 は時刻$t < 0$で開いており, 交流電圧源V_1, V_2はともに実効値E_0および正の角周波数(angular frequency) ω_sで発振している.

(2-1) 時刻$t = 0$で S1 のみを閉じた後, 回路が定常状態となった. 定常状態において破線で囲まれた負荷(electrical load)が消費する有効電力(effective power)をω_s, E_0, L, R, Cの中から必要な物を用いて式で示せ. また, 定常状態で負荷の無効電力(reactive power)がゼロとなるω_sをE_0, L, R, Cの中から必要な物を用いて式で示せ.

(2-2) 時刻$t = 0$で S1 と S2 を閉じた後, 回路が定常状態となった. 定常状態において, キャパシタCに流れる電流i_CとV_1の位相(phase)の差がゼロとなるω_sをE_0, L, R, Cの中から必要な物を用いて式で示せ.

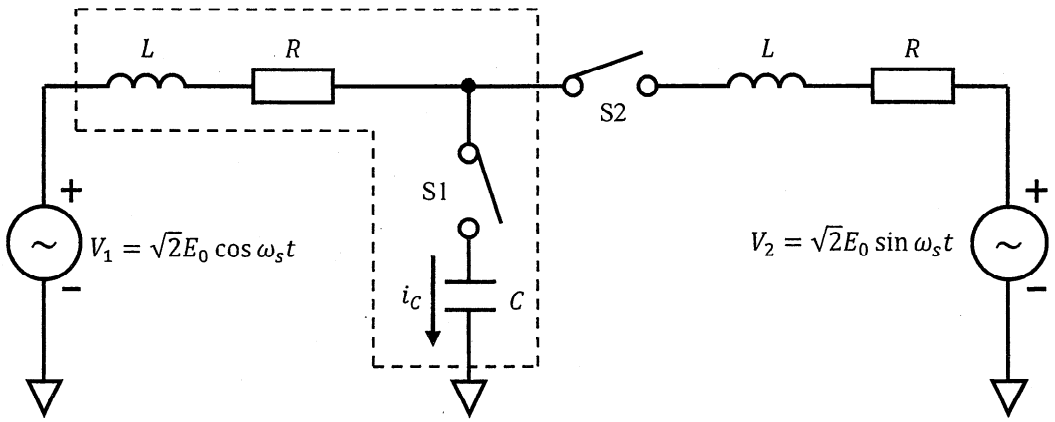


図 3

7 【選択問題】数学解析と信号処理 （情報工学 12/12）

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

以下の各問に答えよ．ただし導出の過程も示すこと．

(1) 式 [1] で実現されるデジタルフィルタ (digital filter) について各小問に答えよ．ただし $x[n]$ は離散時間線形時不変システム（discrete-time linear time-invariant system）の入力 (input)，$y[n]$ はシステムの出力 (output)，n は整数 (integer) とする．

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2]) \quad [1]$$

(1-1) デジタルフィルタの周波数伝達関数 (frequency transfer function)$H(\omega)$ を，実関数 (real function) である $A(\omega)$ と $\theta(\omega)$ を用いて式 [2] の通りに記述する．式 [1] に示すフィルタの振幅特性 (amplitude characteristics) と位相特性 (phase characteristics) をそれぞれ $|A(\omega)|$ と $\theta(\omega)$ と定義するとき，それぞれが式 [3] と式 [4] で表せることを，z 変換 (z-transform) を用いて証明せよ．ただし，ω は角周波数 (angular frequency) であり，j は虚数単位 (imaginary unit) である．

$$H(\omega) = A(\omega)e^{j\theta(\omega)} \quad [2]$$

$$|A(\omega)| = \frac{1}{3}|2\cos(\omega) + 1| \quad [3]$$

$$\theta(\omega) = -\omega \quad [4]$$

(1-2) 式 [1] に示すフィルタの振幅特性の概形を，横軸を ω とし $\{\omega \in \mathbb{R} \mid -\pi \leq \omega \leq \pi\}$ の範囲で図示せよ．

(2) 問 (1) のフィルタを一般化し，式 [5] に示す N 点平均を計算するフィルタを考える．各小問に答えよ．

$$y[n] = \frac{1}{N}(x[n] + x[n-1] + x[n-2] + \cdots + x[n-N+1]) \quad [5]$$

(2-1) 式 [5] に示すフィルタの振幅特性が $\omega \neq 0$ のとき式 [6] の通りになることを，z 変換を用いて証明せよ．また $\omega = 0$ のときの $|A(\omega)|$ を求めよ．

$$|A(\omega)| = \frac{1}{N}\left|\frac{\sin(\frac{\omega N}{2})}{\sin(\frac{\omega}{2})}\right| \quad [6]$$

(2-2) $N = 3$ のとき，式 [6] の振幅特性が式 [3] の通りになることを証明せよ．

(2-3) $\omega > 0$ のとき式 [5] に示すフィルタのゲインが 0 となる最小の ω を N を用いて表せ．

(2-4) 式 [5] に示すフィルタは，移動平均フィルタ (moving average filter) として高周波ノイズ (high frequency noise) を低減するために用いられることが知られている．式 [5] に示すフィルタは，N が大きくなるにつれ，どのようにノイズ低減の挙動が変化するか，小問 (2-3) の結果を用いて記述せよ．