令和７年８月２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８年度　博士前期課程　入試問題

（A）情報工学

【注意事項】

- 解答開始の合図があるまで中を見てはいけない.
- 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.
- 問題用紙は表紙と白紙を除いて 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.
  解答用紙を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解答用紙の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. ただし，解答用紙に枠の指定がある場合は，その枠内に解答を記入すること.
- 解答用紙には，日本語または英語で解答すること. これ以外の言語で記述されていた場合，採点されないことがある.

1 【必須問題】アルゴリズムとプログラミング （情報工学 1/13）

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

図1はANSI Cで記述されたプログラム (program) である．このプログラムは `data.txt` を読み込んで (read) 処理を行い，その結果を出力 (output) する．`data.txt` には，1行目に n, s_1, s_2 が記述されている．n, s_1, s_2 は整数 (integer) であり，$n \geq 1$ および $s_1 < s_2$ を満たす．2行目以降は n 件の識別子 (ID) と得点 (score) の対 (pair) が各行 (each line) に記述されている．識別子および得点は整数とする．図1のプログラムに関する以下の各問に答えよ．

(1) 以下の各小問に答えよ．

(1-1) 関数 (function) `funcA` において，14行目の条件式 (conditional expression) `students[j].score > students[j + 1].score` が評価 (evaluate) される最大 (maximum) の回数を，引数 (argument) `n` を用いて示せ．

(1-2) 39行目で呼び出される関数 `funcB` において，28行目の条件式 `students[mid].score < target` が評価される最大の回数を，引数 `n` を用いて示せ．

(1-3) 関数 `funcB` は，第一引数として与えられる配列 (array) を扱うアルゴリズムを実装している．このアルゴリズムの最悪時間計算量 (worst-case time complexity) のオーダ表記 (order notation) として最も適しているものを，下記の選択肢から一つ選択せよ．ただし，配列のサイズを N とする．

$O(1)$ $O(\log N)$ $O(N)$ $O(N \log N)$ $O(N^2)$ $O(N^3)$ $O(2^N)$ $O(N!)$

(2) `data.txt` が図2の内容である場合，図1のプログラムが開始 (start) してから終了 (finish) するまでに出力される内容を記述せよ．

(3) 関数 `funcA` を改変することにより，14行目の条件式 `students[j].score > students[j + 1].score` が評価される回数を減らすことを考える．ただし，関数 `funcA` の終了時における配列 `students` の要素 (element) の並びは，改変の前後で変わらないようにせよ．以下の各小問に答えよ．

(3-1) `data.txt` が図2の内容である場合，以下の三つの文 (statement) を追加することで14行目の条件式が評価される回数を減らすことができる．図1のどの行の直後に追加すればよいか，それぞれの文に対して行番号を一つずつ答えよ．ただし，三つの文が追加される位置はそれぞれ異なる．また，文の追加によって行番号が変わることは考えなくてよい．

- `int flag = 0;`
- `if(flag == 0) break;`
- `flag = 1;`

(3-2) 小問 (3-1) で完成させた関数 `funcA` は，第一引数として与えられる配列を扱うアルゴリズムを実装している．このアルゴリズムの最悪時間計算量のオーダ表記として最も適しているものを，下記の選択肢から一つ選択せよ．ただし，配列のサイズを N とする．

$O(1)$ $O(\log N)$ $O(N)$ $O(N \log N)$ $O(N^2)$ $O(N^3)$ $O(2^N)$ $O(N!)$

（次ページへ続く）

```c
#include <stdio.h>
#include <stdlib.h>

typedef struct{
  int id;
  int score;
} student_type;

void funcA(student_type students[], int n)
{
  int i, j;
  for(i = 0; i < n - 1; i++){
    for(j = 0; j < n - 1 - i; j++){
      if(students[j].score > students[j + 1].score){
        student_type tmp = students[j];
        students[j] = students[j + 1];
        students[j + 1] = tmp;
      }
    }
  }
}

int funcB(student_type students[], int n, int target)
{
  int left = 0, right = n;
  while(left < right){
    int mid = (left + right) / 2;
    if(students[mid].score < target)
      left = mid + 1;
    else
      right = mid;
  }
  return left;
}

void funcC(student_type students[], int n, int s1, int s2)
{
  int i;
  int lower = funcB(students, n, s1);
  int upper = funcB(students, n, s2);

  for(i = lower; i < upper; i++)
    printf("%d %d\n", students[i].id, students[i].score);
}

int main(void)
{
  int n, s1, s2, i;
  student_type *students;
  FILE *fp;

  fp = fopen("data.txt", "r");
  fscanf(fp, "%d %d %d\n", &n, &s1, &s2);
  students = malloc(sizeof(student_type) * n);
  for(i = 0; i < n; i++)
    fscanf(fp, "%d %d\n", &students[i].id, &students[i].score);
  fclose(fp);

  funcA(students, n);
  funcC(students, n, s1, s2);

  return 0;
}
```

図 1. プログラム

```
10 70 80
1006 70
1002 60
1008 69
1005 80
1001 78
1009 79
1004 100
1007 79
1010 81
1003 85
```

図 2. data.txt 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 3/13）

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

(1) 計算機 (computer) における数の表現 (representation) と算術演算 (arithmetic operation) に関する以下の各小問に答えよ.

(1-1) $42+(-5)$ の加算を 8 [ビット (bit)] の 2 の補数表現 (two's complement representation) で行う過程を示せ．最上位ビット (the most significant bit) からの桁上げ (carry) 出力の扱いも記すこと.

(1-2) N [ビット] の2 の補数表現を用いて $4+5$ を計算する．正しく計算するために必要な N の最小値 $N_{\min}$ を示せ.

(1-3) 符号部 (sign part) を 1 [ビット]，指数部 (exponent part) を 5 [ビット]，仮数部 (mantissa part) を 10 [ビット] でそれぞれ表現する IEEE754 半精度浮動小数点表現 (half precision floating-point representation) を考える．指数部のバイアス (bias) は 15 とする．このとき，以下の (1-3-1)，(1-3-2) に答えよ.

(1-3-1) 5.25 をこの浮動小数点表現によって表した際の，符号部，指数部，仮数部をそれぞれビット列で答えよ.

(1-3-2) この浮動小数点表現によって e^{16}（e は自然対数の底 (base of natural logarithm)）を表現しようとするとオーバーフロー (overflow) が発生することを証明せよ．なお，$2<e<3$ を用いて良い.

(次ページへ続く)

(2)　プロセス（process）のスケジューリング（scheduling）に関する以下の各小問に答えよ．

(2-1)　単一プロセッサ（single processor）のマルチタスク（multitask）環境において，プロセス P1～P4 を FCFS（First Come First Service; 到着順），RR（Round Robin; ラウンドロビン），SJF（Shortest Job First; 最短要求時間順），SRT（Shortest Remaining Time First; 最小残余時間順）の 4 種類のスケジューリング方式で処理した結果の違いを考える．

図 1 はプロセス P1～P4 をあるスケジューリング方式によって処理した結果であり，○はプロセスが生成された時刻，●は完了した時刻を示す．太い線（thick line）はプロセッサが割り当てられている時間，細い線（thin line）は割り当てられていない時間を示す．

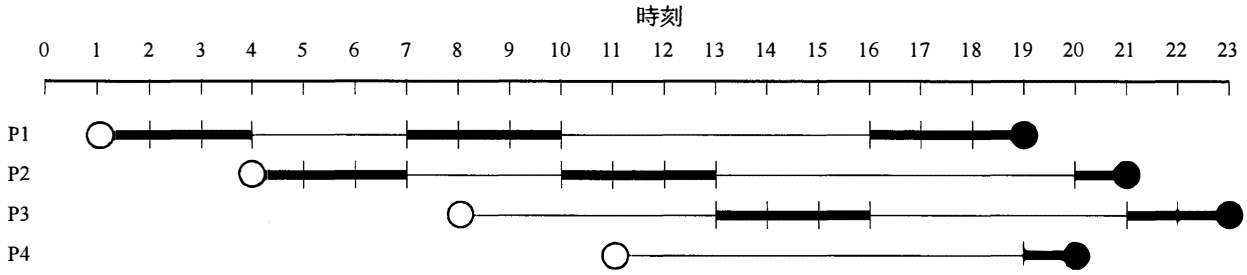

図 1．プロセス P1～P4 のスケジューリング結果

また，以下の点を前提とする．

- 実行可能キュー（ready queue）は一つのみで，時刻 0 でプロセッサと実行可能キューは空である．
- ある時刻に生成あるいはプリエンプション (preemption) されたプロセスは，ただちに実行可能キューに到着する．プロセッサが空いたとき，ただちに実行可能キューからプロセスが一つ取り出されてディスパッチ（dispatch）される．
- プロセッサによる処理は最小単位を 1 とする離散的な時刻（discrete time）で進行し，プロセスの切り替えにかかる時間は無視できる．プロセスの生成や完了，切り替えもこの離散的な時刻でのみ発生する．
- プロセス P1～P4 の生成時刻および各プロセスが完了するまでに必要な処理時間は，いずれのスケジューリング方式においても共通である．
- FCFS と SJF は非プリエンプティブ（non-preemptive）であり，RR と SRT はプリエンプティブ（preemptive）である．
- ターンアラウンド時間（turnaround time）はプロセスが生成されてから完了するまでの時間，応答時間（response time）はプロセスが到着してから初めてプロセッサが割り当てられるまでの時間とする．

以下の (2-1-1)～(2-1-4) に答えよ．

(2-1-1)　図 1 のスケジューリング結果は，FCFS，RR，SJF，SRT のうち，どの方式によるものか，最も適切なものを答えよ．

(2-1-2)　図 1 における P2 と P4 のターンアラウンド時間と応答時間を示せ．なお，プロセス P1 のターンアラウンド時間は 18 で応答時間は 0，プロセス P3 のターンアラウンド時間は 15 で応答時間は 5 である．

(2-1-3)　プロセス P1～P4 の平均ターンアラウンド時間が最も短くなるのは，FCFS，RR，SJF，SRT のどの方式か答えよ．またその平均ターンアラウンド時間を示せ．

(2-1-4)　プロセス P1～P4 の平均応答時間が最も短くなるのは，FCFS，RR，SJF，SRT のどの方式か答えよ．またその平均応答時間を示せ．

(2-2)　スケジューリングにおいては，実行可能キュー内のプロセスに優先度 (priority) を付与する方式が利用されることも多い．この方式において発生しやすい問題として最も適切なものを，以下の選択肢 **(A)～(D)** から一つ選べ．

(A) きわどい部分（critical section）　**(B)** ミューテックス（mutex）
(C) 飢餓（starvation）　**(D)** デッドロック（deadlock）

(2-3)　固定優先度（fixed-priority）スケジューリングは，実行中にプロセスの優先度が変化しないため，処理の順序や応答時間を予測しやすいという特徴がある．この特徴によるメリットを 2，3 行で説明せよ．

3 【選択問題】離散構造 （情報工学 5/13）

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

正の整数 (positive integer) n に対して，$I(n) = \{1, 2, 3, \ldots, n\}$ と定める．また，任意の有限集合 (finite set) S に対し，そのべき集合を 2^S で表す．任意の有限集合 S に対して，S に含まれる元 (element) の個数を S のサイズと呼び，$|S|$ で表す．S, T を 2 つの集合とするとき，$S \setminus T$ を，S には含まれるが T には含まれない元すべてからなる集合とする．S を有限集合，$\preceq$ を S 上の部分順序関係 (partial order relation) とする．S 中の二つの元 $s, s' \in S$ について，$s \preceq s'$ または $s' \preceq s$ が成り立つとき，s と s' は部分順序集合 $(S, \preceq)$ において比較可能 (comparable) であると呼び，$s \preceq s'$ と $s' \preceq s$ のいずれも成り立たないとき，s と s' は比較不可能 (incomparable) であると呼ぶ．部分集合 (subset) $X \subseteq S$ において，X 中の任意の二つの元が比較可能であるとき，X を $(S, \preceq)$ の鎖 (chain) と呼ぶ．また，部分集合 $Y \subseteq S$ において，Y 中の任意の相異なる二つの元が比較不可能であるとき，Y を $(S, \preceq)$ の反鎖 (antichain) と呼ぶ．以下の各問に答えよ．

(1) 部分集合族 $\mathcal{F} \subseteq 2^{I(7)}$ を以下の通りに定める．

$$\mathcal{F} = \{\{1,2\}, \{2,3\}, \{1,2,3\}, \{1,2,4\}, \{1,2,5\}, \{1,2,5,6\}, \{1,2,5,7\}, \{1,2,3,4,5,6,7\}\}.$$

以下の各小問に答えよ．

(1-1) $(\mathcal{F}, \subseteq)$ の極大元 (maximal element)，極小元 (minimal element) をそれぞれすべて列挙せよ．

(1-2) $(\mathcal{F}, \subseteq)$ の鎖でサイズが 4 のものを一つ答えよ．

(1-3) $(\mathcal{F}, \subseteq)$ の反鎖でサイズが最大のものを一つ答えよ．

(2) $(S, \preceq)$ を有限部分順序集合とする．$(S, \preceq)$ の鎖 X について，$X \cup \{s\}$ が鎖となるような $s \in S \setminus X$ が存在しないならば，X を極大鎖と呼ぶ．このとき，以下の各小問に答えよ．

(2-1) 部分順序集合 $(2^{I(4)}, \subseteq)$ の極大鎖を一つ答えよ．

(2-2) $\mathcal{C}$ を部分順序集合 $(2^{I(n)}, \subseteq)$ の任意の鎖とすると，$(\mathcal{C}, \subseteq)$ は定義より全順序集合 (totally ordered set) となる．$\mathcal{C}$ 中のすべての元を全順序関係 $\subseteq$ に従い整列した列を $(C_1, C_2, \cdots, C_{|\mathcal{C}|})$ とする．ただし，$C_1 \subseteq C_2 \subseteq \cdots \subseteq C_{|\mathcal{C}|}$ が成り立つものとする．このとき，$\mathcal{C}$ について以下の三つの性質がすべて成立するとき，かつそのときに限り $\mathcal{C}$ は極大鎖となることを示せ．

- (性質 1) $|C_1| = 0$,
- (性質 2) $|C_{|\mathcal{C}|}| = n$,
- (性質 3) 任意の i $(1 \leq i \leq |\mathcal{C}| - 1)$ について，$|C_{i+1}| = |C_i| + 1$.

(2-3) 部分順序集合 $(2^{I(n)}, \subseteq)$ の異なる極大鎖の数はいくつになるか．理由とともに答えよ．

(2-4) k を $0 \leq k \leq n$ を満たす整数とし，$C' \subseteq I(n)$ をサイズが k の任意の集合とする．部分順序集合 $(2^{I(n)}, \subseteq)$ の極大鎖のうち，C' を元として含むものの数はいくつになるか答えよ．

配点: (1-1) 10, (1-2) 10, (2) 45, (3-1) 20, (3-2) 10, (3-3) 30

有限オートマトン (FA: finite automaton) を $(Q, \Sigma, \delta, q, F)$ で表す．ここで，Q は状態 (state) の有限集合，Σ は入力記号 (input symbol) の有限集合であるアルファベット (alphabet)，δ は遷移関数 (transition function)，$q \in Q$ は開始状態 (initial state)，$F \subseteq Q$ は受理状態 (accepting state) の集合である．決定性有限オートマトン (DFA: deterministic finite automaton) の遷移関数は $\delta : Q \times \Sigma \to Q$，非決定性有限オートマトン (NFA: non-deterministic finite automaton) の遷移関数は $\delta : Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ である．ただし，ε は空語 (empty word)，$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$，$\mathcal{P}(Q)$ は Q のべき集合 (power set) である．ある FA M が受理 (accept) する全ての語 (word) の集合を，M が認識 (recognize) する言語 (language) とし，$L(M)$ で表す．FA の状態遷移図 (state transition diagram) は，開始状態には太い矢印 (thick arrow) を付与し，受理状態は二重丸 (double circle) で表す．また，ε による遷移 (transition) である ε-遷移 (ε-transition) は，ラベル ε で表す．以下の各問に答えよ．

(1) 図 1 と図 2 は，それぞれアルファベット $\{0, 1\}$ 上の NFA N_1 と N_2 の状態遷移図である．以下の各小問に答えよ．

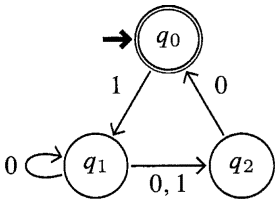

図 1 N_1

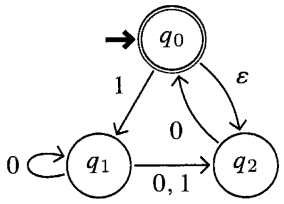

図 2 N_2

(1-1) N_1 と N_2 それぞれが受理する語のうち，長さが 4 以下のものを全て示せ．

(1-2) D を $L(N_2) = L(D)$ を満たし，かつ状態数が最小の DFA とする．D の状態遷移図を示せ．D の状態と N_2 の状態の対応関係も記述すること．ただし，D の全ての状態と入力記号の組に対して遷移を定義すること．

(2) DFA M_1 と M_2 を，それぞれ $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ と $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ とする．このとき，$L(M_1) \cup L(M_2)$ を認識する DFA $M = (Q, \Sigma, \delta, q, F)$ を構成したい．空欄 (a)〜(f) を埋めて定義を完成させよ．

$$Q = \{ \boxed{\text{(a)}} \mid r_1 \in Q_1, r_2 \in Q_2 \}$$

$$\delta(\boxed{\text{(b)}}, a) = \boxed{\text{(c)}} \qquad (\forall r_1 \in Q_1, \forall r_2 \in Q_2, \forall a \in \Sigma)$$

$$q = \boxed{\text{(d)}}$$

$$F = \{ \boxed{\text{(e)}} \mid \boxed{\text{(f)}} \}$$

(3) NFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ に対して，$q_0 \notin Q_1$ として，NFA $\mathcal{F}(M_1)$ を以下の通りに定義する．

$$\mathcal{F}(M_1) = (Q_1 \cup \{q_0\}, \Sigma, \delta_f, q_0, F_1 \cup \{q_0\})$$

（次ページへ続く）

ここで，全ての $r \in Q_1 \cup \{q_0\}$，$a \in \Sigma_\varepsilon$ について，δ_f を以下の通りに定義する．

$$\delta_f(r,a) = \begin{cases} \{q_1\} & \text{if } r = q_0 \text{ and } a = \varepsilon \\ \emptyset & \text{if } r = q_0 \text{ and } a \neq \varepsilon \\ \delta_1(r,a) \cup \{q_0\} & \text{if } r \in F_1 \text{ and } a = \varepsilon \\ \delta_1(r,a) & \text{otherwise} \end{cases}$$

ただし，$\emptyset$ は空集合 (empty set) である．

また，$Q_1 \cap Q_2 = \emptyset$ を満たす NFA M_1 と $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ に対して，NFA $\mathcal{G}(M_1, M_2)$ を以下の通りに定義する．

$$\mathcal{G}(M_1, M_2) = (Q_1 \cup Q_2, \Sigma, \delta_g, q_1, F_2)$$

ここで，全ての $r \in Q_1 \cup Q_2$，$a \in \Sigma_\varepsilon$ について，δ_g を以下の通りに定義する．

$$\delta_g(r,a) = \begin{cases} \delta_2(r,a) & \text{if } r \in Q_2 \\ \delta_1(r,a) \cup \{q_2\} & \text{if } r \in F_1 \text{ and } a = \varepsilon \\ \delta_1(r,a) & \text{otherwise} \end{cases}$$

以下の各小問に答えよ．

(3-1)　図 3 と図 4 の状態遷移図で定義される NFA N_3 と N_4 それぞれが認識する言語を表す正則（正規）表現 (regular expression) を (A)〜(P) の中から選べ．ただし，+ は和集合 (union) 演算子，* はスター (star) 演算子である．

(3-2)　$\mathcal{F}(N_4)$ の状態遷移図を示せ．ただし，初期状態を q_0 とし，q_0 と s_0〜s_3 を用いて各状態にラベルを付与せよ．

(3-3)　以下の 3 つの NFA (a)〜(c) それぞれが認識する言語と等価な正則表現を (A)〜(P) の中から選べ．また，選択した理由を，$\mathcal{F}$ と $\mathcal{G}$ の定義の観点から簡潔に説明せよ．

(a) $\mathcal{F}(N_3)$　　(b) $\mathcal{G}(N_4, \mathcal{F}(N_4))$　　(c) $\mathcal{F}(\mathcal{G}(N_4, N_3))$

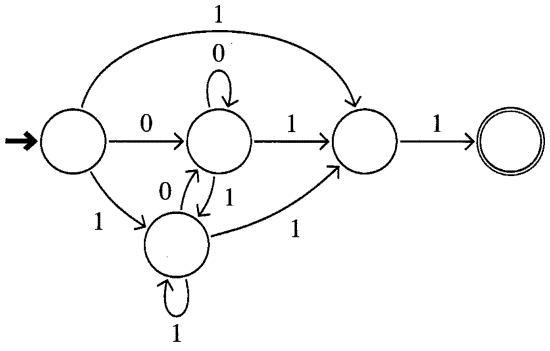

図3　N_3

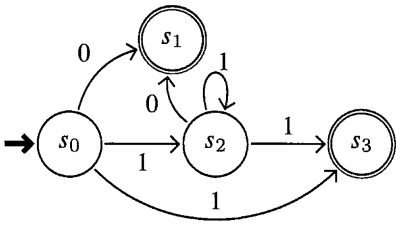

図4　N_4

(A) ε	(B) 1*(0+1)	(C) 1(0+1)*01	(D) (0+1)*11
(E) (0+1)*01	(F) (0+1)*(0+1)(0+1)	(G) (0+1)*(00+10+01)	(H) (0+1)*00(0+1)*
(I) ((0+1)*11)*	(J) ((0+1)*01)*	(K) (0+1)*00(0+1)*11	(L) (0+1)(11*0)*11
(M) (1*(0+1)(0+1)*11)*	(N) 1*(0+1)(0+1)*11	(O) ((0+1)*111*(0+1))*	(P) 1*(0+1)(1*(0+1))*

5 【選択問題】ネットワーク

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

ルーティングプロトコル (routing protocol) に関する以下の各問に答えよ．

(1) 図 1 に示すノード (node) A，B，C，D を接続したネットワーク 1 において，以下のルーティングプロトコル 1 を動作させている場合を考える．

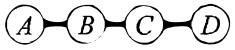

図 1　ネットワーク 1

ルーティングプロトコル 1

ネットワーク内のノードの集合 (set) を $\mathcal{N}$，ノード $x \in \mathcal{N}$ の隣接ノード (adjacent node) の集合を $\mathcal{N}_x$ とする．また，$\mathcal{N}$ からノード $x \in \mathcal{N}$ を取り除いた集合を $\mathcal{N} \setminus \{x\}$ と表す．

各ノード $x \in \mathcal{N}$ は，隣接ノードの集合 $\mathcal{N}_x$ とルーティングテーブル (routing table) を保持し，ルーティングテーブルに，各目的地 (destination) $z \in \mathcal{N} \setminus \{x\}$ に対して以下の情報を記録している．

- $d_x(z)$： ノード x から目的地 z までのホップ数 (number of hops).
- $n_x(z)$： ノード x から目的地 z への経路 (route) 上の次ホップノード (next hop node) .

ただし，ホップ数は目的地に到達するまでに経由するリンク (link) の数とし，$d_x(z)$ は M を上限とする．なお，$M > |\mathcal{N}| + 1$ とする．また，$d_x(z)$ が M となった目的地 z は到達不能 (unreachable) とみなし，$n_x(z)$ に “–”と記録するものとする．

各ノード $x \in \mathcal{N}$ は時刻同期し，同一の制御周期 (control cycle) において以下の (a)〜(c) の手順を完了する．

(a) ルーティング情報 (routing information) の送受信： ルーティング情報としてすべての目的地 $z \in \mathcal{N} \setminus \{x\}$ と対応する $d_x(z)$ をすべての隣接ノード $y \in \mathcal{N}_x$ に送信する．

(b) 隣接ノードの集合の更新： 隣接ノード $y \in \mathcal{N}_x$ からのルーティング情報をしばらく待ち，受信できなかった場合，隣接ノード y とのリンクの故障を検知し，隣接ノード y を隣接ノードの集合 $\mathcal{N}_x$ から取り除く．

(c) ルーティングテーブル更新： ルーティングテーブル中のすべての目的地 $z \in \mathcal{N} \setminus \{x\}$ に関する情報を隣接ノード $y \in \mathcal{N}_x$ から受信した $d_y(z)$ を用いて以下のように更新する．

- $z \in \mathcal{N}_x$ の場合：$d_x(z) \leftarrow 1$, $n_x(z) \leftarrow z$
- $z \notin \mathcal{N}_x$ かつ $\min_{y \in \mathcal{N}_x} \left(d_y(z) + 1\right) < M$ の場合：
  $d_x(z) \leftarrow \min_{y \in \mathcal{N}_x} \left(d_y(z) + 1\right)$, $n_x(z) \leftarrow \arg\min_{y \in \mathcal{N}_x} \left(d_y(z) + 1\right)$
- それ以外の場合：$d_x(z) \leftarrow M$, $n_x(z) \leftarrow$ “ – ”

ただし，$\arg\min_{y \in \mathcal{N}_x} \left(d_y(z) + 1\right)$ は，$d_y(z) + 1$ が最小となる $y \in \mathcal{N}_x$ を示す．

その後，次の制御周期を待つ．

（次ページへ続く）

i 回目の制御周期におけるルーティングテーブル更新後のノード A，B，C のルーティングテーブルを図2に示す．その後，$i+1$ 回目の制御周期のルーティングテーブル更新より前に，ノード C がノード D とのリンクの故障を検知した．以下の各小問に答えよ．

(1-1) $i+1$ 回目の制御周期のルーティングテーブル更新後，$i+2$ 回目の制御周期のルーティングテーブル更新後のそれぞれについて，ノード A，B，C それぞれのルーティングテーブルを図2にならって示せ．

(1-2) ノード D 以外のすべてのノードにおいて，ルーティングテーブルに記録されたノード D までのホップ数が M となり，ノード D が初めて到達不能とみなされるのは何回目の制御周期か．i を用いて答えよ．また，ノード D が初めて到達不能とみなされるのが，その制御周期になる理由を説明せよ．

ノード A

目的地 z	$d_A(z)$	$n_A(z)$
B	1	B
C	2	B
D	3	B

ノード B

目的地 z	$d_B(z)$	$n_B(z)$
A	1	A
C	1	C
D	2	C

ノード C

目的地 z	$d_C(z)$	$n_C(z)$
A	2	B
B	1	B
D	1	D

図2 i 回目の制御周期におけるルーティングテーブル更新後のノード A，B，C のルーティングテーブル

(2) リンクステート (link-state) 型のルーティングプロトコルを動作させているネットワークを考える．以下の各小問に答えよ．

(2-1) 図1のネットワーク1においてリンクステート型のルーティングプロトコルを動作させている．このネットワークにおいて，ノード C がノード D とのリンクの故障を検知してから，ネットワーク内のノード D を除くすべてのノードのルーティングテーブルでノード D が到達不能とみなされるまでの，ノード A，B，C の動作を2，3行で説明せよ．

(2-2) リンクステート型のルーティングプロトコルを動作させているネットワークにおいて，リンクの故障発生後，ルーティングプロトコルに従ってノード間で交換する情報を，一部のノードが受信できていない場合について考える．この場合，各ノードが更新したルーティングテーブルにもとづいてパケット (packet) を転送すると，故障したリンクを経由せずに目的地まで到達できる経路が存在するにも関わらず，目的地まで到達できないパケットが生じることがある．なぜ目的地まで到達できないパケットが生じうるか．その理由を，ノード間で交換する情報を受信できていないノードが存在するときと存在しないときのパケットが転送される経路を対比する形で，説明せよ．

配点：(1) 10，(2) 30，(3) 50，(4) 15，(5) 20

図 1 に示すデータパス (dat apath) は 2 個の 8 進アップカウンタ (octal up count er)Count er1，Count er2 と 8 ビット 2 進数データ (8-bit binary dat a) を 8 個格納可能なメモリ (memory) Memory A から構成される．

- Count er1 はカウント値を 3 ビット 2 進数 (3-bit binary) $\boldsymbol{x} = (x_2\ x_1\ x_0)$ で dat_1 に出力 (output) する．rst_1 はリセット信号 (reset signal) で $rst_1 = 1$ の時は他の入力 (input) によらずクロック信号 (clock signal) CLK に同期 (synchronous) してカウント値が (0, 0, 0) に初期化 (initialization) される．en_1 はイネーブル信号 (enable signal) で $rst_1 = 0$ かつ $en_1 = 1$ の時に CLK に同期してカウント値がカウントアップし，$rst_1 = 0$ かつ $en_1 = 0$ の時はカウント値が同じ値を保持し続ける．Count er1 のカウント値 $\boldsymbol{x}$ は (0, 0, 0) → (0, 0, 1) → (0, 1, 0) → (0, 1, 1) → (1, 0, 0) → (1, 0, 1) → (1, 1, 0) → (1, 1, 1) → (0, 0, 0) → … とカウントアップを繰り返す．
- Counter 2 は Counter 1 と同様にカウント値を 3 ビット 2 進数 $\boldsymbol{y} = (y_2, y_1, y_0)$ で dat_2 出力する．rst_2 はリセット信号で $rst_2 = 1$ の時は他の入力によらずクロック信号 CLK に同期してカウント値が (0, 0, 0) に初期化される．en_2 はイネーブル信号で $rst_2 = 0$ かつ $en_2 = 1$ の時に CLK に同期してカウント値がカウントアップし，$rst_2 = 0$ かつ $en_2 = 0$ の時はカウント値が同じ値を保持し続ける．Counter 2 のカウント値 $\boldsymbol{y}$ は (0, 0, 0) → (0, 0, 1) → … → (1, 1, 0) → (1, 1, 1) → (0, 0, 0) → … とカウントアップを繰り返す．
- Memory A は w_dat に入力された 8 ビット 2 進数データ $\boldsymbol{w} = (w_7, \ldots, w_0)$ を $w\_en = 1$ の時にクロック信号 CLK に同期して w_adr 番地に書き込む．また Memory A は r_adr 番地に格納された 8 ビット 2 進数データ $\boldsymbol{r} = (r_7, \ldots, r_0)$ を CLK に同期して読み出し，r_dat に出力する．Memory A は書き込みと読み出しが同時に可能で，全ての番地のデータは (0, …, 0) で初期化されているとする．

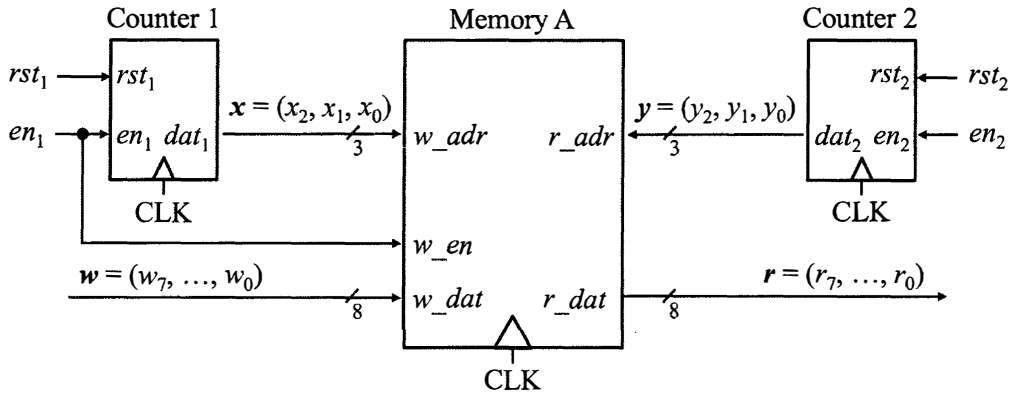

図 1

（次ページへ続く）

以下の各問に答えよ．

(1) 図 1 のデータパスにおいて Counter 1 と Counter 2 のカウント値が同じ値を示す時に 1，それ以外は 0 となる関数 f の積和標準形 (canonical sum-of-product form) を x_2，x_1，x_0，y_2，y_1，y_0 の論理関数 (logic function) として求めよ．ここで，解答の各積項のリテラル (literal) は x_2，x_1，x_0，y_2，y_1，y_0 の順で記載すること．

(2) Counter 1 を Mealy 型同期式順序回路 (Mealy synchronous sequential circuit) として設計する．このカウンタは三つの D フリップフロップ (D flip-flop) を持ち，出力するカウント値をそのまま状態割り当て (state assignment) に用いる．現在の状態 (current state) (x_2, x_1, x_0) に対する次状態 (next state) を (q_2, q_1, q_0) と表す時，x_2，x_1，x_0 とイネーブル信号 en_1 で表される，q_2，q_1，q_0 の論理式の最簡積和形 (最小積和形, minimum sum-of-products form) をそれぞれ求めよ．ここで，リセット信号 rst_1 は考慮しない．

(3) 図 2 に図 1 のデータパスのタイミングチャート (timing chart) を示す．①～⑩に当てはまる適切な値をそれぞれ答えよ．ここで，$\boldsymbol{x}$，$\boldsymbol{y}$，$\boldsymbol{w}$，$\boldsymbol{r}$ の値は全て 16 進数 (hexadecimal) で記載されており，解答も 16 進数で記載すること．

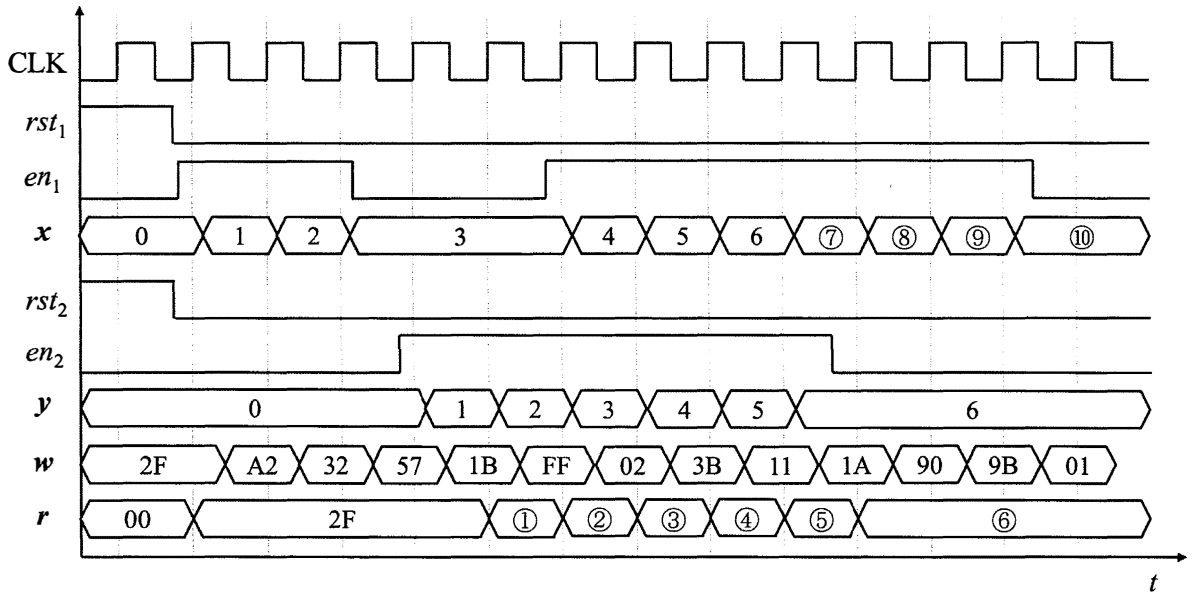

図 2

(4) 図 2 のタイミングチャートは図 1 のデータパスをある抽象データ型 (abstract data type) として使用した際の振る舞いである．どのような抽象データ型か名称を答えよ．

(5) このデータパスを前問 (4) の抽象データ型として使用中に，前問 (1) で求めた関数 f の値が 1 となった．この時この抽象データ型はどのような状況か，格納されているデータ数に着目して考えられる状況を全て答えよ．

7 【選択問題】数学解析と信号処理 （情報工学 12/13）

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

関数（function）$f(x;\theta)$ は，θ をパラメータ集合（set of parameters）とする確率密度関数（probability density function）を表すことにする．特に，確率密度関数が，平均（mean）μ，分散（variance）σ^2 をパラメータ $\theta_N = \{\mu, \sigma^2\}$ として以下のように表されるとき，この分布を正規分布（normal distribution）と呼ぶ．

$$N(x;\theta_N) = N(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{1}$$

ここで，データ集合 $\mathcal{D} = \{x_i \in \mathbb{R} \mid i = 1, ..., n\}$（ただし n は正の整数（positive integer））が与えられたとする．関数 $L_f(\theta) = \prod_{i=1}^{n} f(x_i;\theta)$ は尤度関数（likelihood function）であり，このとき尤度関数の対数 $\log L_f(\theta) = \log \prod_{i=1}^{n} f(x_i;\theta)$（対数尤度（log likelihood））を最大化する θ を求める問題は最尤推定（maximum likelihood estimation）と呼ばれ，情報科学のあらゆる分野において重要である．

以下の各問に答えよ．ただし，対数の底（base of logarithm）は e とする．

(1) ある確率密度関数 $f(x;\theta)$ に関する対数尤度 $\log L_f(\theta)$ が微分可能（differentiable）かつ上に凸（upper convex）であるとして，あるパラメータ $\theta_k \in \theta$ についての対数尤度の極大値を求めるための偏微分方程式（partial differential equation）を書け．

(2) 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) 正規分布 $N(x;\mu,\sigma^2)$ に従うデータ集合 $\mathcal{D}$ が与えられたとき，対数尤度が以下の形になることを示せ．また，このときの α, β を答えよ．

$$\log L_N(\theta_N) = \log L_N(\mu,\sigma^2) = \alpha \log\left(2\pi\sigma^2\right) - \beta \sum_{i=1}^{n} (x_i - \mu)^2 \tag{2}$$

(2-2) 式 (2) を，問 (1) で得られた微分方程式に代入し，正規分布 $N(x;\mu,\sigma^2)$ の μ についての対数尤度の最大化（最尤推定）が $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$ で得られることを示せ．ただし，$\hat{\mu}$ は，対数尤度を最大化する μ を表す．

（次ページへ続く）

(3)　最小二乗法（least square method）は，観測ベクトル $\boldsymbol{y}$ と，モデル $\boldsymbol{g}(\boldsymbol{x};\phi)$ による出力ベクトル $\hat{\boldsymbol{y}}$ の間の残差（residual）の二乗和（square sum）を最小化する（minimize）モデルパラメータ集合 $\hat{\phi}$ を求める方法である．この方法は，$\boldsymbol{y},\hat{\boldsymbol{y}}$ を n 次元の実数ベクトル（$\boldsymbol{y},\hat{\boldsymbol{y}}\in\mathbb{R}^n$）であるとして，以下のように定式化できる．

$$\hat{\phi}=\arg\min_{\phi}\|\boldsymbol{y}-\boldsymbol{g}(\boldsymbol{x};\phi)\|_2^2=\arg\min_{\phi}\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|_2^2=\arg\min_{\phi}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \tag{3}$$

ただし，関数 $\arg\min_{\phi}$ は，その引数が最小となる ϕ を求めるものである．

ここで，残差 $\boldsymbol{r}=\boldsymbol{y}-\hat{\boldsymbol{y}}$ の各次元が独立（independent）であり，$\boldsymbol{r}=[r_1,...,r_n]$，$\boldsymbol{y}=[y_1,...,y_n]$，$\hat{\boldsymbol{y}}=[\hat{y}_1,...,\hat{y}_n]$ について，$r_i=y_i-\hat{y}_i$ (ただし $i=1,...,n$) が平均 0 分散 σ^2 の正規分布 $N(r_i;0,\sigma^2)$ に従うとする．このときのモデルパラメータ ϕ の最尤推定が最小二乗法と一致することを示せ．ただし，問 (2) の導出過程の一部を適宜用いて良い．

(4)　図 1(a)〜(d) に，単一の正弦波（sine wave）からなる連続時間信号（continuous-time signal）である原信号（original signal）$y^*(t)$ を実線で示す．ここで，原信号に未知のノイズ（noise）が加わった状態で，窓幅（window size）n [sec]，サンプリング周波数（sampling frequency）1 [Hz] で観測したとき，それぞれ (a)〜(d) 中の点列（dots）で示す離散時間信号（discrete-time signal）$y[i]$（ただし $i=1,...,n$）が得られた．

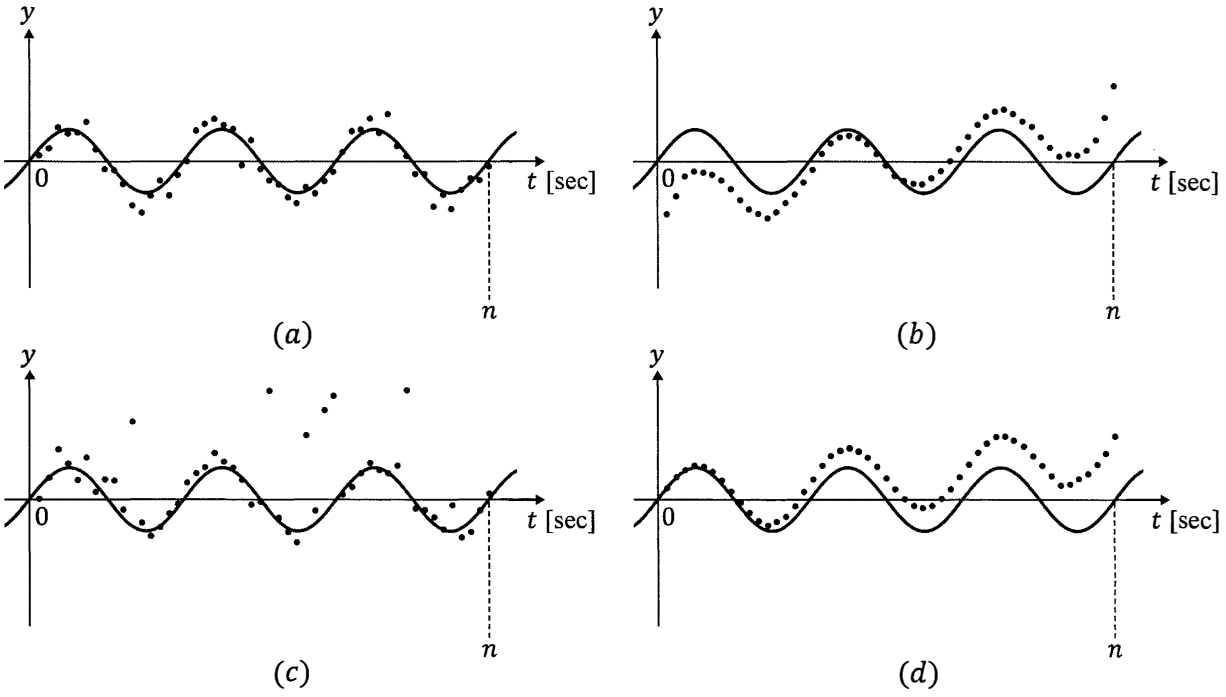

図 1　原信号 $y^*(t)$（実線）および，観測された離散時間信号 $y[i]$（点列）

図 1(a)〜(d) において，$y[i]$ の各サンプル（sample）$y[1],...,y[n]\in\mathbb{R}$ を観測とし，モデルを $\hat{y}(t;\gamma,\omega)=\gamma\sin\omega t$ として (a)〜(d) それぞれに定義し，最小二乗法によるパラメータ $\phi=\{\gamma,\omega\}$ の推定を行った．結果，(a), (b) それぞれの観測信号に対する推定では原信号を復元できたが，(c), (d) では復元できなかった．(a), (b) で復元できた理由，(c), (d) で復元できなかった理由をそれぞれ 2〜3 行で説明せよ．その際，ノイズにどのような仮定を置いたのか，明確に論ずること．