1 【必須問題】アルゴリズムとプログラミング

配点：(1) 30 点，(2) 20 点，(3) 15 点，(4) 25 点，(5) 20 点，(6) 15 点

図 1 は ANSI-C 準拠である C 言語のプログラム（program）である．このプログラムにおいて，insert 関数はある特定の規則に従ってデータ列（data sequence）に新しいデータ（data）を挿入（insert）する関数（function）であり，データ列を格納する配列（array）A，配列の大きさ SIZE，挿入するデータ d を引数（arguments）とする．delete 関数はデータ列からある特定の規則に従ってデータを一つ取り出し(retrieve)，削除（delete）する関数であり，データ列を格納する配列 A を引数とし，取り出したデータを戻り値（return value）とする．main 関数は，それらの二つの関数を実行する一例を示している．以下の各問に答えよ．

(1) 図 1 のプログラムにおいて，43 行目および 46 行目の処理を実行した結果をそれぞれ示せ．

(2) 図 1 のプログラムにおいて，43 行目の処理を実行した直後の変数 front および変数 rear の値をそれぞれ示せ．

(3) 図 1 の insert 関数内の 11～21 行目では，新たに挿入するデータの挿入位置を決定している．この際，データ列を最初から一つずつ調べることなく，処理を効率化している．具体的にどのようなことをしているかを簡潔に説明せよ．

(4) データ列のデータ数を n としたとき，insert 関数および delete 関数を実行する際の時間計算量（time complexity）のオーダー（order）を示せ．その理由も簡潔に説明せよ．

(5) 図 1 の main 関数において，47 行目以降に insert 関数および delete 関数を多数回実行した場合，データ列のデータ数に関わらず，実行途中でデータの挿入に失敗してしまう．その原因，および，データの挿入が失敗する条件を，データ挿入回数の観点から簡潔に説明せよ．

(6) 上記(5)の問題を解決するために，配列を十分大きくする，および，データの挿入回数に制限を設ける以外に，どのような方法があるか，簡潔に説明せよ．ただし，その方法が insert 関数および delete 関数の時間計算量のオーダーを変えることがあってはならない．また，できるだけ時間計算量の増加が少ない方法を示すこと．なお，その方法を実現するために，関数に新たな引数を追加することがあってもよいが，新たな関数は追加してはならない．

```
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  int front = 0;
5  int rear = 0;
6
7  void insert(int A[], int SIZE, int d) {
8    int p, left, right, m, i;
9    if (rear > SIZE-1) { printf ("Overflow !!\n"); exit(1); }
10
11   p = -1;
12   if (front == rear) p = front;
13   else {
14     left = front; right = rear-1;
15     while (left < right) {
16       m = (left+right)/2;
17       if (A[m] == d) { p = m+1; break; }
18       if (d < A[m]) right = m-1; else left = m+1;
19     }
20     if (p == -1) { if (A[left] > d) p = left; else p = left+1; }
21   }
22
23   i = rear;
24   while (i > p) { A[i] = A[i-1]; i--; }
25   A[p] = d; rear++;
26 }
27
28 int delete(int A[]) {
29   int x;
30   if (front == rear) { printf ("Underflow !!\n"); exit(1); }
31   x = A[front]; front++;
32   return x;
33 }
34
35 int main () {
36   int i;
37   int A[20]; int SIZE = 20;
38
39   int a1[5] = {10, 5, 20, 6, 13};
40   int a2[5] = {2, 5, 18, 7, 5};
41
42   for (i = 0; i < 5; i++) insert(A, SIZE, a1[i]);
43   for (i = 0; i < 3; i++) printf("%d ", delete(A)); printf ("\n");
44
45   for (i = 0; i < 5; i++) insert(A, SIZE, a2[i]);
46   for (i = 0; i < 7; i++) printf("%d ", delete(A)); printf ("\n");
47
48   return 0;
49 }
```

図 1：プログラム

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 3/12）

配点：(1-1) 36 点，(1-2) 8 点，(1-3) 8 点，(1-4) 18 点，(2-1) 20 点，(2-2) 35 点

(1) 計算機の演算器に関する以下の小問に答えよ．解答は全て解答欄に書くこと．

(1-1) 以下に示す仕様の計算機およびC言語コンパイラを考える．

- 演算器 (ALU: arithmetic logic unit)，レジスタ，メモリの語 (word) 長は全て 8 ビット．
- int 型は 8 ビット符号付き (signed) 整数 (integer) で，2 の補数系 (2's complement system) 表現．
- 整数演算時にはオーバーフローは無視され，演算器での演算結果がそのまま格納される．
- float 型は 8 ビット浮動小数点数 (floating point number)．表現したい数を $(-1)^s \times (1+m) \times 2^{(e-3)}$ の形式で表し，最上位ビット (MSB: most significant bit) から s，e，m の順にメモリに格納する．s は 1 ビット，e は 3 ビット，m は 4 ビットである．仮数部 (significand) は正規化 (normalize) ($1 \leq 1+m < 2$) されており，整数部 (integer part) の 1 はメモリには格納されない (隠しビット (hidden bit))．仮数部の 2 進数表現の小数部 (fractional part) 上位 4 桁をメモリに格納する．5 桁目以降は切り捨て (round down) により丸める．

この計算機およびC言語コンパイラを用いて，図 1 に示すプログラムをコンパイル，実行した．return 文の時点での各変数 (variable) の値を 10 進数で示せ．また，それらを格納するメモリの内容をビット列 (bit string) で示せ．なお，float 型変数の値は，指数表記 (scientific notation) を用いずに表すこと．また，メモリの内容 (ビット列) は，左端を最上位ビットとして示すこと．

(1-2) 計算機において，符号付き整数の表現に 2 の補数系表現を用いることの利点を述べよ．

(1-3) 入力として 8 ビット符号なし (unsigned) 整数 y (ビット列：$y_7\, y_6 \cdots y_0$) と 1 ビット制御信号 (control signal) w が与えられたとき，出力 y' (ビット列：$y'_7\, y'_6 \cdots y'_0$) として，$w=0$ ならば y そのものを，$w=1$ ならば y の 1 の補数 (1's complement) を出力する回路を作ることを考える．この回路は，同じ構成の回路を 8 ビット分並べて構成することができる．第 i ビット ($i \in [0,7]$) 1 ビット分の回路を，図 2 に示す排他的論理和 (exclusive OR) を用いて構成し，その回路図を示せ．なお，排他的論理和の論理式は $f = g\bar{h} + \bar{g}h$ である．必要ならば NOT 回路も用いてよい．

(1-4) 図 3 に示す 8 ビット符号なし整数加算器を用いて，2 の補数系表現による 8 ビット符号付き整数の加算ならびに減算を行う演算器を作ることを考える．被演算数 (operand) x (ビット列：$x_7\, x_6 \cdots x_0$)，y (ビット列：$y_7\, y_6 \cdots y_0$)，1 ビットの演算制御信号 w が与えられ，演算結果 z (ビット列：$z_7\, z_6 \cdots z_0$) が出力される．$w=0$ ならば $z=x+y$，$w=1$ ならば $z=x-y$ となる．この回路を，排他的論理和と 8 ビット符号無し整数加算器を用いて構成し，その回路図を書け．なお，必要ならば NOT 回路も用いてよい．

```
int main()
{
  int i, j, k, m, n;
  float d, e;

  i = 90;
  j = -40;
  k = i + j;
  m = i - j;
  n = j - i;
  d = 0.4;
  e = d + 0.8;

  return 0;
}
```

図 1

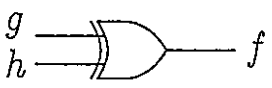


図 2

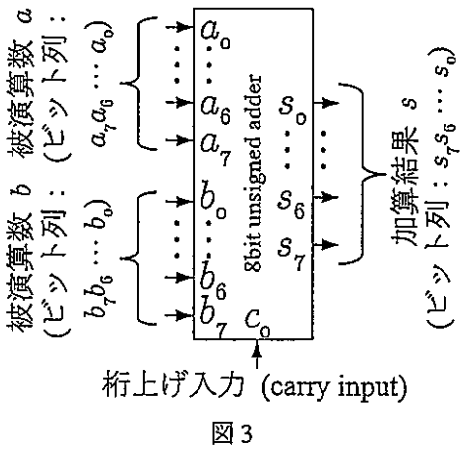


図 3

(2) 排他制御(mutual exclusion)に関する以下の小問に答えよ.

(2-1) 以下の空欄 [a] ～ [c] を適切な用語で埋めよ.

共有資源(shared resource)の排他制御を実現する方法としてテストアンドセット(test and set)やセマフォ(semaphore)等がある. テストアンドセットは一つしかない共有資源の排他制御を実現する機械語命令(machine instruction)である. これは,「共有資源の [a] 状態と [b] 状態とを監視し制御する」機能を不可分操作(atomic operation) として実行する命令である. 不可分操作として実行されるのは以下の二つである.

1) テスト : 共有資源が [a] 状態か [b] 状態かをチェックする. [b] 状態の場合, 終了する.

2) セット : 共有資源を [b] 状態に変更する.

ここでテストとセットは不可分操作のため, [c] が発生しない.

(2-2) テストとセットを不可分操作とする必要性を, 不可分操作でなかった場合に排他制御が失敗する状況を例示することにより明らかにせよ. なお回答では以下の状況を仮定せよ.

- 二つのプロセス X と Y が一つしかない共有資源 Z を取り合う.
- プロセス X が先にテストを実行する.

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

(1) 任意の空でない有向グラフ (directed graph) $G = (V, E)$ $(E \subseteq V \times V)$ について，$V \times V$ 上の以下の二項関係 (binary relation) $R_i\,(i = 1, \ldots, 5)$ を考える．なお，$S = \{(x, x) | x \in V\}$ である．有向経路 (directed path) および有向閉路 (directed cycle) はいずれも，異なる頂点を含むものとする．R_3 に関しては，ある $s \in V$ が存在し，s から他のすべての頂点への有向経路が存在するような G のみを対象とする．以下の各小問に答えよ．

$$R_1 = \{(x, y) \mid x \text{ から } y \text{ への有向経路が存在する }\} \cup S$$
$$R_2 = \{(x, y) \mid x \text{ と } y \text{ を共に含む有向閉路が存在する }\} \cup S$$
$$R_3 = \{(x, y) \mid s \text{ から } y \text{ へのすべての有向経路が } x \text{ を含む }\}$$
$$R_4 = \{(x, y) \mid (x, y) \in E\} \cup S$$
$$R_5 = \{(x, y) \mid (\exists z \in V)\,[(x, z) \in E \wedge (y, z) \in E]\} \cup S$$

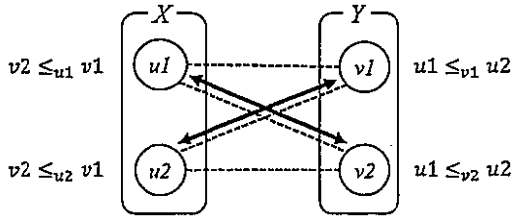
(1-1) 同値関係 (equivalence relation) である R_i をすべて挙げ，それぞれ同値関係であることを証明せよ．

(1-2) 半順序関係 (partial order relation) である R_i をすべて挙げ，それぞれ半順序関係であることを証明せよ．

(1-3) グラフ G を有向閉路のない有向グラフに限定した場合に，半順序関係である R_i をすべて挙げよ．

(2) 論理記号 $\Leftrightarrow$, $\rightarrow$, $\wedge$, $\vee$, $\neg$ 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．$\bigwedge_{z \in Z} p(z)$ で集合 Z に属する値 z を引数に持つ論理式 $p(z)$ をすべて論理積で結合した論理式を，$\bigvee_{z \in Z} p(z)$ でそのような $p(z)$ をすべて論理和で結合した論理式を表す．

完全 2 部グラフ (complete bipartite graph) $G = (X \cup Y, X \times Y)$ で $|X| = |Y|$，$X \cap Y = \emptyset$ とする．各頂点 x $(x \in X)$ に対して Y の頂点間の全順序関係 (total order relation) $\leq_x$ が与えられ，かつ各頂点 y $(y \in Y)$ に対して X の頂点間の全順序関係 $\leq_y$ が与えられるとする．このとき，$\forall (x, y) \in ((X \times Y) \backslash M)\ (\exists (a, b) \in M\ (x \leq_y a \vee y \leq_x b))$ であるようなマッチング (一対一対応，matching) M は安定 (stable) であるという．下図は安定なマッチングの例である (破線 (dashed line) が辺を，矢印 (arrow) がマッチングをそれぞれ表す)．



安定なマッチングを得るあるアルゴリズム A が満たす性質を，以下の述語 $r(h, x, y)$ および $m(h, x, y)$ を用いて表現することを考える．なお，h，x および y の対象領域をそれぞれ非負整数，X および Y とする．A はラウンドと呼ばれる時間単位で動作し，0 を初期ラウンドとする．

$r(h, x, y)$ $\Leftrightarrow$ ラウンド h において，x が y にマッチング要求をする
$m(h, x, y)$ $\Leftrightarrow$ ラウンド h において，(x, y) はマッチングの要素である

A は以下の 1.~3. にしたがって動作する．

1. ラウンド 0 ではどの頂点もマッチング要求をせず，かつどの辺もマッチングの要素でない．これらは以下の論理式 $CZ1$ および $CZ2$ で表せる．

$CZ1$: $\forall x \forall y\ \neg r(0, x, y)$
$CZ2$: $\forall x \forall y\ \neg m(0, x, y)$

2. 各ラウンド h ($h \geq 1$) ではある条件を満たす X の頂点がマッチング要求をする．これに関する性質記述として論理式 $CX1 \sim CX5$ を与える．例えば $CX1$ は各ラウンドにおいて x のマッチング要求はただ一つだけであることを表している．なお，ラウンドに対する述語 $<$ ならびに二項演算子 $-$ の解釈は整数上のそれに従い，$n(x,x')$ は二つの頂点 x，x' が異なるときかつそのときに限り真となる述語である．

$CX1$: $\forall h \forall x \forall y \forall w \, (\, (\, r(h,x,y) \wedge n(y,w) \,) \rightarrow \neg r(h,x,w) \,)$

$CX2$: $\forall h \forall k \forall x \forall y \, (\, (\, r(h,x,y) \wedge k < h \,) \rightarrow \neg r(k,x,y) \,)$

$CX3$: $\forall h \forall x \, (\, (\bigwedge_{y \in Y} \neg m(h-1,x,y)) \rightarrow \bigvee_{y \in Y} r(h,x,y) \,)$

$CX4$: $\forall h \forall k \forall x \forall y \forall w \, (\, (\, r(h,x,y) \wedge r(k,x,w) \wedge n(y,w) \wedge k < h \,) \rightarrow y \leq_x w \,)$

$CX5$: $\forall h \forall k \forall x \forall y \forall w \, (\, (\, r(h,x,y) \wedge \neg r(k,x,w) \wedge n(y,w) \wedge k < h \,) \rightarrow w \leq_x y \,)$

3. ラウンド h のマッチング要求は Y の頂点ごとに処理される．その性質記述として論理式 $CY1 \sim CY5$ を与える．

$CY1$ $\forall h \forall x \forall y \forall z \, (\, (\, r(h,x,y) \wedge m(h-1,z,y) \wedge x \leq_y z \,) \rightarrow \neg m(h,x,y) \,)$

$CY2$ $\forall h \forall x \forall y \forall z \, (\, (\, r(h,x,y) \wedge r(h,z,y) \wedge x \leq_y z \,) \rightarrow \neg m(h,x,y) \,)$

$CY3$ $\forall h \forall x \forall y \, ($ (a) $)$

$CY4$ $\forall h \forall x \forall y \, ($ (b) $)$

$CY5$ $\forall h \forall x \forall y \, ($ (c) $)$

(2-1) $CY3 \sim CY5$ が以下を表すように，空欄 (a)~(c) の論理式を埋めよ．$\exists$ および $\forall$ は利用しないこと．

$CY3$: 任意の (x,y) とラウンド h について，h において x が y にマッチング要求するならば，h において y を含む辺の少なくとも一つはマッチングの要素である

$CY4$: 任意の (x,y) とラウンド h について，$h-1$ において (x,y) がマッチングの要素ならば，h において y を含む辺の少なくとも一つはマッチングの要素である

$CY5$: 任意の (x,y) とラウンド h について，h において (x,y) がマッチングの要素であれば，ラウンド $h-1$ において (x,y) はマッチングの要素であるか，あるいは h において x が y にマッチング要求する

(2-2) 「A のもとで得られるマッチングが，あるラウンドにおいて安定であること」を，ある論理式 P が充足不能であることを示すことにより示したい．P を，$CX1 \sim CX5$，$CY1 \sim CY5$，$CZ1 \sim CZ2$，および，あるラウンド h におけるマッチングが安定であることを表す閉論理式 H を用いて表せ．

(2-3) $X = \{u1, u2\}$，$Y = \{v1, v2\}$ とし，$v2 \leq_{u1} v1$，$v2 \leq_{u2} v1$，$u1 \leq_{v1} u2$，$u1 \leq_{v2} u2$ とする．

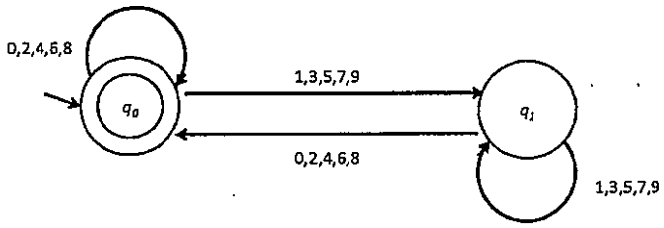
(2-3-1) 以下の空欄 (d)~(h) を，述語の引数に指定された値を代入した命題あるいはその論理和の形の論理式で埋めよ．$CZ1$ と $CZ2$，および $n(x,x')$，$\leq_z$ ($z \in X \cup Y$)，$<$，$-$ の解釈のもとで偽である命題は除くこと．

$CX3$ で $(h,x) \leftarrow (1,u1)$ とすると	$A1$: $r(1,u1,v1) \vee r(1,u1,v2)$
$CX3$ で $(h,x) \leftarrow (1,u2)$ とすると	$A2$: $r(1,u2,v1) \vee r(1,u2,v2)$
$CX5$ で $(h,k,x,y,w) \leftarrow (1,0,u1,v2,v1)$ とすると	$A3$: (d)
$CX5$ で $(h,k,x,y,w) \leftarrow (1,0,u2,v2,v1)$ とすると	$A4$: (e)
$CY2$ で $(h,x,y,z) \leftarrow (1,u1,v1,u2)$ とすると	$A5$: $\neg r(1,u1,v1) \vee \neg r(1,u2,v1) \vee \neg m(1,u1,v1)$
$CY3$ で $(h,x,y) \leftarrow (1,u1,v1)$ とすると	$A6$: (f)
$CY5$ で $(h,x,y) \leftarrow (1,u1,v2)$ とすると	$A7$: (g)
$CY5$ で $(h,x,y) \leftarrow (1,u2,v2)$ とすると	$A8$: (h)

(2-3-2) 上記 (2-3-1) の制約のもとでは，ラウンド 1 の終了時に $(u2,v1)$ のみがマッチングの要素であることを，導出原理 (resolution principle) を用いて証明せよ．論駁過程 (refutation process) も示すこと．

配点:(1-1) 25 点，(1-2) 25 点，(1-3) 10 点，(2-1) 15 点，(2-2) 30 点，(2-3) 20 点

(1) 有限オートマトン (finite automaton) は 5 項組 $(Q, \Sigma, \delta, q_0, F)$ で与えられる．ここで $Q, \Sigma, \delta, q_0, F$ は，それぞれ，状態 (state) の有限集合，入力記号 (input symbol) の有限集合であるアルファベット (alphabet)，遷移関数，開始状態 (initial state) ($q_0 \in Q$)，受理状態集合 ($F \subseteq Q$) である．有限オートマトンに関する以下の各小問に答えよ．アルファベット Σ を $\{0,1,2,3,4,5,6,7,8,9\}$ とする．以下の各小問では Σ 上の語 (word) を先頭を上位桁とする 10 進数の非負整数とみなす．通常の非負整数表現以外の語 (例えば 00045) についての動作は考慮しなくて良い．次に与える有限オートマトン $\mathcal{A}$ は与えられた語が 2 で割り切れるときのみ受理 (accept) するオートマトンである．



(1-1) 与えられた語が 3 で割り切れるときのみ受理する有限オートマトン $\mathcal{B}$ の構築の方法を説明せよ．説明にあたってはオートマトンの図を用いてもよい．

(1-2) 小問 (1-1) の構築方法で得られる有限オートマトン $\mathcal{B}$ を用いて 15 で割り切れるときのみ受理する有限オートマトン $\mathcal{D}$ を構築したい．まず与えられた語が 5 で割り切れるときのみ受理する有限オートマトン $\mathcal{C}$ を示せ．ついでこの 2 つの有限オートマトン $\mathcal{B}$ と $\mathcal{C}$ をどのように用いればオートマトン $\mathcal{D}$ を構築できるかを説明せよ．

(1-3) 小問 (1-2) の構築方法で得られる有限オートマトン $\mathcal{D}$ を少し変更し，3 または 5 で割り切れるときのみ受理する有限オートマトン $\mathcal{E}$ を構築するにはどのようにすればよいか，変更の方法を説明せよ．

(2) 文脈自由文法 (context-free grammar) G は 4 項組 (N, T, P, S) で与えられる．ここで N, T, P, S は，それぞれ，非終端記号 (non-terminal symbols) 集合，終端記号 (terminal symbols) 集合，生成規則 (production rules) 集合，開始記号 (start symbol) である．文脈自由文法 G の生成文全体を表す言語を $L(G)$ とする．この言語は一般に文脈自由言語 (context-free language) と呼ばれる．これらに関する以下の各小問に答えよ．

(2-1) 文脈自由言語 $L_1 = \{c^m \mid m \geq 1\}$ を生成する文脈自由文法 G_1 を示せ．

(2-2) 言語 $L_2 = \{a^n b^n c^m \mid n \geq 1, m \geq 1\}$ および $L_3 = \{a^m b^n c^n \mid n \geq 1, m \geq 1\}$ が文脈自由言語であることを証明せよ．言語 L_2, L_3 を生成する文法を示して証明する場合は，それらの生成言語がそれぞれ L_2, L_3 と等価になることを証明すること．なお必要であれば小問 (2-1) の結果と以下の補題 1(lemma) を証明なしに用いてよい．

[補題 1]

文脈自由言語全体の集合は連接演算 $\cdot$ について閉じている．すなわち，任意の 2 つの文脈自由言語 L_α と L_β に対して言語 $L_\alpha \cdot L_\beta = \{xy \mid x \in L_\alpha, y \in L_\beta\}$ も文脈自由言語である．

(2-3) 文脈自由言語全体の集合が積演算 $\cap$ について閉じていないことを証明せよ．必要であれば小問 (2-2) の結果と以下の補題 2 を証明なしに用いてよい．

[補題 2]

言語 $L_4 = \{a^n b^n c^n \mid n \geq 1\}$ は文脈自由言語ではない．

5 【選択問題】ネットワーク

配点: (1-1)6 点, (1-2)8 点, (1-3-1)12 点, (1-3-2)14 点,
(2-1)20 点, (2-2)10 点, (2-3)10 点,
(3-1)20 点, (3-2)10 点, (3-3)15 点

(1) 記憶がない情報源 (memoryless information source) の符号化について以下の小問に答えよ.

(1-1) 記憶がない情報源とは，どのような情報源か説明せよ.

(1-2) 以下の文中の空欄 (あ) ～ (え) それぞれに適当な語を選択肢から一つ選び，その記号を答えよ.

> (あ) が最小となる符号化を最適符号化 (または，コンパクト符号化) (optimum coding, compact coding) という．最適符号化を求める方法として (い) のアルゴリズムが知られている．また，(あ) の上界と下界を (う) を用いて与える (え) の情報源符号化定理が知られている.

選択肢： (a) クラフト (Kraft) (b) シャノン (Shannon) (c) ハフマン (Huffman) (d) ハミング (Hamming)
(e) エントロピー (entropy) (f) 最大符号語長 (maximum codeword length)
(g) 平均符号語長 (average codeword length) (h) 符号化率 (code rate)

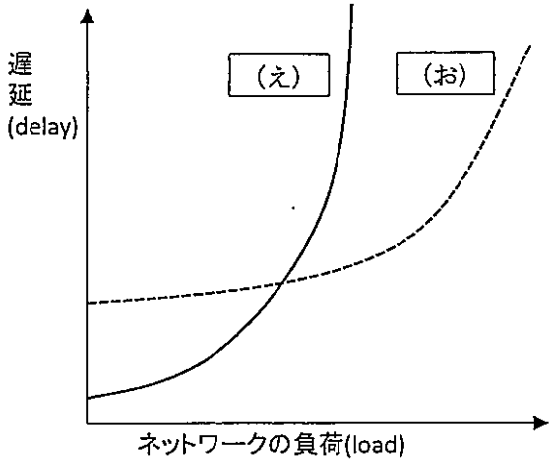
(1-3) 情報源記号が s 個の記憶がない情報源を考える．各記号の生起確率 $p_1, p_2, \ldots, p_s$ は，$p_1 > p_2 > \cdots > p_s > 0$ を満たすとする．小問 (1-2) の文中の最適符号化を求めるアルゴリズムによって得られる最適符号化について考える.

(1-3-1) $s = 4$ の場合に，符号語の長さがすべて等しくなる 2 元最適符号化が存在するための必要十分条件 (記号の生起確率についての条件) を書け.

(1-3-2) $s = 5$ の場合に，5 個のうち 4 個の符号語の長さが等しくなる 2 元最適符号化が存在するための必要十分条件 (記号の生起確率についての条件) を書け．また，それが必要十分条件となることを説明せよ.

(2)メディアアクセス制御プロトコル(media access control protocol)に関する説明文を読み，以下の各小問に答えよ．

バス型のネットワーク(network)では，ネットワークを効率良く使用するため，［(あ)］方式や［(い)］方式のメディアアクセス制御プロトコルが開発されている．［(あ)］方式は，送信ホスト(host)間でフレーム(frame)を送信する権利を順にまわしていくものであり，一方，［(い)］方式は，送信ホスト間で調整を行わずにフレームを送信する．［(い)］方式の一つである CSMA/CD(Carrier Sense Multiple Access / Collision Detection)では，送信ホストは送信前に［(う)］を行い，他のホストがフレームを送信していないことを確認する．これらの方式では，送信ホスト数が増加してネットワークの負荷(load)が高まるにつれて，送信ホストから受信ホストにフレームが到達するまでの遅延(delay)が増加する．下記のグラフでは，ネットワークの負荷を変化させた時の遅延の変化を模式的に示している．



(2-1) 説明文の［(あ)］～［(お)］に当てはまる最も適切な語句を選択肢から一つ選び，その記号を答えよ．ここで，［(え)］及び［(お)］にはそれぞれ，［(あ)］，［(い)］で選択したいずれかの選択肢が入る．

選択肢：

(a)ポイント・ツー・ポイント(point-to-point)
(b)コネクション(connection)
(c)コネクションレス(connectionless)
(d)ランダムアクセス(random access)
(e)トークンパッシング(token passing)
(f)搬送波検知(carrier sense)
(g)バックオフ(back off)
(h)タイマ(timer) 設定

(2-2) ネットワークの負荷が高くなった場合，［(お)］方式が［(え)］方式と比較して，遅延が急激に増加しない理由を説明せよ．

(2-3) CSMA/CD ではフレームが衝突すると，定められた平均値になるよう設定されたランダムな待ち時間だけ，送信ホストはフレームの再送を待つ．ここで，再送フレームが再び衝突した場合，さらなる衝突が発生しないように，CSMA/CD でなされている工夫を述べよ．

(3) TCP Reno が実装する輻輳制御(congestion control)に関する説明文を読み,以下の各小問に答えよ.

送信ホストは，輻輳ウィンドウ(congestion window)と呼ばれる，ACK セグメント(acknowledge segment)を受信せずに送信可能な DATA セグメント(data segment)のデータ量を用いて送信レート(transmission rate)を増減する．通信開始時の輻輳ウィンドウは，MSS (Maximum Segment Size)と呼ばれる最大セグメント長 1 個分である．通信開始後，送信ホストは ACK セグメントを受信するごとに，以下の式に従って輻輳ウィンドウを増加させる．

新しい輻輳ウィンドウ ＝ 現在の輻輳ウィンドウ ＋ [(あ)] ×MSS

送信ホストは，[(い)] ことで DATA セグメントの紛失を検出すると，高速リカバリ(fast recovery)を開始し，それが終了した時点で輻輳ウィンドウを [(う)] 倍に減少させる．その後，送信ホストは，ACK セグメント受信毎の増加量を [(え)] ×MSS にする．またこの送信レートの増減のしかたは，[(お)] と呼ばれる．

(3-1) 上記の説明文の [(あ)]～[(お)] に当てはまる最も適切な語句を選択肢から一つ選び，その記号を答えよ.

選択肢：

(a)MSS÷現在の輻輳ウィンドウ　(b)2MSS÷現在の輻輳ウィンドウ　(c)1　(d)2
(e)1/2　(f)1/4　(g)1/8
(h)重複した ACK セグメントを 3 個受信する
(i)送信した DATA セグメントを確認応答する ACK セグメントを一定時間受信しない
(j)ベストエフォート(best effort)　(k)Additive Increase/Multiplicative Decrease
(l)スロースタート(slow start)　(m)コネクションレス

(3-2) 送信ホストに送信すべき DATA セグメントが常に存在すると仮定する場合，以下の条件に従って通信を開始すると，送信ホストが最初の DATA セグメントを送信後，最長で何ミリ秒後までに最初の ACK セグメントを受信するかを求めよ.

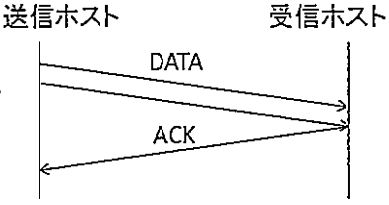
●受信ホストは Delayed ACK に準じて，以下のいずれかの条件が満たされると，ACK セグメントを 1 個返送する.

条件 1：DATA セグメントを受信した時，当該 DATA セグメントを含めて，ACK セグメントを返送していない DATA セグメントのデータ量が MSS 2 個分以上である.

条件 2：200 ミリ秒周期の Delayed ACK タイマ発火時に，ACK セグメントを返送していない DATA セグメントが存在する.

●片道の伝送遅延時間 5 ミリ秒.　●MSS 1K バイト.
●DATA 及び ACK セグメントは紛失しない.　●全ての DATA セグメントのデータ量は MSS.
●受信ホストが送信ホストに通知する最大のウィンドウサイズ 12K バイト.
●以上の条件で所要時間が規定されていない処理については，所要時間は無視して良い.

(3-3) 小問(3-2)と同じ条件で通信を開始する時，送信ホストが最初の DATA セグメントを送信してから，最長で何ミリ秒後までに 5 個目の ACK セグメントを受信するかを求めよ．さらに，送信ホストが 5 個目の ACK セグメントを受信するまでの流れを,右の例に倣った図で示せ.



時間0で連続してDATAセグメントを送信する場合は，上図に示すように，それらを示す矢印をずらして記載すること.

6 【選択問題】電子回路と論理設計 （情報工学 11/12）

配点：(1-1) 25 点，(1-2) 10 点，(1-3) 20 点，(1-4) 20 点，(2-1) 20 点，(2-2) 30 点

2 入力マルチプレクサ（multiplexor）について，次の各問に答えよ．

2 入力マルチプレクサは，3 個の入力 select，a，b と 1 個の出力 out をもち，入力 select が 1 のとき out に a の値を出力し，select が 0 のとき out に b の値を出力する．

(1) 2 入力マルチプレクサを次の手順で，論理回路（logic circuit）で実現する．

(1-1) 2 入力マルチプレクサの真理値表（truth table）を作成せよ．

(1-2) 2 入力マルチプレクサのカルノー図（Karnaugh map）を作成せよ．

(1-3) (1-2)で作成したカルノー図を利用して，out 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．

(1-4) (1-3)で求めた最簡積和形の論理式を用いて得られる 2 入力マルチプレクサの論理回路図を示せ．ただし，論理ゲート（logic gate）は，2 入力 NAND ゲートのみが使用できるものとする．

(2) (1-4)で設計した 2 入力マルチプレクサの遅延時間（delay）について考える．遅延時間とは，入力が変化してから出力が変化するまでに要する時間である．ただし，設計に用いた 2 入力 NAND ゲートの遅延時間は，出力が 0 から 1 に変化するときは T，出力が 1 から 0 に変化するときは 2T であるとする．

(2-1) 入力 (select, a, b) が (0, 0, 0) から (0, 0, 1) に変化するときの，マルチプレクサの遅延時間を T を用いて表せ．

(2-2) select，a，b のいずれか一つの入力が変化するときの，マルチプレクサの遅延時間の最大値を求めよ．また，そのときの入力の変化を下の例にならって答えよ．

例：(select, a, b) が (1, 1, 1) から (0, 1, 1) に変化するとき．

配点: (1-1)35 点, (1-2)40 点, (2)50 点

以下の各問に答えよ.

(1)

(1-1) $f(t) = \cos \omega t$ のラプラス変換 (Laplace transform) が $\frac{s}{s^2 + \omega^2}$ であることを示せ.
ただし，この関数 f は $(0, +\infty)$ で定義され，ω は定数 (constant number)，s は複素変数 (complex variable) である.
(1-2) ラプラス変換を用いて，次の積分方程式 (integral equation) を満たす $x(t)$ を求めよ．ただし，この関数 x は $(0, +\infty)$ で定義される.

$$x(t) = 1 - 4 \int_0^t (t - \tau) x(\tau) d\tau$$

(2)

図 1 に示すような周期 (period) $T = 2$ の矩形波 (square wave) $f(t)$ をフーリエ級数展開 (Fourier series expansion) せよ．ただし，n は整数 (integer number) である.

$$f(t) = \begin{cases} 0 & (2n - 1 < t \leq 2n - 0.5) \\ 1 & (2n - 0.5 < t \leq 2n + 0.5) \\ 0 & (2n + 0.5 < t \leq 2n + 1) \end{cases}$$

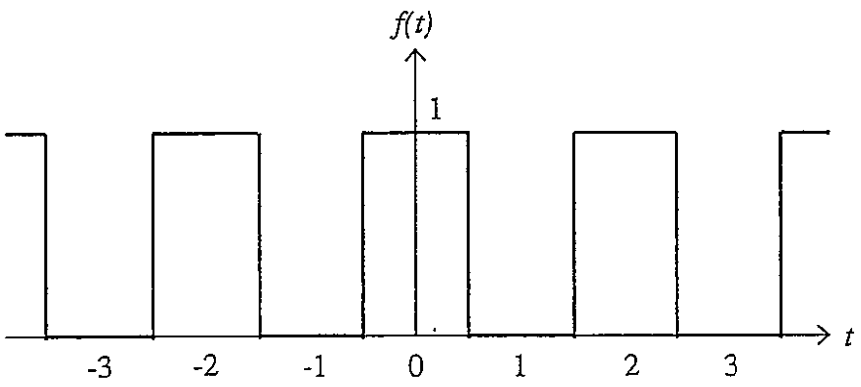


図 1: 周期 $T = 2$ の矩形波 $f(t)$