1 【必須問題】アルゴリズムとプログラミング

配点：(1-1) 20 点，(1-2) 20 点，(1-3) 25 点，(2-1) 32 点，(2-2) 28 点

(1)

図 1 に示す ANSI-C 準拠である C 言語のプログラム（program）は，重複のない N（自然数，1≦N≦2000 とする）個の整数を昇順に配列（array）A の要素 A[1]～A[N]に格納し，変数 x の値が配列 A の要素に含まれているかどうかを 2 分探索法（binary search）を用いて探索するプログラムである．このプログラムは，探索の途中経過と，x の値が配列 A の要素に含まれる場合はその値が格納されている配列 A のインデックス（index），x の値が配列 A の要素に含まれない場合は 0 を出力する．N の値の設定，配列 A[1]～A[N]に整数値を設定する処理，変数 x の値を設定する処理，及び，これらの値が妥当かどうかを確認する処理は省略している．以下の各小問に答えよ．

(1-1)　図 1 のプログラムに対して，N，A[1]～A[N]，x を下記のように設定した時の出力結果を書け．

N=8, A[1]=2, A[2]=5, A[3]=8, A[4]=10, A[5]=15, A[6]=20, A[7]=25, A[8]=30, x=8

(1-2)　N=1000 で，x の値が A[1]～A[N]のどこかに存在する場合，図 1 中の do-while ループが実行される回数は最大何回か．その理由も簡潔に説明せよ．

(1-3)　図 1 の do-while ループ中の □で囲われた部分を下記のように変更する．

変更前

```
if (A[mid]<x) left = mid+1;
else          right= mid-1;
```

⇨

変更後

```
if (A[mid]<x) left = mid;
else          right= mid;
```

変更前のプログラムは正しく結果が出力されるが，変更後のプログラムは正しく結果が出力されないことがある．そのような N，A[1]～A[N]，x の設定例を一つあげ，「正しく結果が出力されない」状況がどのようなものかを簡潔に説明せよ．

```
#include <stdio.h>
/*  この部分でNの値が設定される. */
int main(void)
{
    int A[N+1];
    int mid, left, right, x, index;
/* この部分に下記の処理が記述されているとする.
    ・配列A[1]～A[N]の値が設定される.
    ・変数xに探索すべき値が設定される.
    ・配列A[1]～A[N], xの値が妥当かどうか確認される.
*/
   left = 1;
   right = N;
   do {
          mid= (left+right)/2;
          printf("%d %d %d\n",left,mid,right);
          if (A[mid]<x) left = mid+1;
          else          right= mid-1;
    } while (A[mid] != x && left <= right);
    if (A[mid]==x) index=mid;
    else           index=0;
    printf("%d\n", index);
    return 0;
}
```

図 1　2 分探索法のプログラム

(2) 図 2 に示す ANSI-C 準拠である C 言語のプログラム（program）は，0-1 ナップサック問題（0-1 knapsack problem）の解を求めて表示するプログラムである．なお，図の左端の数値は行番号を表す．0-1 ナップサック問題は，ある容量（capacity）のナップサックが一つと，それぞれ大きさ（size）と価値（value）が定められた複数の品物が与えられた時，ナップサックの容量を超えない範囲で価値の和が最大となる品物の組み合わせを求める問題である．本プログラムでは，四つの品物のそれぞれに他と重複しない番号が与えられており，番号 i（1≦i≦4）の品物の大きさと価値が，配列 size の要素 size[i] と配列 value の要素 value[i] としてそれぞれ格納されている．なお，各番号の品物はそれぞれ一つしかなく，また，分割できない．品物の大きさと価値は自然数とする．以下の各小問に答えよ．

(2-1) 22 行目の処理を実行する直前の配列 sack の内容を，解答用紙の表の空欄を埋めることにより答えよ．

sack[i][j]	j=0	j=1	j=2	j=3	j=4	j=5	j=6	j=7	j=8	j=9	j=10	j=11	j=12	j=13	j=14	j=15
i=0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
i=1	0	-1	20	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
i=2	0	-1	20	-1	-1	-1	30	-1	50	-1	-1	-1	-1	-1	-1	-1
i=3																
i=4																

(2-2) 図 2 のプログラム中の空欄（ア），（イ）を適切な式（expression）で埋めよ．

```
1  #include <stdio.h>
2  #define capacity 15       /* ナップサックの容量 */
3  #define item 4            /* 品物の個数 */
4  int main(void) {
5    int size[] = {0, 2, 6, 6, 2};
6    int value[] = {0, 20, 30, 15, 25};
7    int sack[item+1][capacity+1];
8    int i, j, max, index;
9
10   for (i = 0; i <= item; i++)
11     for (j = 0; j <= capacity; j++)
12       sack[i][j] = -1;    /* ナップサックの初期化 */
13   sack[0][0] = 0;         /* 品物が入っていない(大きさの和が 0)のナップサックの総価値は 0 */
14
15   for (i = 1; i <= item; i++)
16     for (j = 0; j <= capacity; j++)
17       if (sack[i-1][j] != -1) {
18         if (sack[i-1][j] > sack[i][j]) sack[i][j] = sack[i-1][j];
19         if (j + size[i] <= capacity) sack[i][j+size[i]] = sack[i-1][j] + value[i];
20       }
21
22   max = 0; index = 0;
23   for (j = 0; j <= capacity; j++)
24     if (sack[item][j] > max) {max = sack[item][j]; index = j;}
25   for (i = item; i >= 1; i--)
26     if (index >= size[i] && [    (ア)    ]==[    (イ)    ]) {
27       printf("item %d is in a knapsack\n",i); index = index - size[i];
28     }
29   return 0;
30 }
```

図 2　0-1 ナップサック問題を解くプログラム

2 【必須問題】計算機システムとシステムプログラム

配点：(1-1) 5 点, (1-2) 40 点, (1-3) 15 点,
(2-1) 10 点, (2-2) 18 点, (2-3) 24 点, (2-4) 13 点

(1) 浮動小数点数 (floating point number) に関する以下の小問に答えよ．なお x を 2 進数 (binary number) として解釈する場合，$[x]_2$ と表記する．角括弧 (square bracket) をつけない場合は 10 進数と解釈する．

(1-1) 0.625 を 2 進数として表記せよ．

(1-2) IEEE 754 で規定された半精度浮動小数点数表現 (representation of half precision floating point number) において，以下の値に最も近い値を表すビット列を解答用紙に書け．

(a) $[1.101]_2$

(b) 5

(c) 0.125

(d) 0.1

(1-3) 2 進数の浮動小数点数表現では，0.1 を誤差なく有限桁で表現できない理由を述べよ．

資料： IEEE 754 半精度浮動小数点数表現 簡略版 (simplified version)

符号部 (sign) を 1 ビット，指数部 (exponent) を 5 ビット，仮数部 (significand) を 10 ビットで表現．指数部が表す符号無し整数を e, 仮数部のビット列を f とすると，この形式が表す値は

$$(-1)^{\text{符号部}} \times 2^{e-15} \times [1.f]_2$$

である．ただし，$e = 0, 31$ の時は上式をあてはめない非正規化数 (special value) として扱う．本問題では非正規化数を扱わないため，その説明を省略する．

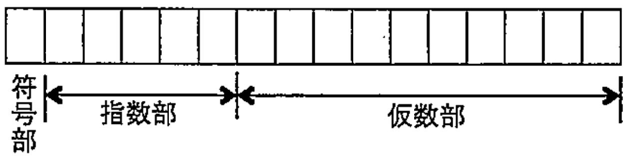

図: IEEE 754 半精度浮動小数点数表現

(2) メモリ管理に関する以下の小問に答えよ．解答は全て解答欄の太線内に書くこと．

(2-1) 計算機 (computer) においては，読み書きの速度 (reading and writing speed)，容量 (capacity) が異なる複数種類の記憶素子 (memory device) や記憶装置 (memory unit) を連携動作させて，高速な読み書きと大容量の両立を図る，記憶階層 (memory hierarchy) の考え方が導入されている．以下に示す記憶素子または記憶装置 (ア)～(オ) を有する一般的な計算機において，これらを，読み書きが速く，容量の少ない順に整列し，記号で答えよ．

(ア) 主記憶装置 (main memory unit)，(イ) 2 次キャッシュメモリ (secondary cache memory)，(ウ) 補助記憶装置 (auxiliary memory unit)，(エ) レジスタ (register)，(オ) 1 次キャッシュメモリ (primary cache memory)

(2-2) 以下の文章の空欄 (a)～(i) に当てはまる最も適切な語句を，下記の選択肢から選び，記号で答えよ．同じ選択肢を複数回用いても良い．

記憶階層の一部を成す仮想記憶 (virtual memory) では，主記憶装置に置くべきプログラムやデータ (以下では "情報" と呼ぶ) の一部を [(a)] に置く．これにより，主記憶装置の持つアドレス空間よりもより広いアドレス空間を有するように，ユーザープログラムに見せかける効果がある．機械語命令 (machine instruction) の中に記載されるアドレスは [(b)]，アドレスバスに送出されるアドレスは [(c)] と呼ばれる．

仮想記憶において，必要な情報が主記憶装置上に無く，[(d)] 上にある場合，この情報は主記憶装置に移動される．この動作は [(e)] と呼ばれる．この時，主記憶装置上に空き領域が無ければ，使用される可能性が低い情報が主記憶装置から [(f)] へ移動され，空き領域が作られる．この動作は [(g)] と呼ばれる．[(e)] や [(g)] が頻繁に発生すると，計算機の処理性能が低下する．

仮想記憶の代表的な実現方法として，メモリを固定長 (fixed size) のブロック (block) で管理する [(h)] 方式と，可変長 (variable size) のブロックで管理する [(i)] 方式とがある．

【選択肢】

(ア) レジスタ	(イ) スワップイン (swap in)	(ウ) フラグメンテーション (fragmentation)
(エ) キャッシュメモリ	(オ) 仮想アドレス (virtual address)	(カ) セグメンテーション (segmentation)
(キ) 補助記憶装置	(ク) スワップアウト (swap out)	(ケ) コンテクストスイッチ (context switch)
(コ) ページング (paging)	(サ) 実アドレス (real address)	(シ) 直接マッピング (direct mapping)

(2-3) ページ枠 (page frame) 数が 4 のページング方式を採用した仮想記憶を考える．以下に示すページ番号 (仮想アドレス空間におけるページの番号) の順にメモリ参照が起こった時，ページ置換アルゴリズム (page replacement algorithm) が LRU (least recently used) 法と FIFO (first-in-first-out) 法のそれぞれについて，主記憶装置上に置かれる各ページ枠が保持するページの番号を答えよ．なお，初期状態では全てのページ枠は空であるとし，解答欄に示されている前半の例を参考に，後半の太線内について解答すること．また，ページフォルト (page fault) が発生する場合はページ番号を○で囲むこと．

参照ページ番号：0, 1, 2, 3, 3, 4, 2, 1, 0, 5, 1, 2

(2-4) ページング方式を採用した仮想記憶を有する計算機で，主記憶装置に入り切らないサイズの整数型配列 (array) `A[W*H]` を扱うプログラムを考える．以下に示す C 言語で書かれた二つのプログラム断片 (ア)，(イ) の内，その処理時間が短くなる方を選び，記号で答えよ．また，その理由を仮想記憶と関連付けて説明せよ．なお，コンパイラ (compiler) による最適化は行われず，キャッシュメモリを有しない計算機で実行するものとする．

(ア)

```
for(x=0; x<W; x++) {
    for(y=0; y<H; y++) {
        sum = sum + x * A[x+(y*W)];
    }
}
```

(イ)

```
for(y=0; y<H; y++) {
    for(x=0; x<W; x++) {
        sum = sum + x * A[x+(y*W)];
    }
}
```

3 【選択問題】離散構造

配点：(1-1-1)〜(1-1-3) 各 12 点，(1-2-1) 10 点，(1-2-2) 10 点，(1-2-3) 15 点
(2-1-1)〜(2-1-3) 各 6 点，(2-2) 6 点，(2-3) 15 点，(2-4) 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，一階述語論理式（first-order predicate 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，全称作用素と存在作用素を併せて限定作用素（quantifier）と呼ぶ．$\Leftrightarrow$, $\rightarrow$, $\wedge$, $\vee$, $\neg$ 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）を表す論理演算子とする．特に断りのない限り，u, w, x, y, z で変数（variable）記号，a, b で定数（constant）記号，f で関数（function）記号，p, q で述語（predicate）記号を表わす．

(1-1) 一階述語論理式の解釈（interpretation）I は（D, C, F, P）の 4 項組で与えられる．ここで，D は値集合，C は各定数記号への D の要素への割り当て，F は各 n 引数関数記号への $\mathrm{D}^n \rightarrow \mathrm{D}$ の要素の割り当て，P は各 n 引数述語記号への $\mathrm{D}^n \rightarrow \mathrm{B}$ の要素の割り当てである．ただし，真（true），偽（false）を表わす true, false の 2 値からなる集合を B とする．

例えば，一階述語論理式 $\forall x\ p(f(b,x),a)$ に対して，解釈 I_0 として

- D を非負整数（nonnegative integer）全体からなる集合とし，
- C として a, b それぞれへ非負整数値 0, 1 を割り当て，
- F として 2 引数関数記号 $f(u,w)$ へ非負整数上の加算 $u+w$ を割り当て，
- P として 2 引数述語関数 $p(u,w)$ へ非負整数上の比較演算 $u > w$ を割り当てたとき（例えば $4>3$ の値は true である），

式 $\forall x\ p(f(b,x),a)$ の解釈 I_0 のもとでの評価値は true となる．

以下の (1-1-1)〜(1-1-3) の各論理式が，(a) 恒真（valid），(b) 充足可能（satisfiable）であるが恒真ではない，(c) 充足不能（unsatisfiable）のいずれであるか判断し記号で答えよ．また，(b) である場合は，D を非負整数（nonnegative integer）全体からなる集合として，真にする解釈と偽にする解釈をそれぞれ一つずつ挙げよ．なお，定数記号や関数記号が含まれないものは，それぞれ C や F は考えなくてよい．

(1-1-1) $(p(a) \wedge p(b)) \rightarrow \forall x\ p(x)$

(1-1-2) $\forall x\ (p(x) \vee q(x)) \rightarrow (\forall x\ p(x) \vee \forall x\ q(x))$

(1-1-3) $\neg\forall x\ \neg p(x) \Leftrightarrow \exists x\ p(x)$

(1-2) 以下で与えられる論理式 E が恒真（valid）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．

$A = \forall x\ \forall y\ (p(x,y) \rightarrow p(y,x))$

$B = \forall x\ \forall y\ \forall z\ ((p(x,y) \wedge p(y,z)) \rightarrow p(x,z))$

$C = \forall x\ \exists y\ p(x,y)$

$D = \forall z\ p(z,z)$

$E = (A \wedge B \wedge C) \rightarrow D$

以下の (1-2-1)〜(1-2-3) に答えよ．それぞれ導出過程も示すこと．なお，解答には記号 A,B,C,D を用いないこと．

(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が先頭にある閉論理式（closed formula）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) (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) (1-2-2) で得た論理式 $\neg E'$ をもとに，導出原理を用いて $\neg E'$ が充足不能であることを示せ．

(2) 有限集合（finite set）A について，A の要素数を $|A|$ と書く．A のべき集合（power set）を $P(A)$ として，$P(A)$ 上の 2 項関係（binary relation）$R_{P(A)}$ を

$$R_{P(A)} = \{(a,b) \in P(A) \times P(A) \mid (a \subseteq b) \wedge (\forall c \in P(A))[((a \subseteq c) \wedge (c \subseteq b)) \rightarrow ((c = a) \vee (c = b))]\}$$

とする．以下の各小問に答えよ．

(2-1) 任意の有限集合 A について，$R_{P(A)}$ の以下の各性質が成立するかどうか答えよ．証明は与えなくてもよい．

(2-1-1) 反射性 (reflexivity)

(2-1-2) 対称性 (symmetry)

(2-1-3) 反対称性 (antisymmetry)

(2-2) $|A| = 2$ の時，$|R_{P(A)}|$ を求めよ．

(2-3) $|A| = n$ の時，$|R_{P(A)}|$ を求めよ．

(2-4) $R_{P(A)}$ の反射的推移的閉包 (reflexive transitive closure) を $R^*_{P(A)}$ とする．$|A| = n$ の時，$|R^*_{P(A)}|$ を求めよ．

配点：(1-1) 10 点，(1-2) 15 点，(1-3) 20 点，(1-4) 15 点，
(2-1) 25 点，(2-2) 40 点

(1) 決定性有限オートマトン (deterministic finite automaton)M は 5 項組 $M = (Q, \Sigma, \delta, q_0, F)$ 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 (アルファベット (alphabet))，状態遷移関数 (state transition function)，初期状態 (initial state) ($q_0 \in Q$)，受理状態 (accepting state) の集合 ($F \subseteq Q$) である．また，M が受理する言語 (認識する言語) を $L(M)$ と表す．有限オートマトン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 $M = (Q, \Sigma, \delta, q_0, F)$ の状態 p, q ($p \in Q, q \in Q$) が区別不能 (indistinguishable) とは，任意の記号列 $w \in \Sigma^*$ に対し，$\hat{\delta}(p, w) \in F \Leftrightarrow \hat{\delta}(q, w) \in F$ が成り立つことであり，$p \simeq q$ と表す．ここで，Σ^* は，Σ 上の記号列すべての集合 (空系列を含む) を表す．また，任意の状態 $r \in Q$ と任意の記号列 $x \in \Sigma^*$ に対し，$\hat{\delta}(r, x) \in Q$ を以下のように定義する．

$x = \varepsilon$ のとき，$\hat{\delta}(r, x) = r$ 　(ε は空系列を表す)

$x = ya$ ($y \in \Sigma^*, a \in \Sigma$) のとき，$\hat{\delta}(r, x) = \delta(\hat{\delta}(r, y), a)$

状態集合 Q 上の 2 項関係 (binary relation) $\simeq$ が Q 上の同値関係 (equivalence relation) であることを証明せよ．

(1-2) 決定性有限オートマトン $M_1 = (Q_1, \{0, 1\}, \delta_1, a, \{a, b\})$ が右の状態遷移表 (state transition table) で与えられたとき，状態集合 $Q_1 = \{a, b, c, d, e, f, g, h, i\}$ が同値関係 $\simeq$ によって，どのような同値類 (equivalence class) に分割 (partition) されるか示せ．

	0	1
a	e	d
b	f	d
c	d	f
d	i	g
e	g	i
f	g	h
g	c	b
h	a	c
i	b	c

(1-3) $L(M_1) = L(M)$ となる決定性有限オートマトン M の中で，状態数最小のものを M_2 とする．M_2 を状態遷移図 (state transition diagram) で示せ．状態遷移図では，初期状態，受理状態それぞれが分かるように明示すること．

(1-4) $L(M_1) = L(M)$ となる決定性有限オートマトン M の中で，(1-3) で示した M_2 の状態数が最小であることを証明せよ．

(2) 文脈自由言語 (context-free language) に対して次の反復補題 (Pumping Lemma) が知られている．

反復補題 (Pumping Lemma)
文脈自由言語 L に対し，ある非負整数 n が存在し，L に属する n 以上の長さの任意の文字列 z に対して，ある部分文字列分解 $z = uvwxy$ が存在して，以下が成り立つ．

(i) $|vwx| \leq n$

(ii) $vx \neq \varepsilon$ 　　ただし ε は空系列

(iii) $i \geq 0$ なる任意の i に対して $uv^iwx^iy \in L$

以下の各小問に答えよ．解答用紙にはⓐ〜ⓞ とそれらに対応する解を列挙すること．

(2-1) 文脈自由言語 $L_A = \{a^m b^m \mid m \geq 5\}$ に対して反復補題が成り立つことを確かめたい．以下の文章の空白を埋め，L_A に対し反復補題が成り立つことを示せ．空白ⓐ〜ⓘ にはそれぞれ，指定された条件を満たす要素が入る．例えば $\boxed{\text{ⓐ 非負整数}}$ に対応する要素は非負整数である $1, 40, 100$ 等である．

n として $\boxed{\text{ⓐ 非負整数}}$ を選ぶ．$z = a^K b^K$ ただし，$K \geq$ $\boxed{\text{ⓑ 非負整数}}$ なる任意の文 z に対して $z = uvwxy$ なる部分文字列分解として $u = a^{K-1}$, $v =$ $\boxed{\text{ⓒ 記号列}}$, $w =$ $\boxed{\text{ⓓ 記号列}}$, $x =$ $\boxed{\text{ⓔ 記号列}}$, $y =$ $\boxed{\text{ⓕ 記号列}}$ を選ぶ．明らかに $|\, vwx \,| \leq n$ であり，$vx \neq \varepsilon$，および $i \geq 0$ なる任意の i に対して $\boxed{\text{ⓖ 記号列}}$ $\in L_A$ が成り立つ．なお L_A を生成する文脈自由文法 (context-free grammar) は $(\{S_1, A_1\}, \{a, b\}, P_1, S_1)$ であり (表記の意味は下記参考を参照)，ここで $P_1 = \{S_1 \to$ $\boxed{\text{ⓗ 記号列}}$ $, A_1 \to ab,$ $\boxed{\text{ⓘ 生成規則}}$ $\}$ である．

(2-2) 言語 $L_B = \{a^m b^m c^m \mid m \geq 0\}$ は文脈自由言語ではないことを反復補題を用いて証明したい．以下の証明の空白を埋めよ．空白ⓙ〜ⓞにはそれぞれ，(2-1) と同様指定された条件を満たす要素が入る．ただし，ⓚ, ⓛについては終端記号の数量的性質に言及すること．また，ⓞについては使用した反復補題の条件を明示すること．

証明

背理法（帰謬法;proof by contradiction）で証明する．言語 L_B が文脈自由文法であると仮定する．このとき仮定より L_B に対して反復補題が成り立つ．以下，反復補題が成り立たないことを示し，矛盾を導く．

n として K という非負整数値を考える．以降の議論は K をパラメータとして考えているので，任意の値を K に与えても成立する．L_B の定義より K を超える長さの文 z が存在する．具体的に $z = a^K b^K c^K$ を考える．z を $uvwxy$ に分割する際，すべての考え得る分割パターンに対して，上記の矛盾を説明する必要がある．

場合 1: vwx が終端記号 c を含まない場合：

このとき y は $\boxed{\text{ⓙ 記号列}}$ を部分文字列として含む．また反復補題の条件 (ii) が成り立つとすると，v, x について以下が成り立つ．

$\boxed{\text{ⓚ } v, x \text{ に関する条件}}$

したがって $i = 0$ とおいて L_B に含まれるはずの文 uwy を考えたとき，uwy は L_B に属さない．なぜならば

$\boxed{\text{ⓛ 証明}}$

すなわち反復補題の条件 (iii) を満たすことはできず，反復補題を満たさない．

場合 2: vwx が $\boxed{\text{ⓜ 終端記号に関する条件}}$ 場合：

対象の対称性 (symmetry) より場合 1 と同様である．

場合 3: その他の場合，すなわち vwx が $\boxed{\text{ⓝ 終端記号に関する条件}}$ 場合:

場合 3 はそもそも起こりえない．なぜならば

$\boxed{\text{ⓞ 証明}}$

いずれの場合であっても反復補題を満たさず，よって，矛盾を生じる．

以上の議論は仮定していた命題「言語 L_B は文脈自由文法である」が偽であることを示している． Q.E.D.

参考

文脈自由文法 (context-free grammar) G は 4 項組 $G = (N, T, P, S)$ 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 symbol) 集合，終端記号 (terminal symbol) 集合，生成規則 (production rule) 集合，開始記号 (start symbol) である．

配点: (1-1)6 点, (1-2)8 点, (1-3-1)4 点, (1-3-2)22 点,
(2-1)30 点, (2-2)20 点, (2-3)15 点, (2-4)20 点

(1) ビット誤り率 (bit error rate) p (ただし $0 < p < 1/2$) の 2 元対称通信路 (binary symmetric channel) における, 2 元符号 $C = \{00000, 11111\}$ を用いた通信を考える. C の各符号語は等確率で送信されるとする.

(1-1) 受信語における誤りの個数が 3 ビット, 4 ビット, 5 ビットになる確率を, それぞれ p を用いて表せ.

(1-2) 最尤復号法 (maximum likelihood decoding) を用いて (すなわち, 最大尤度基準に基づいて) 復号した場合の, 誤って復号する確率を, p を用いて表せ.

(1-3) 限界距離復号法 (bounded distance decoding) を用いて復号することを考える.

(1-3-1) 限界距離 d の最大値を求めよ.

(1-3-2) $p = 1/10$ とする. 誤って復号する確率を 1/1000 以下に抑えつつ, 正しく復号する確率を最大にするような限界距離 d を求めよ. また, そのときの, 正しく復号する確率を求めよ. 計算過程も示すこと.

(2) IP (Internet Protocol) の経路制御 (routing) に関する以下の各小問に答えよ.

(2-1) 経路制御プロトコルに関する以下の説明文の空欄 (あ) ～ (か) に当てはまる最も適切な語句を選択肢から選び, その番号を答えよ. なお, 異なる空欄には異なる語句が当てはまる.

IP アドレス (IP address) が割り当てられたホスト (host) がルーター (router) を介して接続される IP ネットワークにおいて, ルーターが経路情報 (routing information) を管理する手法として, (あ) と (い) がある. (あ) は, 経路情報をネットワークの各ルーターに手動で設定する手法である. これに対し, (い) は, 経路制御プロトコル (routing protocol) を用いてルーターが経路情報を交換および収集し, 経路を決定する手法であり, ホストやルーター等の機器の接続状況に変更が生じた際に経路情報が自動で更新される. インターネットの AS (Autonomous System) 内で用いられる経路制御プロトコルには, (う) 型プロトコルである OSPF (Open Shortest Path First) や, (え) 型プロトコルである RIP (Routing Information Protocol) がある. (う) 型プロトコルは, (え) 型プロトコルと比較して, (お) などの利点がある. 一方, AS 間で用いられる経路制御プロトコルとして (か) がある.

選択肢

(a) IS-IS protocol (Intermediate System to Intermediate System protocol)
(b) コスト最小 (cost-minimized)
(c) 距離ベクトル (distance vector)
(d) ブロードキャスト (broadcast)
(e) RSVP (Resource reSerVation Protocol)
(f) BGP (Border Gateway Protocol)
(g) IPIP (IP-within-IP encapsulation protocol)
(h) スタティックルーティング (static routing)
(i) 最適ルーティング (optimal routing)
(j) リンクステート (link state)
(k) プロトコルで交換される情報量が少ない
(l) 接続状況に変更が生じた際の経路収束時間が短い
(m) ダイナミックルーティング (dynamic routing)
(n) 経路計算量が少ない
(o) HTTP (Hyper Text Transfer Protocol)

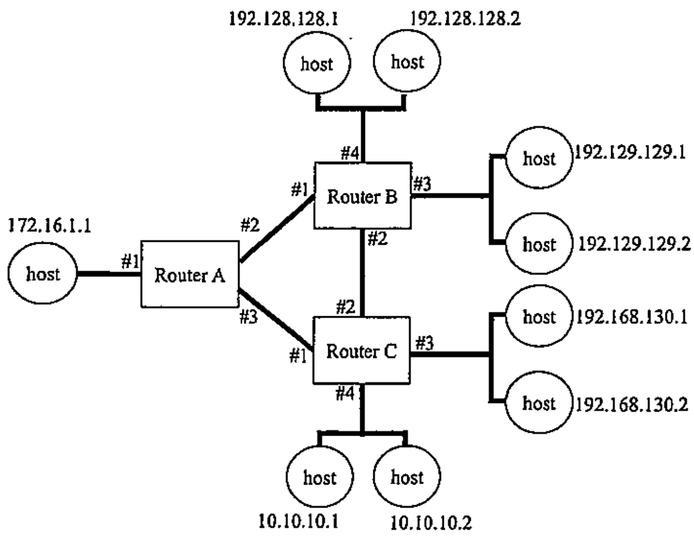

図 1

(2-2)　図 1 に示す IP ネットワークの構成を考える．図には，ホストの IP アドレスをドットアドレス表記で示している．また，#で示した番号はルーターのインターフェース番号である．クラスフルルーティング（classful routing）により経路情報を管理する場合に，ルーター A が保持する経路表（routing table）を表 1 にならって示せ．経路表のエントリー数が最小となるように記載すること．ただし，ホスト間の経路は経由するルーター数が最小となるよう設定されるものとする．また，デフォルトゲートウェイは考えないものとし，図に記載のない IP アドレスに対する経路エントリーは保持しないものとする．必要に応じて，クラスフルルーティング（classful routing）とクラスレスルーティング（classless routing）に関する以下の説明文を参考にすること．

IPv4 における IP アドレス長は 32 ビットであるため，理論上約 40 億の機器に IP アドレスを割当てることができるが，IP アドレス毎に経路情報を保持すると経路情報が肥大化する．そのため，IP アドレスをネットワークアドレス（network address）部とホストアドレス（host address）部に分割し，ネットワークアドレス部に対する経路情報を管理する方法が採られている．IP の経路制御プロトコルが導入された当初は，ネットワークアドレス部の長さを固定とするアドレスクラス（address class）に基づいた経路制御プロトコルが用いられていた．例えば IP アドレスの最上位から 3 ビットを 110 とするアドレスクラス C の IP アドレスは，上位 24 ビットがネットワークアドレス部，下位 8 ビット はホストアドレス部と定められている．同様に，最上位ビットを 0 とするアドレスクラス A は上位 8 ビットがネットワークアドレス部であり，最上位ビットから 2 ビットを 10 とするアドレスクラス B は上位 16 ビットがネットワークアドレス部となる．このようなアドレス規則にもとづく経路制御をクラスフルルーティングと呼ぶ．最近ではアドレスクラスを撤廃し，サブネットマスク（subnet mask）を用いて指定される任意のネットワークアドレス部の長さにもとづいて経路を制御するクラスレスルーティングが導入されている．

表 1

エントリー番号	宛先IPアドレス	サブネットマスク	出力先のインターフェース
1	172.16.0.0	255.255.0.0	#1
2			
3			
4			
5			
⋮	⋮	⋮	⋮

(2-3)　図 1 の IP ネットワークのルーター B, ルーター C に新たにインターフェース#5 をそれぞれ追加し，ルーター B のインターフェース#5 に H_B 台のホストを，ルーター C のインターフェース#5 に H_C 台のホストを新たに接続することを考える．H_B 台のホストに対して 192.130. で始まるアドレスクラス C のネットワークアドレスを新たに N_B 個割り当てた．また，H_C 台のホストに対しては，192.1. で始まるアドレスクラス C のネットワークアドレスを新たに N_C 個割り当てた．この時，1) ホスト毎に経路表のエントリーを用意する場合，2) クラスフルルーティングを行う場合，3) クラスレスルーティングを行う場合のそれぞれについて，ルーター A が保持するホストに対する経路表の最小エントリー数を求めよ．必要に応じて H_B，H_C，N_B，N_C を用いること．(2-2) と同様に，ホスト間の経路は経由するルーター数が最小となるよう設定されるものとする．また，デフォルトゲートウェイは考えないものとする．

(2-4)　図 1 のルーター C のインターフェース#1 とインターフェース#2 に対し，IP アドレス 192.168.130.1 宛のパケットが同時刻に到着する場合を考える．インターフェース#1 に到着するパケット X と，インターフェース#2 に到着するパケット Y の二つのパケットが同時刻に到着してから出力先のインターフェースから送り出されるまでに，ルーターが到着パケットに対して行う処理を簡潔に説明せよ．ただし,「経路表」,「バッファ」(buffer),「競合回避」(contention resolution) の語句を用いて説明すること．

6 【選択問題】電子回路と論理設計

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

(1) 次の性質を持つ同期式カウンタ(synchronous counter)について，以下の各小問に答えよ．

1 個の入力(input) x と 6 個の状態(state)をもち，入力 $x=0$ のときはクロックが入力される毎に状態が $S0 \rightarrow S1 \rightarrow S2 \rightarrow S3 \rightarrow S4 \rightarrow S5 \rightarrow S0$ と遷移(transition)し，$x=1$ のときはクロックが入力される毎に状態が $S0 \rightarrow S2 \rightarrow S4 \rightarrow S0$ と遷移するカウンタを考える．ただし，初期状態(initial state)は S0 とし，入力 $x=1$ で状態が S1, S3, S5 のいずれかのときは，それぞれ S2, S4, S0 に遷移するものとする．本カウンタを図 1 に示すように 3 個の D フリップフロップ(D flip-flop)と論理ゲート(logic gate)を使って順序回路(sequential circuit)として実現する．D フリップフロップの出力を Q_i $(i=0,1,2)$ と表す．各状態は表 1 のように割り当てるものとする．また，入力 x はクロック CLK に同期して遅れなく入力されるものとし，初期化時にリセット信号 RST が入力されるとすべての D フリップフロップの出力 Q_i $(i=0,1,2)$ は0になるものとする．

(1-1) 本カウンタの状態遷移図(state transition diagram)を作成せよ．

(1-2) 状態 (Q_2, Q_1, Q_0) の次状態(next state)を (Q_2^+, Q_1^+, Q_0^+) で表す．状態割当付きの状態遷移表を作成し，x, Q_2, Q_1, Q_0 を変数とする 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．

(1-3) (1-2)で求めた最簡積和形の論理式を用いて，図 1 の順序回路のうち破線内の回路を最小のゲート数で実現せよ．解答用紙には，図 1 全体を記入すること．ただし，論理ゲートは，2 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．

(1-4) (1-3)で設計した順序回路の最大動作周波数(maximum operating frequency)を求めよ．設計に用いた 2 入力 NAND ゲートの 1 段分の遅延時間(delay time)を T_N [s]，D フリップフロップのセットアップ時間(setup time)を T_S [s]とせよ．ただし，それ以外の時間は考慮しなくてもよいものとする．

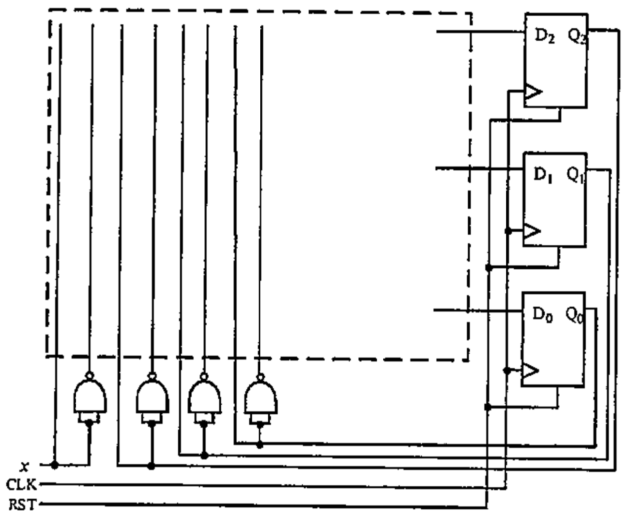

図 1

表 1

状態	状態割り当て		
	Q_2	Q_1	Q_0
S0	0	0	0
S1	0	0	1
S2	0	1	1
S3	1	1	1
S4	1	1	0
S5	1	0	0

(2) MOSFET（Metal-Oxide-Semiconductor Field-Effect Transistor）に関する以下の各小問に答えよ．

(2-1) 図 2 は n-MOSFET の構造を示している．また，図 3 は n-MOSFET を用いた回路の例である．以下の文中の空欄(a)～(f)を適切な語句で埋めよ．

図中の(A)の部分は $\boxed{\text{(a)}}$ 型半導体（semiconductor）で，(B)の部分は $\boxed{\text{(b)}}$ 型半導体でできている．ゲート（gate）電極は酸化膜（oxide）によって基盤（substrate/bulk）とは絶縁されている．Vg はゲート電圧，Vd はドレイン（drain）電圧である．$Vg = 0$ のとき，ソース（source）・ドレイン間は絶縁され，電流が流れない．しかし，Vg を $\boxed{\text{(c)}}$ よりも大きくすると，基盤と酸化膜の境界に $\boxed{\text{(d)}}$ が誘起され，$\boxed{\text{(e)}}$ が形成されて電流が流れるようになる．したがって，Vg を制御することによって，オンオフ・スイッチとして動作させることができる．さらに，極性（polarity）が逆の p-MOSFET と組み合わせて図 4 のような回路を構成すると，$\boxed{\text{(f)}}$ の機能を持つ論理ゲートを実現できる．

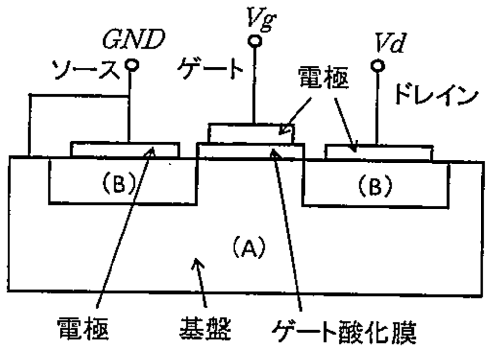

図 2

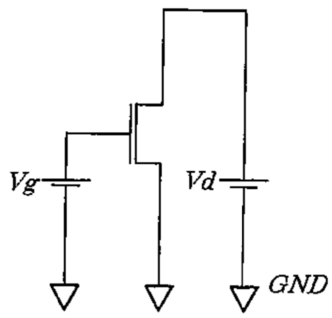

図 3

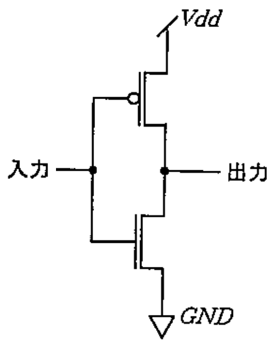

図 4

(2-2) 以下の図 5 および図 6 に示す回路それぞれについて，入力 A 及び B に電位 Vdd あるいは電位 0 が印加されたときの出力 X の電位を求めよ．解答は表 2 の形式で記せ．ただし，オン状態の MOSFET における電圧降下はないものとする．

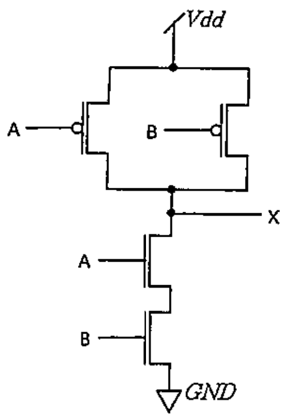

図 5

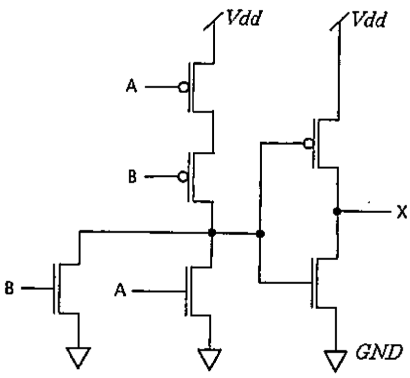

図 6

表 2

A	B	X
0	0	
0	Vdd	
Vdd	0	
Vdd	Vdd	

7 【選択問題】数学解析と信号処理

配点: (1-1)30 点, (1-2)35 点, (2)35 点, (3)25 点

以下の各問に答えよ.

(1)

(1-1) 以下のベルヌーイ型の微分方程式 (Bernoulli differential equation) に対して適当な変数変換 (variable transformation) を適用し，線形微分方程式 (linear differential equation) に変換せよ．ただし n は実数であり，$n \neq 0,1$ とする.

$$\frac{dx}{dt} + P(t)x = Q(t)x^n$$

(1-2) (1-1) の式において，$P(t) = \dfrac{2}{t}$, $Q(t) = -t^2\cos(t)$, $n = 2$ であるときの $x(t)$ を求めよ.

(2)

留数定理 (residue theorem) を用いて，以下の定積分 (definite integral) I を求めよ.

$$I = \int_{-\infty}^{\infty} \frac{dx}{x^2 + x + 1}$$

(次ページへ続く)

(3)

図1と図2に示すサンプリング周波数が $1/T$ である以下の離散時間信号 (discrete-time signal) $x(nT)$ と $y(nT)$ を考える．

$$x(nT) = \{x(0T) = 2, x(1T) = -3, x(2T) = 1, x(3T) = -2, x(4T) = -1, x(5T) = 0\}$$
$$y(nT) = \{y(0T) = 0, y(1T) = 2, y(2T) = -3, y(3T) = 1, y(4T) = -2, y(5T) = -1\}$$

ただし，n は整数であり，$n < 0$ または $n > 5$ のとき $x(nT) = 0$，$y(nT) = 0$ とする．

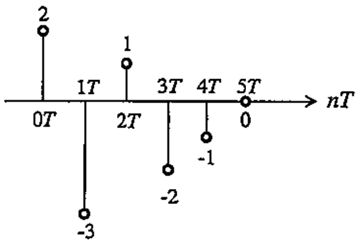

図 1: 離散時間信号 $x(nT)$

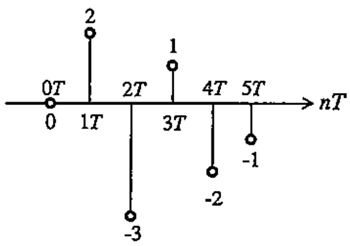

図 2: 離散時間信号 $y(nT)$

以下の文章の空欄 (a) から (e) を適切な語句または数式で埋めよ．ただし，解答用紙には (a) から (e) とそれらに対応する解の組を列挙すること．

図1と図2から，$y(nT)$ は $x(nT)$ に対して $\boxed{\text{(a)}}$ サンプル分だけ遅れている (delay) ことが分かる．一方，$x(nT)$ と $y(nT)$ の Z 変換 (Z-transform) はそれぞれ，$X(z) = \boxed{\text{(b)}}$，$Y(z) = \boxed{\text{(c)}}$ となる．この結果から，$X(z)$ と $Y(z)$ には

$$Y(z) = \boxed{\text{(d)}}\ X(z)$$

の関係があることが分かる．$\boxed{\text{(a)}}$ サンプル分の遅延は，上式の $\boxed{\text{(d)}}$ に対応しており，k サンプル分の遅延は $\boxed{\text{(e)}}$ の乗算に相当する．これは Z 変換の時間シフト (time shifting) に関する性質によるものである．