

令和 5 年 7 月 29 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 6 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。
- 問題用紙は表紙を除いて 14 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。

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

ANSI-C 準拠である C 言語のプログラム(program) 1, 2 は, データ(data)を読み込んで, 要素を昇順(ascending order)に整列(sort)して出力(output)する. 入力(input)するデータは, 1 行目に整列する要素の個数  $n$  ( $n \geq 1$ ) が, また 2 行目から  $n + 1$  行目までに整列する要素として整数(integer)が各行(each line)に並べられた構造を持つ. データは, ファイル data.txt に記述されている. 以下の各問に答えよ.

(1) 図 1 に示すプログラム 1 に関して, 以下の各小問に答えよ.

- (1-1) 図 2 の data.txt からデータを読み込んでプログラム 1 を実行(execute)した場合に, プログラム 1 の 20 行目が 3 回目に実行された直後の `array[0]`, `array[1]`, `array[2]`, `array[3]`, `array[4]` の値をそれぞれ示せ.
- (1-2) 図 2 の data.txt の 2~6 行目を並べ替え, プログラム 1 の 18 行目の実行回数が最大となる data.txt を示せ.
- (1-3) 整列する要素の個数を  $n$  とするとき, プログラム 1 の最悪時間計算量(worst case time complexity)のオーダー表記(order notation)として最も適しているものを下記選択肢から一つ選び, 簡潔に理由も示せ.

$O(1)$	$O(\log n)$	$O(n)$	$O(n \log n)$	$O(n^2)$	$O(n^2 \log n)$	$O(n^3)$
--------	-------------	--------	---------------	----------	-----------------	----------

(2) プログラム 1 を部分的に改変しプログラム 2 を作成したい. 以下の各小問に答えよ. ただし, data.txt は図 2 に限らないものとする.

- (2-1) 探索を高速化させるため, プログラム 1 の 5~15 行目のみを図 3 の通り変更したプログラム 2 を作成する. 空欄 (A) ~ (C) を適切な式で埋めて, プログラム 2 を完成させよ.
- (2-2) 整列する要素の個数を  $n$  とする. 完成したプログラム 2 の最悪時間計算量のオーダー表記として最も適しているものを下記選択肢から一つ選び, 簡潔に理由も示せ.

$O(1)$	$O(\log n)$	$O(n)$	$O(n \log n)$	$O(n^2)$	$O(n^2 \log n)$	$O(n^3)$
--------	-------------	--------	---------------	----------	-----------------	----------

- (2-3) 完成したプログラム 2 の実行を開始してから終了するまでの間に, 図 3 の 10 行目が実行される回数を  $c$  とする.  $c$  は整列する要素の個数  $n$  と並び順に依存する. 整列する要素が全て異なるとき,  $c$  が最小となる要素の並び順が満たす要件を一つ示せ.
- (2-4) (2-3)において,  $c$  の最小値を  $n$  の式で示せ. 導出過程も示すこと.

(次ページへ続く)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void sort (int array[], int n) {
5     int i, j, key, target;
6
7     for (i = 1; i < n; i++) {
8         key = array[i];
9
10        for (j = 0; j < i; j++) {
11            if(array[j] > key) {
12                break;
13            }
14        }
15        target = j;
16
17        for (j = i - 1; j >= target; j--) {
18            array[j + 1] = array[j];
19        }
20        array[target] = key;
21    }
22 }
23
24 int main (void) {
25     int N, i;
26     FILE *fp;
27     int *data;
28
29     fp = fopen("data.txt", "r");
30     fscanf(fp, "%d", &N);
31     data = (int*) malloc(sizeof(int) * N);
32     for(i = 0; i < N; i++) {
33         fscanf(fp, "%d", &data[i]);
34     }
35     fclose(fp);
36
37     sort(data, N);
38
39     for(i = 0; i < N; i++) {
40         printf("%d\n", data[i]);
41     }
42     free(data);
43     return 0;
44 }
```

図1 プログラム1

1	5
2	8
3	4
4	5
5	1
6	2

図2 data.txt

(次ページへ続く)

```
1  int i, j, key, left, right, mid, target;  
2  
3  for (i = 1; i < n; i++) {  
4      key = array[i];  
5      left = 0;  
6      right = i - 1;  
7  
8      while (left <= right) {  
9          mid = (left + right) / 2;  
10         if (array[mid] > key) {  
11             right = (A);  
12         } else {  
13             left = (B);  
14         }  
15     }  
16     target = (C);
```

図3 プログラム2の一部

配点： (1) 15, (2-1) 65, (2-2-1) 30, (2-2-2) 15

メモリ管理方式 (memory management method) に関する以下の各問に答えよ。

- (1) 以下の説明文において、( A ) に入る適切な値を求め、[ X ] を適切なメモリ管理方式の名称で埋めよ。

オーバーレイ (overlay) は必要なメモリ容量 (memory capacity) がメインメモリ容量 (main memory capacity) より大きなプログラム (program) を動かすメモリ管理方式である。オーバーレイの動作を見るために、プログラムモジュール (program module) P0, P1, P2, P3, P4 から構成されるプログラム P を、オーバーレイを用いて動作させる場合を想定する。P0, P1, P2, P3, P4 が実行時に必要とするメモリ容量は、それぞれ 2.0 [キロバイト] (Kbyte), 0.6 [キロバイト], 1.7 [キロバイト], 4.1 [キロバイト], 0.2 [キロバイト] となっている。プログラム P の動作において同時に動作するプログラムモジュールの組み合わせは、(P0, P1), (P0, P1, P2, P4), (P0, P2, P4), (P0, P3) の 4 通りのみとする。オーバーレイを用いて、プログラム P を動作させるのに最低必要なメモリ容量は ( A ) [キロバイト] となる。

オーバーレイの考え方を発展させた方式として仮想記憶がある。仮想記憶の典型的な方式には、仮想アドレス空間を固定長 (fixed length) で区切るページング (paging) や、可変長 (variable length) で区切る [ X ] が存在する。

- (2) ページングの動作を見るために、ページ枠の内容は「空」で初期化され、ページ参照列 (reference string) は 5 種類の参照 (reference) 0, 1, 2, 3, 4 のみで構成されるページングを用いた仮想記憶を想定する。ページ参照列 R は (0, 1, 2, 1, 3, 2, 4, 2, 0, 1, 2, 3) という順序の 12 回の参照から構成される。以下の各小問に答えよ。

- (2-1) 以下の説明文において、( B ), ( C ), ( D ), ( E ) に入る値を求め、[ Y ] に入る最も適切な内容を a)~d) より一つ選べ。

Belady の異常 (Belady's anomaly) の発生を調べるために、ページ置き換えアルゴリズムとして FIFO (first-in first-out) を用い、ページ枠の数を 2 から 3, 4, 5 と増加させて、ページ参照列 R のページフォールト率を求めた。ページ枠 (page frame) の数が 5 におけるページフォールト率 (page fault rate) は ( B ) % であり、ページ枠の数が 2, 3, 4 のとき、それぞれのページフォールト率は ( C ) %, ( D ) %, ( E ) % となった。

Belady の異常は [ Y ]

- 確認できなかった。
- ページ枠の数を 2 から 3 に増やしたときに確認できた。
- ページ枠の数を 3 から 4 に増やしたときに確認できた。
- ページ枠の数を 4 から 5 に増やしたときに確認できた。

(次ページへ続く)

(2-2) ページ置き換えアルゴリズム実行開始時に全てのページ参照列がわかっているとき、ページ置き換えアルゴリズムを工夫すると、与えられたページ枠の数とページ参照列に対する最小のページフォールトの回数を求めることができる。このようなページ置き換えアルゴリズムを $\alpha$ とする。以下の(2-2-1)と(2-2-2)に答えよ。

(2-2-1) ページ置き換えアルゴリズム $\alpha$ を用いるとき、ページ参照列 $R$ 、ページ枠の数が3におけるページフォールトの回数を求めよ。また、ページ置き換えアルゴリズム $\alpha$ において、どのようなページを最優先の置き換え対象とするか簡潔に述べよ。

(2-2-2) ページ参照列 $S$ は、ページ参照列 $R$ の2ヶ所にそれぞれ一つずつ参照が挿入されたページ参照列である。ページ枠の数が3のとき、ページ参照列 $S$ をページ置き換えアルゴリズムLRU (least recently used) で処理した場合のページフォールトの回数は、ページ参照列 $R$ をページ置き換えアルゴリズム $\alpha$ によって処理したページフォールトの回数と同じであった。ページ参照列 $S$ の例を一つ示せ。

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

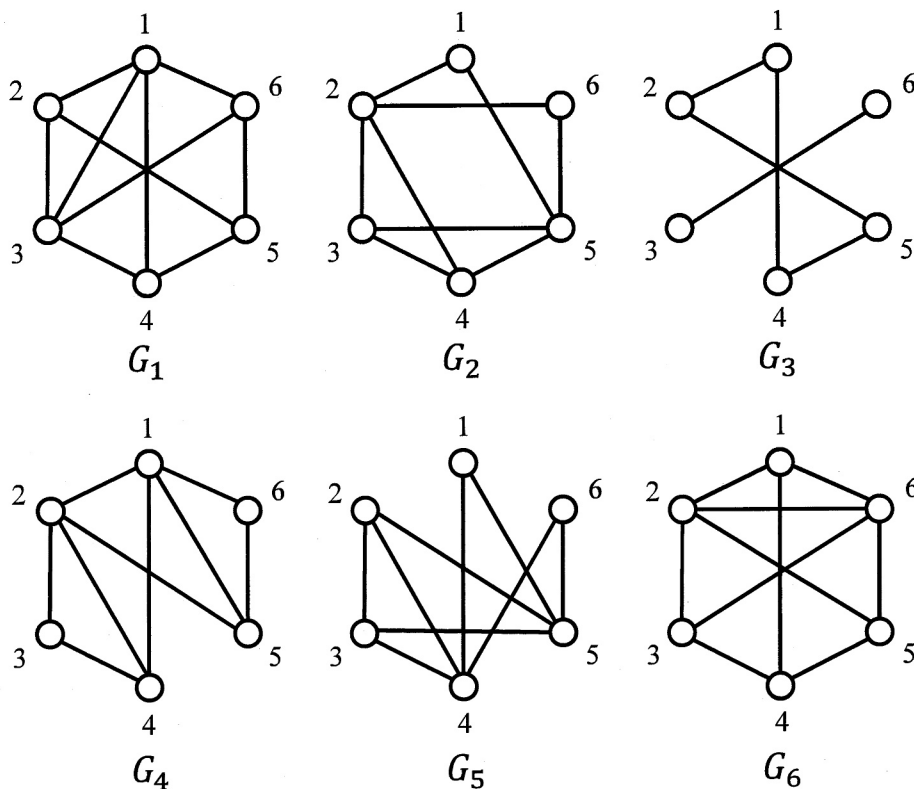
本問題で取り扱われるすべてのグラフ (graph) は無向グラフ (undirected graph) であり, 多重辺 (parallel edge) や自己ループ (self loop) を持たないものとする. グラフ  $G$  は有限 (finite) の頂点集合 (vertex set)  $V$ , および異なる頂点の非順序対 (unordered pair) の集まりである辺集合 (edge set)  $E$  の対 (pair) により  $G = (V, E)$  と表される. 二つのグラフ  $G_1 = (V_1, E_1)$  と  $G_2 = (V_2, E_2)$  が与えられたとき,  $V_1 \neq V_2$  または  $E_1 \neq E_2$  が成立するならば, かつそのときに限り,  $G_1$  と  $G_2$  を異なる (distinct) グラフと見なす.

二つのグラフ  $G_1 = (V_1, E_1)$  および  $G_2 = (V_2, E_2)$  に対して, 以下の条件をともに満たす写像 (mapping)  $\varphi : V_1 \rightarrow V_2$  が存在するとき,  $G_1$  と  $G_2$  は同型 (isomorphic) であると呼ぶ.

- $\varphi$  は全単射 (bijective).
- 任意の  $u, v \in V_1$  について,  $\{u, v\} \in E_1$  ならば, かつそのときに限り  $\{\varphi(u), \varphi(v)\} \in E_2$ .

$\mathcal{G}$  をすべてのグラフの集合 (set) とする. グラフ  $G_1, G_2 \in \mathcal{G}$  が同型であることを,  $\mathcal{G}$  上の二項関係 (binary relation) を表す記号  $\simeq$  を用いて  $G_1 \simeq G_2$  と記述するものとする. また, 正整数 (positive integer)  $n$  について,  $\mathcal{G}_n$  を頂点集合が  $\{1, 2, \dots, n\}$  であるようなすべてのグラフの集合とする. 以下の各問に答えよ.

(1)  $\mathcal{H}$  を, 以下に図示するグラフ  $G_1, G_2, \dots, G_6$  からなる  $\mathcal{G}_6$  の部分集合 (subset) とする.



- (1-1)  $G_i \simeq G_j$  かつ  $i < j$  を満たす対 (pair)  $(i, j)$  をすべて挙げよ.
- (1-2)  $\simeq$  は同値関係 (equivalence relation) であるため,  $\mathcal{H}$  を  $\simeq$  に基づく複数の同値類 (equivalence class) に分割することができる. それらすべての同値類からなる集合  $\mathcal{H}/\simeq$  の要素数 (number of elements) を答えよ.

- (2) 部分集合  $\mathcal{I}_n \subseteq \mathcal{G}_n$  を, 任意の異なる  $G, H \in \mathcal{I}_n$  について  $G \not\sim H$  が成立するような部分集合で要素数が最大のものと定義する (もしそのような条件を満たす部分集合が複数ある場合, そのなかの任意の一つが  $\mathcal{I}_n$  として選ばれているものとする). 有限集合 (finite set)  $X$  に対して, 記法  $|X|$  は集合の要素数を表すものとする.
- (2-1)  $|\mathcal{I}_3|$  の値はいくつになるか. 理由とともに答えよ.
- (2-2) 任意の  $G \in \mathcal{G}_n$  について, 高々  $n!$  個の異なるグラフ  $G' \in \mathcal{G}_n$  が  $G \simeq G'$  を満たすことを証明せよ.
- (2-3)  $|\mathcal{G}_n|$  の値はいくつになるか. 理由とともに答えよ.
- (2-4) 以下の不等式 (inequality) が成立することを証明せよ.

$$\log_2 |\mathcal{I}_n| \geq \frac{n(n-1)}{2} - n \log_2 n$$



配点 : (1-1) 35, (1-2) 20, (2) 15, (3) 15, (4) 40

文脈自由文法 (context-free grammar)  $G = (V, T, P, S)$  は, 変数 (variable) の有限集合 (finite set)  $V$ , 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)  $S$  により定められる. 変数は非終端記号 (nonterminal symbol) と呼ばれることもある. 頭部 (head) が同じ生成規則は, 本体 (body) を “|” で区切って列挙できる. 例えば,  $S \rightarrow A$  および  $S \rightarrow B$  は,  $S \rightarrow A | B$  のように記述できる.

生成規則の形を  $X \rightarrow YZ$ ,  $X \rightarrow a$  に限定した文脈自由文法を, Chomsky 標準形 (Chomsky normal form) と呼ぶ. ただし,  $X, Y, Z$  は変数,  $a$  は終端記号である. Chomsky 標準形で与えられる文脈自由文法  $G = (V, T, P, S)$  および終端記号列  $w = a_1 a_2 \cdots a_n$  ( $n \geq 1$  とし, 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どうかを判定する以下のアルゴリズム (algorithm) を考える.

```

1:  $w = a_1 a_2 \cdots a_n$  に対し, 2次元配列  $Q$  の要素である集合  $Q[i][j] (1 \leq i \leq j \leq n)$  をすべて空集合  $\emptyset$  に初期化.
2: for  $i = 1$  to  $n$  do
3:    $(X \rightarrow a_i) \in P$  となるすべての  $X \in V$  を  $Q[i][i]$  に追加.
4: end for
5: for  $l = 2$  to  $n$  do
6:   for  $i = 1$  to  $n - l + 1$  do
7:      $j$  を  $i + l - 1$  とする.
8:     for  $k = i$  to  $j - 1$  do
9:        $Y \in Q[i][k], Z \in Q[k + 1][j]$  であるすべての  $Y, Z$  に対し,  $(X \rightarrow YZ) \in P$  であるすべての  $X \in V$  を  $Q[i][j]$  に追加.
10:    end for
11:  end for
12: end for
13: if  then “Yes” を出力.
14: else “No” を出力.

```

以下の各問に答えよ.

- (1) 文脈自由文法  $G_1 = (V_1, T_1, P_1, S)$  を,  $V_1 = \{S, A, B, C\}$ ,  $T_1 = \{a, b, c\}$ ,  $P_1 = \{S \rightarrow aSb \mid A, A \rightarrow AB \mid AC \mid a \mid \varepsilon, B \rightarrow BB, C \rightarrow c\}$  と定める. ただし,  $\varepsilon$  は空列 (empty string) を表す.
  - (1-1)  $L(G)$  を, 文法  $G$  により生成される言語 (language) とする.  $L(G_2) = L(G_1) - \{\varepsilon\}$  を満たし, 無用な (useless) 記号を含まない Chomsky 標準形の文法  $G_2$  を示せ.
  - (1-2) 文法  $G_2$  および終端記号列  $w = acb$  に対して, 上記アルゴリズム 13 行目の開始時点での各集合  $Q[i][j]$  ( $1 \leq i \leq j \leq 3$ ) を示せ.
- (2)  $i, j$  を  $1 \leq i \leq j \leq n$  を満たす任意の整数とする. 上記アルゴリズム 13 行目の開始時点で集合  $Q[i][j]$  に含まれる変数はある特定の性質を持つ.  $w = a_1 a_2 \cdots a_n$  として, この性質を  $a_i$  と  $a_j$  を用いて説明せよ.
- (3) 上記アルゴリズム 13 行目の空欄  $\alpha$  は,  $w$  が  $G$  によって生成されるか否かを判定するための条件である. 空欄  $\alpha$  を埋めよ.
- (4) Chomsky 標準形の任意の文法  $G$  の構文木 (parse tree) について考える.  $n$  個の葉 (leaf node) を持つ構文木が, ちょうど  $3n - 1$  個の頂点 (node) を持つことを証明せよ.

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

媒体アクセス制御 (media access control, medium access control) に関する以下の各問に答えよ。

- (1) 複数の端末 (terminal) が同一の伝送 (transmission) 媒体を共有する際の媒体アクセス制御の方式である CSMA (Carrier Sense Multiple Access) 方式と CSMA/CD (Collision Detection) 方式に関する以下の説明文中の空欄 (A) と空欄 (B) にあてはまる最も適切な語句を、以下の(ア)~(エ)からそれぞれ選べ。また、空欄 (X) を適切な文で埋めよ。

- (ア) アイドル (idle) になるのを待って
- (イ) しばらく待って
- (ウ) 送信先の端末からの指示を待って
- (エ) 直ちに

CSMA 方式では、端末は、送信したいフレーム (frame) が発生すると、キャリアセンス (carrier sense) により伝送媒体の状態を確認する。伝送媒体がアイドルであれば、送信したいフレームを (A) 送信する。伝送媒体がビジー (busy) であれば (B) 再度キャリアセンスを繰り返す。CSMA/CD 方式では、さらに、端末がフレームを送信している間もキャリアセンスを行い、(X)。

- (2) CSMA 方式では、伝搬遅延時間 (propagation delay) が長くなると、単位時間 (unit time) あたりに衝突 (collision) なしに端末で受信されるフレーム数が減少する。その理由を説明せよ。

(次ページへ続く)

(3) 図1に示す無線通信 (wireless communication) 環境における媒体アクセスを考える。図1では、無線端末 (wireless terminal) 1と無線端末2は、ちょうどその中間に配置された無線端末3にフレームを送信するとし、それぞれの無線端末のフレームを受信できる距離は無線端末1と無線端末2の間の距離よりも短いものとする。以下の各小問に答えよ。

(3-1) 図1の環境において、キャリアセンスを行う媒体アクセス制御方式を用いた時に生じる隠れ端末問題 (hidden terminal problem) を、キャリアセンスとフレーム衝突の二つの用語を用いて説明せよ。

(3-2) 図1の環境において、隠れ端末問題に対処する媒体アクセス制御の一つについて、その手順を説明せよ。

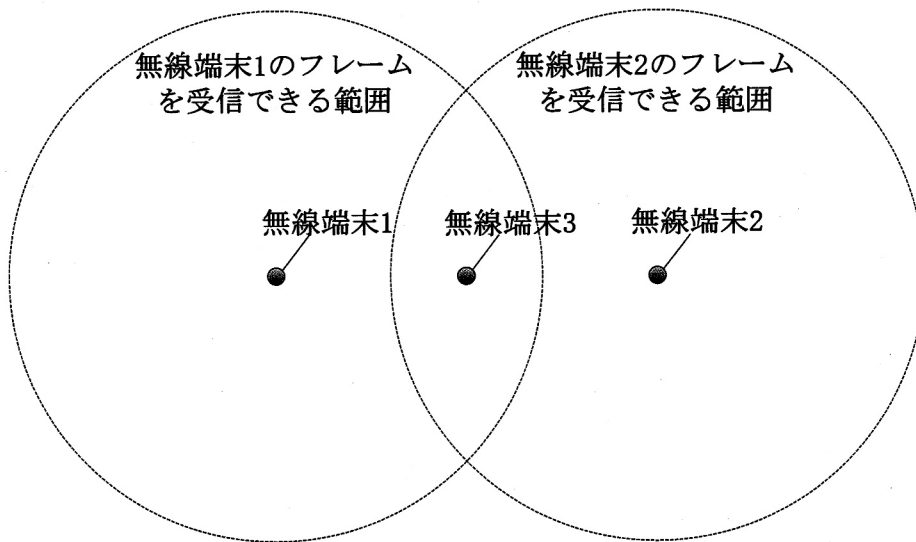
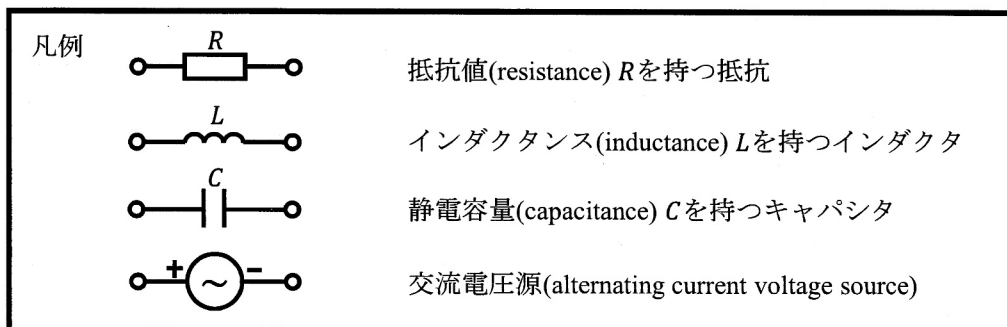


図1

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

- (1) 定常状態(steady state)にある交流回路(alternating current circuit)に関する以下の各小問に答えよ。図中の記号は以下の凡例に従う。交流電圧源の角周波数(angular frequency)を $\omega$ として答えよ。



- (1-1) 図1に示す端子対1—1'と端子対2—2'からなる2端子対回路(two-port network)を考える。 $V_1, I_1$ はそれぞれ端子対1—1'の電圧, 電流であり,  $V_2, I_2$ はそれぞれ端子対2—2'の電圧, 電流である。2端子対回路が受動素子(passive element)のみからなり, 端子対の電圧と電流を $\begin{bmatrix} V_1 \\ I_1 \end{bmatrix}, \begin{bmatrix} V_2 \\ I_2 \end{bmatrix}$ とベクトル表記する場合,  $\begin{bmatrix} V_1 \\ I_1 \end{bmatrix}, \begin{bmatrix} V_2 \\ I_2 \end{bmatrix}$ の関係を式(A)と式(B)で表現できる。式(A), 式(B)中の $F$ を縦続行列(transmission matrix)と呼ぶ。 $A_F, D_F$ は無次元であり,  $B_F$ はインピーダンス(impedance)の次元を持ち,  $C_F$ はアドミタンス(admittance)の次元を持つ。(1-1-1), (1-1-2), (1-1-3), (1-1-4)に答えよ。

$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = F \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \quad \dots (A) \qquad F = \begin{bmatrix} A_F & B_F \\ C_F & D_F \end{bmatrix} \quad \dots (B)$$

- (1-1-1) 図2に示す2端子対回路の縦続行列を $\omega, C$ の中から必要な項を用いて表せ。

- (1-1-2) 図3に示す2端子対回路の縦続行列を $\omega, L$ の中から必要な項を用いて表せ。

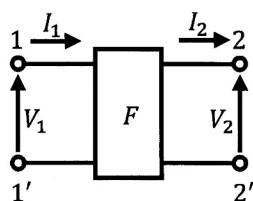


図1

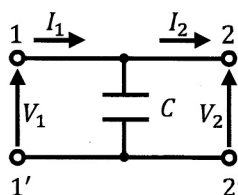


図2

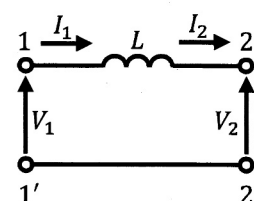


図3

(次ページへ続く)

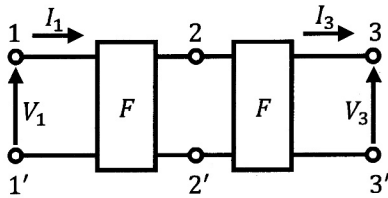


図4

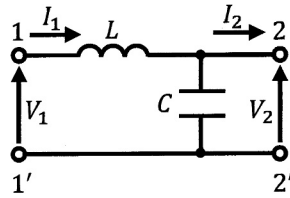


図5

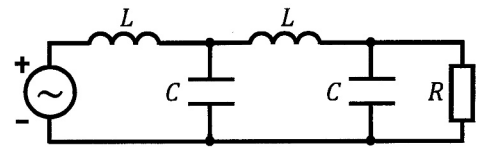


図6

(1-1-3) 図1に示す2端子対回路を2つ接続した回路を図4に示す. 図4の端子対1—1'と端子対3—3'からなる2端子対回路の縦続行列を $A_F, B_F, C_F, D_F$ を用いて表せ.

(1-1-4) 図5に示す2端子対回路の縦続行列を $\omega, L, C$ の中から必要な項を用いて表せ.

(1-2) 図6に示す回路について, 交流電圧源の電圧振幅の実効値は $E$ である. 図6の抵抗で消費される平均電力を $P$ とすると,  $E, P$ に関して式(C)が成り立つ. 式(C)中の空欄  $x$  と  $y$  に記入すべき式を,  $R, P, E$ を用いずに  $\omega, L, C$ から必要な項を用いて表せ.

$$\frac{E^2}{P} = \left\{ (1 + \boxed{x})^2 R + \frac{\omega^2 L^2 (2 - \boxed{y})^2}{R} \right\} \dots \quad (C)$$

(2)  $(x_4, x_3, x_2, x_1, x_0)$ を,  $x_4$ を最上位ビット(most significant bit),  $x_0$ を最下位ビット(least significant bit)とする5ビットの2進数として定義する.  $(x_4, x_3, x_2, x_1, x_0)$ を入力した時に, 1ビットの2進数を出力する論理関数を $f(x_4, x_3, x_2, x_1, x_0)$ とする.  $f(x_4, x_3, x_2, x_1, x_0)$ の出力は, 10進数で表現した入力の集合 $\{0, 8, 12, 16, 20, 30, 31\}$ に対しては1となり,  $\{1, 4, 9, 15, 17, 24, 25, 28\}$ に対してはドントケア(don't care)となり, その他に対しては0となる. 以下の各小問に答えよ.

(2-1)  $f(x_4, x_3, x_2, x_1, x_0)$ のカルノー図(Karnaugh diagram)を, 図7を答案に書き写して空欄を埋めることで完成させよ. カルノー図のマス目には, 出力が1である場合は1を記入し, 0である場合には空欄とし, ドントケアである場合はXを記入すること.

(2-2)  $f(x_4, x_3, x_2, x_1, x_0)$ の最簡積和形(minimum sum of products)を示せ.

(2-3) 小問(2-2)で求めた最簡積和形の論理関数を実現する論理回路を2入力 NAND ゲートのみで実現したい. その回路図を, 下記の図8を答案に書き写して枠内に回路をかくことで完成させよ. ただし, ゲートの総数は9以下とすること.

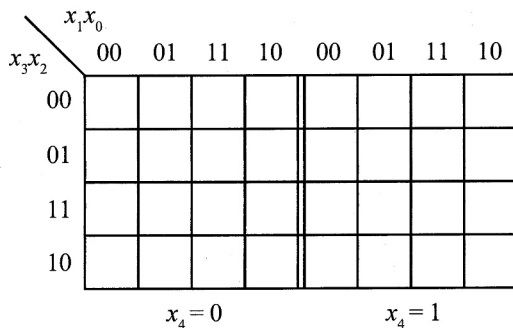


図7

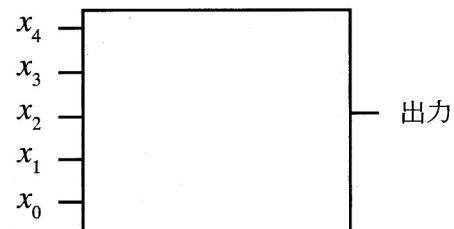


図8

配点: (1-1) 25, (1-2) 20, (1-3) 20, (2-1) 10, (2-2) 10, (2-3) 20, (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) 図1の直交座標系 (Cartesian coordinate system) における点  $(x, y)$  に複素数 (complex number)  $z = x + iy$  を対応させる複素平面 (complex plane) 上において、複素関数 (complex function)  $f(z) = z^2$  を積分路 (path of integration)  $C_1$  または  $C_2$  に沿って積分 (integration) することを考える。なお  $C_1$  と  $C_2$  は直交座標系において以下の通り定義される。

$C_1$ : 点  $P(1,2)$  と  $O(0,0)$  を接続する  $0 \leq x \leq 1$  の区間で定義される線分。ただし始点を  $P$ , 終点を  $O$  とする。

$C_2$ : 点  $O(0,0)$  と  $P(1,2)$  を接続する  $0 \leq x \leq 1$  の区間で定義される四次曲線 (quartic curve)  $y = 2x^4$ 。ただし始点を  $O$ , 終点を  $P$  とする。

以下の各小問に答えよ。

- (1-1)  $f(z)$  の  $C_1$  と  $C_2$  に沿った積分である  $I_1 = \int_{C_1} f(z)dz$  と  $I_2 = \int_{C_2} f(z)dz$  をそれぞれ計算することで、それらの結果に  $I_1 = -I_2$  の関係があることを示せ。

- (1-2) 小問(1-1)の  $I_2$  の値を  $C_2$  に沿った積分を計算せずに求めることを考える。コーシーの積分定理 (Cauchy's integral theorem) と  $I_1$  の計算結果を用いて、 $I_2$  を求めよ。ただしこの計算にコーシーの積分定理を適用できることも示せ。

- (1-3) 複素関数  $g(z) = \frac{1}{z^4+1}$  を  $C_1$  と  $C_2$  からなる積分路で周回積分 (contour integration) せよ。ただしこの積分にコーシーの積分定理を用いることができるか判断し、その理由も説明せよ。

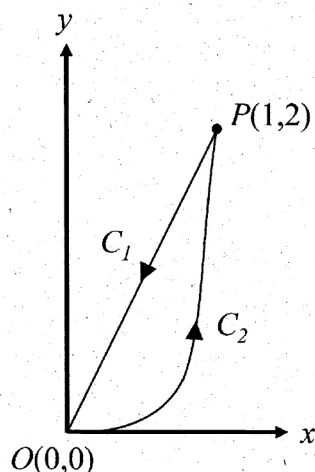


図1: 積分路.  $x$  軸と  $y$  軸はそれぞれ実軸 (real axis) と虚軸 (imaginary axis) である。

(次ページへ続く)

(2) 離散時間信号 (discrete-time signal)  $x[n]$  を送信すると, 観測信号 (observation signal)  $y[n]$  が得られる伝送路 (channel)  $S$  が与えられた. 伝送路  $S$  は, 式 [1] に示す差分方程式 (difference equation) で表される線形時不変システム (linear time-invariant system) としてモデル化される. 図 2 に伝送路  $S$  を表すブロック図 (block diagram) を示す.

$$y[n] = x[n] + \frac{1}{2}x[n-1] + \frac{1}{4}x[n-2] \quad [1]$$

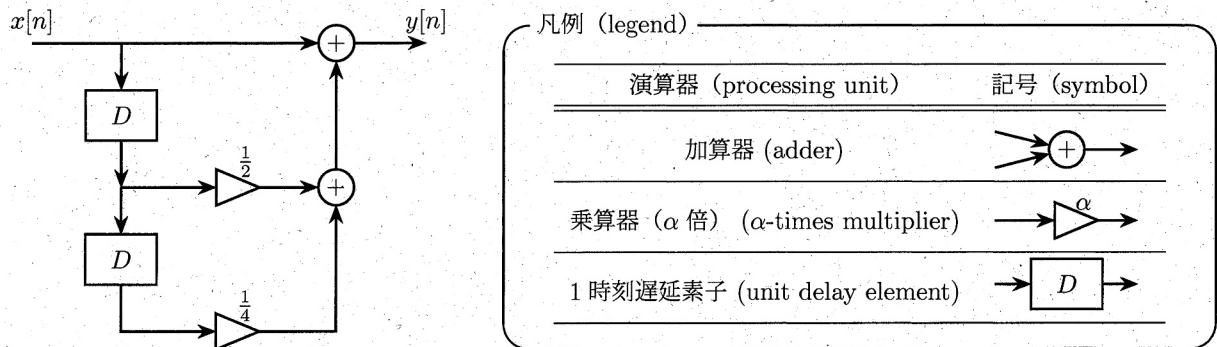


図 2: 伝送路  $S$  を表すブロック図

以下の各小問に答えよ.

- (2-1) 式 [1] に適切な変形と  $z$  変換 ( $z$ -transform) を施すことで, 伝送路  $S$  の伝達関数 (transfer function)  $H_S(z)$  を求めよ.
- (2-2) 伝送路  $S$  における観測信号から, 原信号 (original signal) を厳密に復元するシステム (system)  $T$  を設計したい. システム  $T$  の入力を  $x_T[n]$ , 出力 (復元信号) を  $y_T[n]$  としたとき, システム  $T$  の伝達関数  $H_T(z)$  および差分方程式を書け.
- (2-3) システム  $T$  の極 (pole) と零点 (zero) を求め, システム  $T$  が BIBO 安定 (Bounded-Input Bounded-Output stable) であることを示せ.
- (2-4) 以下は, システム  $T$  の特性 (characteristic) に関する説明 (description) である. 空欄 (blank) に入る適切な語あるいは数式を, 用語集 (terms) から選んで書け. ただし,  $\omega$  は角周波数 (angular frequency) を表す.

説明 (description)

極零点配置 (pole-zero assignment) より, システム  $T$  の振幅 (amplitude) 特性は  $\omega = \boxed{A}$  をピークに持つ. 伝送路  $S$  の特性と対比すると, システム  $T$  は  $\boxed{B}$  フィルタ (filter) として機能する. システム  $T$  は  $\boxed{C}$  インパルス応答 (impulse response) 特性をもち, 線形位相 (linear phase) 特性を  $\boxed{D}$ .

用語集 (terms)

0     $\pm 1$      $\pm \frac{\pi}{4}$      $\pm \frac{\pi}{3}$      $\pm \frac{\pi}{2}$      $\pm \frac{2}{3}\pi$      $\pm \frac{3}{4}\pi$      $\pm \pi$

低域通過 (low-pass)    高域通過 (high-pass)    有限 (finite)    無限 (infinite)    もつ    もたない