令和３年７月31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４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) 25，(2) 25，(3) 25，(4) 25，(5) 25

図 1 に示す ANSI-C 準拠である C 言語のプログラム (program) は，識別子 (id) と得点 (score) を対 (pair) とするデータ (data) を一つ以上読み込んで，得点をキーとして降順 (descending order) に整列 (sort) して出力 (output) するプログラムである．入力 (input) するデータはファイル data.txt から読み込まれる．data.txt の 1 行目には整列するデータの個数 n ($\geqq$ 1)，2 行目から $n+1$ 行目には整列するデータが各行 (each line) に記述されている．以下の各問に答えよ．

(1) 関数 (function) `sort` で実現されている整列アルゴリズム (sorting algorithm) は一般的に何と呼ばれているか名称を答えよ．

(2) 図 2 の data.txt からデータを読み込んでプログラムを実行した場合に，関数 `swap` が 3 回目に実行された直後の `D[0].score`，`D[1].score`，`D[2].score`，`D[3].score`，`D[4].score` の値をそれぞれ求めよ．

(3) キーの値が等しいデータに対して，整列前のデータの並び順の前後関係が整列後も維持される整列アルゴリズムを，安定な (stable) 整列アルゴリズムという．関数 `sort` で実現されている整列アルゴリズムは安定な整列アルゴリズムではない．関数 `sort` が安定な整列アルゴリズムではないことが分かる出力を得たい場合，図 2 の data.txt の 2 行目の得点を 60 からいくつに書き換えれば所望の出力が得られるか数値を答えよ．

(4) 図 2 における 2 行目～6 行目の順番を入れ替えた data.txt からデータを読み込んでプログラムを実行することを考える．関数 `swap` が呼び出される回数が最大となるような data.txt を示せ．また，その際に関数 `swap` が呼び出される回数を示せ．

(5) 関数 `sort` で実現されている整列アルゴリズムの最悪時間計算量 (worst case time complexity) を，整列するデータの個数 n を用いてオーダ表記 (order notation) で理由と共に示せ．

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef struct{
5    int id;
6    int score;
7  } user_t;
8
9  void swap(user_t *A, user_t *B){
10   user_t tmp = *A;
11   *A = *B;
12   *B = tmp;
13 }
14
15 void sort(user_t D[], int left, int right){
16   if(left < right){
17     int j, i = left;
18     user_t pivot = D[right];
19     for(j = left; j < right; j++){
20       if(D[j].score > pivot.score){
21         swap(&D[i], &D[j]);
22         i++;
23       }
24     }
25     swap(&D[i], &D[right]);
26     sort(D, left, i - 1);
27     sort(D, i + 1, right);
28   }
29 }
30
31 int main(){
32   int i, N;
33   user_t *D;
34   FILE *fp = fopen("data.txt", "r");
35   fscanf(fp, "%d", &N);
36   D = (user_t*) malloc(sizeof(user_t) * N);
37   for(i = 0; i < N; i++){
38     fscanf(fp, "%d %d", &D[i].id, &D[i].score);
39   }
40   fclose(fp);
41   sort(D, 0, N - 1);
42   for(i = 0; i < N; i++){
43     printf("%d %d\n", D[i].id, D[i].score);
44   }
45   free(D);
46   return 0;
47 }
```

図 1 プログラム

```
1  5
2  1 60
3  2 90
4  3 50
5  4 70
6  5 100
```

図 2 data.txt

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) 15, (2-1) 10, (2-2) 30, (3-1) 35, (3-2) 35

単一プロセッサ（single processor）上で動作する並行プロセス（concurrent process）について，以下の各問に答えよ．複数のプロセスからアクセスできる共有変数（shared variable）が存在し，あるプロセスが共有変数の値を更新した場合，他のプロセスは直ちに更新後の値を利用できるものとする．以降のプログラムの記述言語はC言語である．

セマフォ（semaphore）を用いたプロセスの並行制御（concurrency control）を実現するために，以下のようにP操作 `P(s)` およびV操作 `V(s)` を定義する．引数 `s` はセマフォ変数を表す．セマフォ変数は整数値（integer）をとり，0以上の初期値（initial value）を持つ．P操作およびV操作はそれぞれ不可分操作（atomic operations）である．

- `P(s)`：`s` から1を減算する．その結果 $s < 0$ となる場合，`P(s)` を呼び出したプロセスを待ち状態（waiting）にする．
- `V(s)`：`s` に1を加算する．その結果 $s \leq 0$ となる場合，`P(s)` の呼び出しによって待ち状態となったプロセスを一つ選び，実行可能状態（ready）にする．

(1) $s < 0$ である場合の $|s|$ の意味として，最も適切な説明を選択肢（ア）～（エ）から一つ選べ．

(ア) 利用可能な共有資源（available shared resources）の数　(イ) 実行可能状態にあるプロセス数
(ウ) 占有されている共有資源（occupied shared resources）の数　(エ) 待ち状態にあるプロセス数

(2) 次の関数（function）`f1` および `f2` について，以下の各小問に答えよ．セマフォ変数 `s` は1に初期化されており，変数 `n` は整数型の共有変数であり0に初期化されている．変数 `i` は共有変数ではない．関数 `f1` を実行するプロセスを二つ並行に実行した後の `n` の値を n_1 とする．また，`n` を再び0に初期化し，関数 `f2` を実行するプロセスを二つ並行に実行した後の `n` の値を n_2 とする．いずれの場合も，これら二つのプロセス以外に，`n` を共有するプロセスはない．

```
1 void f1() {
2   int i;
3   for (i=0; i < 10; i++) {
4     P(s);
5     n = 2*n+3;
6     V(s);
7   }
8 }
```

```
1 void f2() {
2   int i;
3   for (i=0; i < 10; i++) {
4     n = 2*n+3;
5   }
6 }
```

(2-1) 関数 `f1` の5行目のように，排他制御（mutual exclusion）の対象となる部分を表す最も適切な語を，選択肢（オ）～（ク）から一つ選べ．

(オ) クリティカルセクション（critical section）　(カ) ミューテックス（mutex）
(キ) デッドロック（deadlock）　(ク) バリア同期（barrier synchronization）

(2-2) n_1 と n_2 の関係を不等式を用いて示せ．さらにそうなる理由を説明せよ．

(3)　二つのセマフォ変数 s1 および s2 を用いた共有変数の読み書き制御について，以下の各小問に答えよ．以下のプログラムにおいて，変数 n は整数型の共有変数であり 0 に初期化されている．変数 i は共有変数ではない．

(3-1)　まず，読み出しプロセスおよび書き込みプロセスがそれぞれ一つの場合を考える．書き込みプロセスは関数 g1 を実行し，読み出しプロセスは関数 g2 を実行する．これら二つのプロセスを並行に実行して，0 から 9 までの全ての整数を必ず昇順（ascending order）で出力したい．空欄 (a) ～ (c) のそれぞれを一つの P 操作あるいは V 操作で埋め，関数を完成させよ．また，s1 および s2 の初期値を示せ．ただし，これら二つのプロセス以外に，n を共有するプロセスはない．

```
1  void g1() {
2    int i;
3    for (i=0; i < 10; i++) {
4      P(s1);
5      n = n+1;
6      (a) ;
7    }
8  }
```

```
1  void g2() {
2    int i;
3    for (i=0; i < 10; i++) {
4      (b) ;
5      printf("%d\n",n);
6      (c) ;
7    }
8  }
```

(3-2)　次に，読み出しプロセスが一つ，書き込みプロセスが二つの場合を考える．各書き込みプロセスは関数 g3 を実行し，読み出しプロセスは関数 g2 を実行する．これら三つのプロセスを並行に実行して，必ず図 A と一致する結果を出力したい．空欄 (X) を一つの式で，空欄 (d) および (e) のそれぞれを一つの P 操作あるいは V 操作で埋め，関数を完成させよ．また，s1 および s2 の初期値を示せ．ただし，これら三つのプロセス以外に，n を共有するプロセスはない．

```
1  void g3() {
2    int i;
3    for (i=0; i < 10; i++) {
4      P(s1);
5      n = n+1;
6      if ( (X) ) {
7        (d) ;
8      } else {
9        (e) ;
10     }
11   }
12 }
```

```
2
4
6
8
10
12
14
16
18
20
```

図 A: 出力結果

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

正の整数全体の集合 $\mathbb{N}$ における二項関係（binary relation）R を

$$R = \{(k, \ell) \mid k \text{ と } \ell \text{ は相異なり，かつ } 1 \text{ より大きい公約数（common divisor）をもつ}\}$$

により定める．任意の $n \in \mathbb{N}$ に対して，以下の式により定まる頂点集合（vertex set）V_n と辺集合（edge set）E_n をもつ無向グラフ（undirected graph）を G_n とする．

$$V_n = \{k \in \mathbb{N} \mid 1 \leq k \leq n\},$$
$$E_n = \{\{k, \ell\} \mid (k, \ell) \in R \text{ あるいは } (\ell, k) \in R\}.$$

E_n の要素数を $m(n)$ と書く．以下の各問に答えよ．

(1) 二項関係 R に関する以下の各小問に答えよ．

(1-1) R が対称的（symmetric）であるかを判定せよ．その理由を述べよ．

(1-2) R が推移的（transitive）であるかを判定せよ．その理由を述べよ．

(2) 無向グラフ G_n に関する以下の各小問に答えよ．

(2-1) G_6 が持つ辺の数 $m(6)$ と連結成分（connected component）の数をそれぞれ答えよ．

(2-2) 無向グラフ G の部分グラフ（subgraph）H が完全グラフ（complete graph）であるとき，H を G の完全部分グラフ（complete subgraph）と呼ぶ．G_{10} の完全部分グラフの中で頂点数が最も大きいものの頂点集合を求めよ．

(2-3) 不等式

$$m(n) \geq \frac{1}{2}\left(\frac{n}{2} - 1\right)\left(\frac{n}{2} - 2\right)$$

が 1 より大きい任意の $n \in \mathbb{N}$ に対して成り立つことを証明せよ．

(3) 任意の $n \in \mathbb{N}$ に対して

$$\mu(n) = \frac{1}{n}\sum_{k=1}^{n} m(k)$$

とする．極限

$$\gamma = \lim_{n \to \infty} \frac{\mu(n)}{n^2}$$

が存在すると仮定する．以下の各小問に答えよ．

(3-1) 不等式 $\gamma \geq 1/24$ が成り立つことを示せ．

(3-2) 不等式 $\gamma \leq 1/6$ が成り立つことを示せ．

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

文脈自由文法 (context-free grammar) G を (V, T, P, S) で表す．各要素は以下の通りである．

- V: 変数 (variable) の有限集合 (finite set).
- T: 終端記号 (terminal symbol) の有限集合.
- P: 生成規則 (production rule) の有限集合.
- $S \in V$: 開始記号 (start symbol).

空スタックによる受理 (acceptance by empty stack) を行うプッシュダウンオートマトン (pushdown automaton) PDA を $(Q, \Sigma, \Gamma, \delta, q, Z)$ で表す．各要素は以下の通りである．

- Q: 状態 (state) の有限集合.
- Σ: 入力記号 (input symbol) の有限集合.
- Γ: スタック記号 (stack symbol) の有限集合.
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma^*)$: 遷移関数 (transition function). (ε は空文字列 (empty string) を，$\mathcal{P}(Q \times \Gamma^*)$ は $Q \times \Gamma^*$ のべき集合 (power set) を表す．)
- $q \in Q$: 初期状態 (initial state).
- $Z \in \Gamma$: 開始記号.

記号 a, 文字列 (string) w に対し，w に a が現れる回数を $n(a, w)$ とする．言語 (language) L_1 を以下のように定義する．

$$L_1 = \left\{ w \in \{0,1\}^* \;\middle|\; |n(0,w) - n(1,w)| = 1 \right\}$$

(1) 文脈自由文法 $G_1 = (\{S, A\}, \{0,1\}, P_1, S)$ は，L_1 を生成する．P_1 は以下の通りである．

$$S \to A0A, \quad S \to A1A, \quad A \to 0A1A, \quad A \to 1A0A, \quad A \to \varepsilon$$

G_1 があいまい (ambiguous) であることを示せ.

(2) L_1 を生成し，かつ，ε-規則 (ε-production) を持たないように，文脈自由文法 $G_2 = (\{S, A\}, \{0,1\}, P_2, S)$ を定めたい．P_2 を示せ.

(3) L_1 を認識するように，空スタックによる受理を行うプッシュダウンオートマトン $PDA_1 = (\{p, q\}, \{0,1\}, \{X, Y, Z\}, \delta_1, q, Z)$ を定めたい.

$\delta_1(q,0,X) = \{(q,XX)\}$, $\delta_1(q,1,Y) = \{(q,YY)\}$, $\delta_1(q,\varepsilon,X) = \delta_1(q,\varepsilon,Y) = \{(p,\varepsilon)\}$, $\delta_1(q,\varepsilon,Z) = \emptyset$, $\delta_1(p,0,X) = \delta_1(p,0,Y) = \delta_1(p,0,Z) = \delta_1(p,1,X) = \delta_1(p,1,Y) = \delta_1(p,1,Z) = \delta_1(p,\varepsilon,X) = \delta_1(p,\varepsilon,Y) = \emptyset$, $\delta_1(p,\varepsilon,Z) = \{(p,\varepsilon)\}$ とする．$\emptyset$ は空集合 (empty set) を表す.

$\delta_1(q,0,Y)$, $\delta_1(q,0,Z)$, $\delta_1(q,1,X)$, $\delta_1(q,1,Z)$ をそれぞれ示せ.

(次ページに続く)

(4) L_2 を，以下の条件をすべて満たす任意の正則言語 (regular language) とする．

- $L_2 \subseteq L_1$.
- L_2 の要素数は無限である．
- 任意の $w \in L_2$ について，$ww'w \in L_2$ となる $w' \in L_2$ が存在する．

空スタックによる受理を行う決定性プッシュダウンオートマトン (deterministic pushdown automaton) によって，L_2 を認識することができるかどうか，理由と共に答えよ．

なお，プッシュダウンオートマトン $PDA = (Q, \Sigma, \Gamma, \delta, q, Z)$ が決定性プッシュダウンオートマトンである必要十分条件は，以下の通りである．

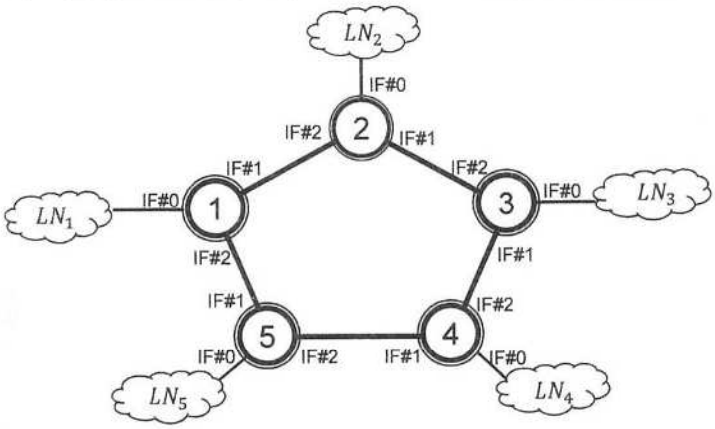
- 任意の $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $X \in \Gamma$ について $|\delta(q, a, X)| \leq 1$，かつ
- 任意の $q \in Q$, $X \in \Gamma$ について，$\delta(q, a, X) \neq \emptyset$ となる $a \in \Sigma$ があれば，$\delta(q, \varepsilon, X) = \emptyset$.

(5) 問 (4) における L_2 の例を，正則表現 (regular expression) を用いて一つ示せ．

5 【選択問題】ネットワーク

配点: (1-1) 25, (1-2) 25, (1-3) 35, (2-1) 10, (2-2) 10, (2-3) 20

(1) $2n+1$ 個のノード（node）がリング（ring）状に接続されるネットワークにおける経路制御（routing）を考える．n は正の整数であり，以下の図は $n=2$ とした 5 ノードのネットワークである．各ノード i $(1 \leq i \leq 2n+1)$ は，三つのインターフェース（interface）IF#0〜IF#2 を持ち，IF#0 は自身のローカルネットワーク（local network）LN_i に接続されている．IF#1 は時計回り（clockwise）方向，IF#2 は反時計回り（counter-clockwise）方向のノードに接続されている．各リンク（link）のコスト（cost）は 1 とする．以下の各小問に答えよ．



(1-1) 距離ベクトル（distance vector）型プロトコル（protocol）により最小コスト（minimum cost）経路を求めることを考える．時刻ステップ（time step）t が 1 経過すると，隣接ノード間の経路情報交換（routing information exchange）が完了するものとする．

$n=2$ である時，時刻ステップ $t=1,2$ それぞれにおいて，ノード 1 が保持する経路表（routing table）を示せ．時刻ステップ t が 0 である時の各ノード i $(1 \leq i \leq 5)$ の経路表のエントリー（entry）の数は 1 であるとし，経路表は以下の表で与えられるものとする．

宛先 (destination)	インターフェース (interface)	到達コスト (cost to reach)
LN_i	IF#0	1

(1-2) 各ノードの経路表のエントリーの最大数を 2 としつつ，すべてのローカルネットワーク間でパケット（packet）送受信を可能とする経路表を考える．ノード 1 が保持すべき経路表を一つ提示せよ．また，あるパケットが経由するリンク数を経路長（path length）とし，その最大値である最大経路長（maximum path length）を n を用いて表せ．

(1-3) 各ノードの経路表のエントリーの最大数を k $(k \geq 2)$ としつつ，すべてのローカルネットワーク間でパケット送受信を可能とし，さらに，最大経路長を最小とする経路表を考える．最大経路長を n および k を用いて表せ．導出過程も示すこと．

（次ページに続く）

(2) 情報源アルファベット（source alphabet）$A = \{a, b, c, d, e, f\}$ を符号化（coding）するための符号（code）C_1，C_2，C_3，C_4，C_5 が以下の表で与えられている．以下の各小問に答えよ．

A	C_1	C_2	C_3	C_4	C_5
a	111	000	01	100	11
b	011	001	1	10	01
c	01	010	11	000	10
d	00	011	101	01	001
e	010	100	00	110	000
f	110	101	100	001	011

(2-1) 符号 C_1，C_2，C_3，C_4，C_5 のうち，瞬時に復号可能（instantaneously decodable）なものをすべて答えよ．瞬時に復号可能な符号は，語頭符号（prefix code）とも呼ばれる．

(2-2) 符号 C_1，C_2，C_3，C_4，C_5 のうち，一意に復号可能（uniquely decodable）なものをすべて答えよ．

(2-3) 以下の文章の空欄 [(a)]，[(b)]，[(c)] に当てはまる最も適切な数式または文を，下記の選択肢 (ア) ～ (ケ) の中から一つ選び，その記号を答えよ．

情報源アルファベット $\mathcal{X}$ に対する 2 元符号（binary code）C において，各 $x \in \mathcal{X}$ に対する符号語（codeword）の長さを $\ell(x)$ と表す．このとき，[(a)] はクラフトの不等式（Kraft's inequality）と呼ばれており，符号 C は，[(b)]．また，$\mathcal{X}$ 上の記憶のない情報源（memoryless information source）X のエントロピー（entropy）$H(X)$ ならびに X に対する符号 C の平均符号語長（average codeword length）$L_C(X)$ を，それぞれ

$$H(X) = -\sum_{x \in \mathcal{X}} P_X(x) \log_2 P_X(x), \quad L_C(X) = \sum_{x \in \mathcal{X}} P_X(x) \ell(x)$$

と定める．ただし，$P_X(x)$ は情報源 X が値 $x \in \mathcal{X}$ をとる確率（probability）である．このとき，一意に復号可能な符号 C の平均符号語長に関し，不等式 [(c)] が必ず成り立つ．

選択肢

(ア) $\sum_{x \in \mathcal{X}} 1/\ell(x) > 1$　　(イ) $\sum_{x \in \mathcal{X}} 1/\log_2 \ell(x) \leq 1$　　(ウ) $\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1$

(エ) クラフトの不等式を満たすとき，かつそのときに限り，瞬時に復号可能である

(オ) クラフトの不等式を満たすが，瞬時に復号可能でない場合がある

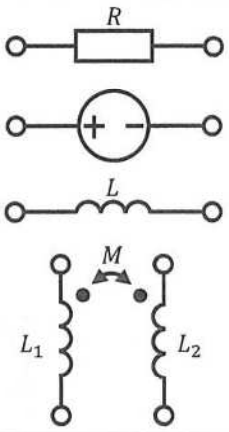
(カ) 瞬時に復号可能であるが，クラフトの不等式を満たさない場合がある

(キ) $L_C(X) \geq H(X)$　　(ク) $L_C(X) < H(X) + 1$　　(ケ) $L_C(X) \leq H(X)$

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

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



抵抗値 (resistance) R を持つ抵抗

電圧源 (voltage source)

インダクタンス (inductance) L を持つインダクタ

相互インダクタンス (mutual inductance) M を持ち，1次側，2次側のインダクタンスがそれぞれ L_1, L_2 の回路 (circuit)

(1) 図 1 に示す回路を図 2 に示す等価回路 (equivalent circuit) に置き換えることができる．インダクタンス L_a, L_b, L_c を L_1, L_2, M を用いて表せ．なお，図 1 に示す回路の電圧 v_1, v_2 および電流 i_1, i_2 に対し，時刻 t を用いて式 (A) が成立する．

$$\begin{cases} v_1 = L_1 \dfrac{\mathrm{d}i_1}{\mathrm{d}t} + M \dfrac{\mathrm{d}i_2}{\mathrm{d}t} \\ v_2 = M \dfrac{\mathrm{d}i_1}{\mathrm{d}t} + L_2 \dfrac{\mathrm{d}i_2}{\mathrm{d}t} \end{cases} \quad \cdots \quad \text{(A)}$$

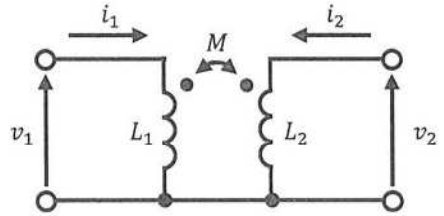


図 1

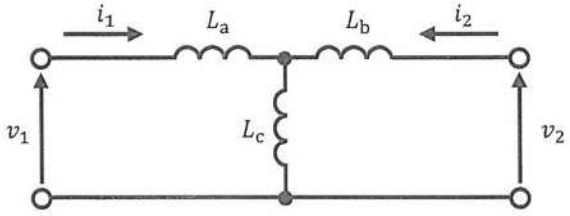


図 2

(2) 図 3 に示す回路に対し，時刻 $t < 0$ において回路が定常状態 (steady state) にあった．図 3 に示す回路に対して，式 (B) に示すステップ電圧 (voltage step) を与える．以下の各小問に答えよ．

$$V_{\text{in}} = \begin{cases} 0 & (t < 0) \\ V_0 & (t \geq 0) \end{cases} \qquad \cdots \quad \text{(B)}$$

(2-1) 時刻 $t < 0$ において，電流 i_1 を求めよ．

(2-2) 時刻 $t = 0$ から十分に時間が経過した後，回路が定常状態になった．このとき，電流 i_1 を求めよ．

(2-3) 回路中の各インダクタンス L_1，L_2 と相互インダクタンス M は，$L_1 = L$，$L_2 = 3L$，$M = L$ の各値を取るものとする（L は正の定数）．時刻 $t \geq 0$ において電流 i_1 を時刻 t の関数として求めよ．L_1，L_2，M の表記を用いずに，L を用いて解答せよ．

(3) 図 3 に示す回路の各インダクタンス L_1，L_2 と相互インダクタンス M は，$L_1 = L$，$L_2 = 3L$，$M = L$ の各値を取るものとする（L は正の定数）．図 3 の回路に対し，入力電圧として交流電圧 (alternating current voltage) $V_{\text{in}} = V_0 \sin \omega t$ を印加した．ある R_1 を設定したところ，十分時間が経過した後，a–b 間の電位差がゼロに収束した．以下の各小問に答えよ．L_1，L_2，M の表記を用いずに，L を用いて解答せよ．

(3-1) R_1 を求めよ．

(3-2) 電圧源に流れる電流 I_{in}，および電圧源の電圧 V_{in} の位相 (phase) の差が 45° になる角周波数 (angular frequency) ω_0を求めよ．$\omega_0 > 0$ の範囲で求めよ．

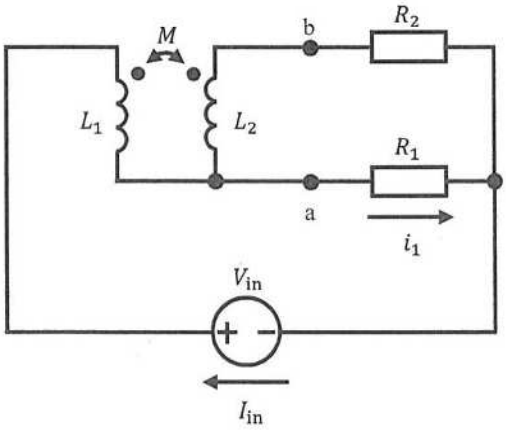


図 3

7 【選択問題】数学解析と信号処理 （情報工学 12/12）

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

$\mathcal{H}(s)$ は，ラプラス変換（Laplace transform）により得られる，あるアナログフィルタ（analog filter）の伝達関数（transfer function）である．ただし，N は 1 以上の整数，s は複素変数（complex variable），j は虚数単位（imaginary unit）を表す．

$$\mathcal{H}(s) = \frac{1}{\prod_{k=1}^{N}(s - s_k)}, \qquad s_k = \exp\left(j\frac{2k+N-1}{2N}\pi\right) \tag{1}$$

以下の各問に答えよ．

(1) $\mathcal{H}(s)$ を部分分数分解（partial fraction decomposition）すると以下のように書ける．

$$\mathcal{H}(s) = \frac{1}{\prod_{k=1}^{N}(s - s_k)} = \sum_{k=1}^{N}\frac{A_k}{s - s_k} \tag{2}$$

ここで，以下のように定義される関数 $f(t)$（ただし t は実数（real number））について考える．

$$f(t) = \begin{cases} \sum_{k=1}^{N} f_k(t) & (t \geq 0) \\ 0 & (otherwise) \end{cases}, \qquad f_k(t) = A_k e^{s_k t} \tag{3}$$

$f(t)$ のラプラス変換 $\mathcal{L}[f(t)](s)$ が存在する場合に $\mathcal{L}[f(t)](s) = \mathcal{H}(s)$ となることを，計算過程とともに示せ（このときの A_k の値を答える必要はない）．また，$\mathcal{L}[f(t)](s)$ の収束域（region of convergence）を s を用いて説明せよ．

(2) 離散時間（discrete-time）における線形時不変システム（linear time-invariant system）の入力を $x[n]$，出力を $y[n]$ とする（ただし n は整数（integer））．ここで，式 [3] を離散化した関数 $h[n]$ をインパルス応答（impulse response）とするディジタルフィルタ（digital filter）を設計することを考える．以下の各小問に答えよ．

$$h[n] = \begin{cases} \sum_{k=1}^{N} h_k[n] & (n \geq 0) \\ 0 & (otherwise) \end{cases}, \qquad h_k[n] = A_k e^{s_k n} \tag{4}$$

(2-1) $n \geq 0$ としたとき，畳み込み演算子（convolution operator）$*$ を使うと，入出力信号とインパルス応答の関係は以下のように表せる．

$$y[n] = \sum_{k=1}^{N} y_k[n], \qquad y_k[n] = h_k[n] * x[n] \tag{5}$$

式 [5] を変形すると以下のような差分方程式（difference equation）を導くことができることを示せ．

$$y[n] = \sum_{k=1}^{N}\left(e^{s_k} y_k[n-1] + A_k x[n]\right) \tag{6}$$

(2-2) 式 [6] に，適切な変形と z 変換（z-transform）を施すことで，ディジタルフィルタの伝達関数 $H(z)$ を求めよ．

(2-3) 小問 (2-1) および (2-2) は，$\mathcal{H}(s)$ を伝達関数とするアナログフィルタをプロトタイプ（prototype）として無限インパルス応答（infinite impulse response, IIR）ディジタルフィルタを設計する流れの一部である．有限インパルス応答（finite impulse response, FIR）フィルタを用いても類似した特性を持つディジタルフィルタを設計することは可能であるが，IIR フィルタを用いる利点は何か．差分方程式の形と関連させて述べよ．