

平成 2 9 年度
筑波大学大学院博士課程
システム情報工学研究科
コンピュータサイエンス専攻
博士前期課程（一般入学試験 2 月期）
試験問題 基礎科目（数学，情報基礎）

Mathematics／Fundamentals of Computer Science (CS)

[注意事項] [Instructions]

1. 試験開始の合図があるまで，問題の中を見てはいけません．また，筆記用具を手に持っていないけません．

Do NOT open this booklet before the examination starts. In addition, do not have a pen in hand before the examination starts.

2. 試験開始の合図のあとで，全ての解答用紙の定められた欄に，研究科，専攻，受験番号を記入すること．

Fill in the designated spaces on each answer sheet with the name of the graduate school, name of your main field, and the examination number after the examination starts.

3. この問題は全部で 22 ページ（表紙を除く）です．1～11 ページは日本語版，12～22 ページは英語版です．

This booklet consists of 22 pages, excluding the cover sheet. The Japanese version is shown on pages 1–11 and the English version on pages 12–22.

4. 解答用紙（罫線有り）を 4 枚，下書き用紙（白紙）を 1 枚配布します．

You are given four answer sheets (ruled paper) and one draft sheet (white paper).

5. 問題は全部で 4 問（**数学**1，**数学**2，**情報**1，**情報**2）あります．このうち，以下の組み合わせで 2 問選択すること．**数学**1と**情報**1を同時に選択しないこと．各問題には小問が 2 問（**(1)**と**(2)**）あります．小問ごとに解答用紙を分けて記入すること．

There are four problems (**Math**1, **Math**2, **CS**1, and **CS**2). Select two problems from the following combinations. Do NOT select **Math**1 and **CS**1 together. Each problem has two sub-problems **(1)** and **(2)**. Write your answer to each sub-problem in a different answer sheet.

	解答用紙 1 枚目と 2 枚目 / First and second answer sheets	解答用紙 3 枚目と 4 枚目 / Third and fourth answer sheets
1	数学 ／ Math 1 (1) (2)	数学 ／ Math 2 (1) (2)
2	数学 ／ Math 1 (1) (2)	情報 ／ CS 2 (1) (2)
3	数学 ／ Math 2 (1) (2)	情報 ／ CS 1 (1) (2)
4	数学 ／ Math 2 (1) (2)	情報 ／ CS 2 (1) (2)
5	情報 ／ CS 1 (1) (2)	情報 ／ CS 2 (1) (2)

6. 解答用紙に解答を記述する際に，問題番号を必ず明記すること．

When writing the answers, clearly label the problem number on each answer sheet.

平成 2 9 年 2 月 2 日

問題 数学 1 (1) は, 数学 1 (2) とともに解答すること. 数学 1 を選択した場合は, 情報 1 を選択しないこと. この解答用紙の先頭に, 「数学 1 (1)」と明記し, それ以外の問題を解答しないこと.

問題 数学 1 (1)

\mathbb{Z} を整数の集合とする. 次の 2 つの条件を両方とも満たす関数 $f: \mathbb{Z} \rightarrow \mathbb{Z}$ が存在すると仮定する.

- すべての $x > 100$ に対して, $f(x) = x - 10$ である.
- すべての $x \leq 100$ に対して, $f(x) = f(f(x + 11))$ である.

この時, 次の問に答えなさい.

- (a) $f(100) = 91$ であることを示しなさい.
- (b) $90 \leq x < 100$ となるすべての整数 x に対して, $f(x) = f(x + 1)$ であることを示しなさい.
- (c) $f(90)$ の値を求めなさい. 計算過程を示すこと.
- (d) $f(0)$ の値を求めなさい. 計算過程を示すこと.
- (e) すべての非負の整数 n に対して, $f(100 - n) = f(0)$ が成立することを証明しなさい.

問題 数学 1 (2) は、数学 1 (1) とともに解答すること。数学 1 を選択した場合は、情報 1 を選択しないこと。この解答用紙の先頭に、「数学 1 (2)」と明記し、それ以外の問題を解答しないこと。

問題 数学 1 (2)

集合 $S = \{1, 2, 3, 4\}$ の直積集合 $S \times S$ 上の二項関係 R を

$$R = \{((a, b), (c, d)) \mid \text{ある } n \in \mathbb{Z}_{>0} \text{ に対して } a \equiv nc \pmod{5} \text{ かつ } b \equiv nd \pmod{5}\}$$

と定義する。ここで、 $\mathbb{Z}_{>0}$ は正の整数全体の集合を表す。また、正の整数 x, y, m に対し、 $x \equiv y \pmod{m}$ は、 $x - y$ が m で割り切れることを表す。

(a) $((1, 4), (3, 2)) \in R$ であることを示せ。

関係 R は、反射律・対称律・推移律を満たすため、集合 $S \times S$ 上の同値関係となる。集合 $S \times S$ の要素 (x, y) を含む R の同値類 $[(x, y)]$ とは、

$$[(x, y)] = \{(a, b) \in S \times S \mid ((a, b), (x, y)) \in R\}$$

によって定義される集合のことである。

(b) R の同値類 $[(1, 1)]$ を求めよ。

(c) R の同値類 $[(1, 2)]$ を求めよ。

(d) R の相異なる同値類の個数を求めよ。

集合 $T = \{1, 2, 3\}$ の直積集合 $T \times T$ 上の二項関係 R' を

$$R' = \{((a, b), (c, d)) \mid \text{ある } n \in \mathbb{Z}_{>0} \text{ に対して } a \equiv nc \pmod{4} \text{ かつ } b \equiv nd \pmod{4}\}$$

と定義する。

(e) 関係 R' は集合 $T \times T$ 上の同値関係とならないことを示せ。

問題 数学 2 (1) は, 数学 2 (2) とともに解答すること. この解答用紙の先頭に, 「数学 2 (1)」と明記し, それ以外の問題を解答しないこと.

問題 数学 2 (1)

以下の設問に答えなさい.

(a) 行列 A の逆行列を求めよ.

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

(b) 行列 B の行列式を求めよ.

$$B = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 1 & 2 \\ 1 & 2 & 4 \end{pmatrix}$$

(c) n 次単位行列を E_n とする. E_n の第 i 行に第 j 行を c 倍したものを加えて得られた行列を $P_n(i, j; c)$ とする. E_n の第 i 行を c 倍した行列を $Q_n(i; c)$ とする. ただし $c \neq 0$ である. E_n の第 i 行と第 j 行を入れ替えた行列を $R_n(i, j)$ とする. ただし $i \neq j$ である. このとき, $Q_n(j; -1)R_n(i, j)$ を $P_n(k, l; c)$ の積の形で表せ.

問題 数学 2 (2) は, 数学 2 (1) とともに解答すること. この解答用紙の先頭に, 「数学 2 (2)」と明記し, それ以外の問題を解答しないこと.

問題 数学 2 (2)

直角座標 (x, y) において以下の式で定義される曲線はレムニスケート曲線と呼ばれる.

$$(x^2 + y^2)^2 = (x^2 - y^2) \quad (1)$$

次の問題に答えなさい.

- (a) 式 (1) は, 同じ曲線を定義する極座標形式の式 (2) に変換できることを証明しなさい.

$$r^2 = \cos 2\theta \quad (2)$$

ここで r は原点からの距離, θ は x 軸 (正の側) からの角度である. ただし, 極座標 (r, θ) から直角座標 (x, y) への変換は次のように与えられるものとする.

$$x = r \cos \theta$$

$$y = r \sin \theta$$

- (b) このレムニスケート曲線の概形を 2 次元 x - y 平面上に示しなさい. その際, 曲線上の点の x 座標と y 座標の最大値と最小値を図の中に記入しなさい.
- (c) このレムニスケート曲線で囲まれる領域の面積を計算しなさい. 計算過程を示すこと. 極座標形式の式 $r = f(\theta) \geq 0$ が与えられたとき図 1 に示される領域 S の面積に関する以下の式を使ってよい.

$$S = \frac{1}{2} \int_{\alpha}^{\beta} (f(\theta))^2 d\theta$$

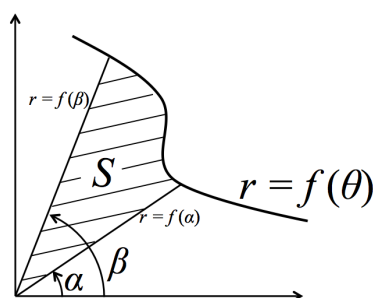


図 1

問題 情報 1 (1) は、情報 1 (2) とともに解答すること。情報 1 を選択した場合は、数学 1 を選択しないこと。この解答用紙の先頭に、「情報 1 (1)」と明記し、それ以外の問題を解答しないこと。

問題 情報 1 (1)

\mathbb{Z} を整数の集合とする。次の 2 つの条件を両方とも満たす関数 $f: \mathbb{Z} \rightarrow \mathbb{Z}$ が存在すると仮定する。

- すべての $x > 100$ に対して、 $f(x) = x - 10$ である。
- すべての $x \leq 100$ に対して、 $f(x) = f(f(x + 11))$ である。

この時、次の問に答えなさい。

- (a) $f(100) = 91$ であることを示しなさい。
- (b) $90 \leq x < 100$ となるすべての整数 x に対して、 $f(x) = f(x + 1)$ であることを示しなさい。
- (c) $f(90)$ の値を求めなさい。計算過程を示すこと。
- (d) $f(0)$ の値を求めなさい。計算過程を示すこと。
- (e) すべての非負の整数 n に対して、 $f(100 - n) = f(0)$ が成立することを証明しなさい。

問題 情報 1 (2) は、情報 1 (1) とともに解答すること。情報 1 を選択した場合は、数学 1 を選択しないこと。この解答用紙の先頭に、「情報 1 (2)」と明記し、それ以外の問題を解答しないこと。

問題 情報 1 (2)

ブール代数 $\{0, 1\}$ 上の 3 変数関数 $f(x, y, z)$ を

$$f(x, y, z) = (\bar{x} \cdot \bar{y} + y) \cdot \bar{z} + \bar{x} \cdot y + y + y \cdot z$$

によって定義する。このとき、以下の問に答えよ。

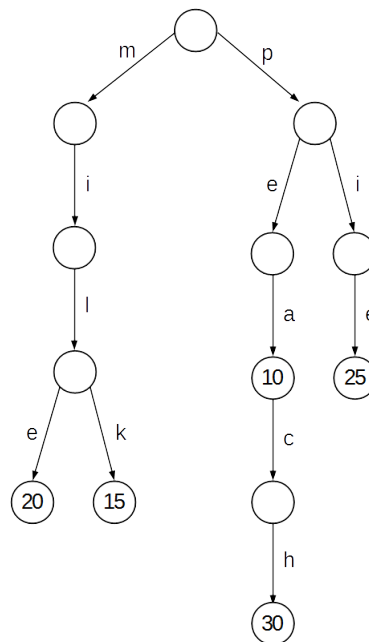
- (a) $f(x, y, z)$ の真理値表を書け。
- (b) $f(x, y, z)$ の乗法標準形を求めよ。
- (c) $f(x, y, z)$ の加法標準形を求めよ。
- (d) 問 (c) で求めた $f(x, y, z)$ の加法標準形をカルノー図を用いて簡単化せよ。
- (e) $f(x, y, z)$ を論理回路で表現せよ。ただし、論理ゲートは、NOT ゲート、AND ゲート、OR ゲートのみを用い、かつ、できるだけ少ない数の論理ゲートを用いること。

問題 情報 **2** (1) は、情報 **2** (2) とともに解答すること。 この解答用紙の先頭に、「情報 **2** (1)」と明記し、それ以外の問題を解答しないこと。

問題 情報 **2** (1)

木構造を用いてキー (key) と値 (value) のペアを格納する連想配列を C 言語で実装することを考える。ただし、キーは a~z の英小文字からなる文字列、値は正の整数とする。

ここで用いる木構造は以下の図のような根付き木である。



上図の例では、以下の表の 5 つのペアを格納した根付き木を表している。

キー	値
mile	20
milk	15
pea	10
peach	30
pie	25

上図の木のノードを表現するために、以下の構造体 `node` を用いる。

```
1 struct node {
2     struct node *c[26];
3     int isEnd;
4     int value;
5 };
```

配列 `c` には、a~z の各文字に対する子ノードへのポインタが格納される。なお、子ノードがない場合には `NULL` が格納される。また、`isEnd` には、そのノードがキーの最後の文字に対するノードである場合には `TRUE` を、そうでない場合には `FALSE` を格納する。`isEnd` に `TRUE` が格納される場合には、`value` にそのキーとペアになる値が格納される。

この構造体を用いて以下の関数を実装することを考える。

- `struct node *new_node()`

新たなノードを作成して初期化し、そのノードへのポインタを返す。

- `void insert(struct node *root, char *key, int value)`

`root` が指すノードを根とする根付き木に対して、キー `key` と値 `value` のペアを追加する。キー `key` が根付き木にすでに存在している場合、古い値を `value` でおきかえる。

- `int search(struct node *root, char *key)`

`root` が指すノードを根とする根付き木に `key` をキーとするペアがあるかを探し、あればそのペアの値を返す。なければ `-1` を返す。

(a) 以下のプログラムは構造体 `node` を用いて関数 `new_node`, `insert`, `search` を実装したものである。

(A) ~ (F) を埋めてこのプログラムを完成させなさい。

```

1  #define TRUE 1
2  #define FALSE 0
3
4  struct node *new_node()
5  {
6      int i;
7      struct node *n = (struct node *) malloc(sizeof(struct node));
8
9      for (i = 0; i < 26; i++) {
10         n->c[i] = NULL;
11     }
12     n->isEnd = FALSE;
13     return n;
14 }
15
16 void insert(struct node *root, char *key, int value)
17 {
18     int l;
19     int len = strlen(key);
20     struct node *n = root;
21
22     for (l = 0; l < len; l++)
23     {
24         int i = key[l] - 'a';
25         if ( (A) ) {
26             n->c[i] = new_node();
27         }
28         n = (B) ;
29     }
30     n->isEnd = (C) ;
31     (D) ;
32 }
33
34 int search(struct node *root, char *key)
35 {
36     int l;
37     int len = strlen(key);
38     struct node *n = root;
39
40     for (l = 0; l < len; l++)
41     {
42         int i = key[l] - 'a';
43
44         if (n->c[i] == NULL)
45             return -1;
46
47         n = (E) ;
48     }
49     if ( (F) ) {

```

```
50         return n->value;
51     } else {
52         return -1;
53     }
54 }
```

- (b) n 個のペアが格納されている `root` が指すノードを根とする根付き木に対して，長さ k の文字列を `key` として関数 `search` を呼び出したときの最悪の場合の漸近的な時間計算量を示し，その理由も示しなさい．

問題 情報 **2** (2) は、情報 **2** (1) とともに解答すること。 この解答用紙の先頭に、「情報 **2** (2)」と明記し、それ以外の問題を解答しないこと。

問題 情報 **2** (2)

次ページに掲載されている C 言語で記述されたプログラムリストについての以下の設問に答えなさい。

- (a) このプログラムは、固定長の配列を用いて、可変長の文字列のバッファを実現する。プログラムリストの 8 行目から 22 行目で定義されている関数は、与えられた文字列をバッファへ格納する。いま、`head` と `tail` が共に 12 にセットされ、配列 `store` の中身がすべて `'\0'` (NULL 文字) で初期化されている `struct buffer` 型の構造体 `buf` があると仮定する。以下に示す順序でこの関数を実行した後の `buf.store` の内容を答えよ。

```
put_str(&buf, "ten");
put_str(&buf, "six");
put_str(&buf, "three");
put_str(&buf, "four");
put_str(&buf, "seven");
put_str(&buf, "two");
put_str(&buf, "eight");
```

- (b) プログラムリストの 24 行目から 35 行目は、バッファに格納された文字列を 1 つ取り出す関数である。この関数は、バッファに何も格納されていない場合は 0 を返す。それ以外の場合は、`head` から始まる文字列を `dest` の指すメモリ領域にコピーし、1 を返す。(A) から (D) の空欄を埋めてプログラムを完成させよ。なお、空欄の中では関数呼び出しは用いないこと。
- (c) このプログラムで実現されているデータ構造は、FIFO (First-In, First-Out) と LIFO (Last-In, First-Out) のどちらであるか？
- (d) このプログラムは、`head` や `tail` の値の桁あふれによって誤動作を起こすかもしれない。それを防ぐために、33 行目と 34 行目の間に以下の 3 行を追加することを考える。(E) の空欄を埋めて桁あふれを防止せよ。なお、空欄の中では関数呼び出しは用いないこと。

```
i = (E);
buf->head -= i;
buf->tail -= i;
```

```

1  #define BUFSIZE 24
2
3  struct buffer {
4      char store[BUFSIZE];
5      int head, tail;
6  };
7
8  int put_str(struct buffer *buf, char *str) {
9      int i = buf->tail;
10
11     while (i - buf->head < BUFSIZE) {
12         buf->store[i++ % BUFSIZE] = *str;
13         if (*str == '\0') {
14             buf->tail = i;
15             return 1;
16         } else {
17             str++;
18         }
19     }
20
21     return 0;
22 }
23
24 int get_str(struct buffer *buf, char *dest) {
25     int i = buf->head;
26
27     if (i == ) return 0;
28
29     do {
30         *dest = ;
31     } while () != '\0');
32
33      = i;
34     return 1;
35 }

```

Answer **Problem Math** 1 (1) with **Math** 1 (2). If you select **Math** 1, then do not select **CS** 1. Clearly label the answer sheet at the top of the page as “**Math** 1 (1)” and do not write answers to other problems on the answer sheet.

Problem Math 1 (1)

Let \mathbb{Z} be the set of integers. Suppose that there exists a function $f : \mathbb{Z} \rightarrow \mathbb{Z}$ which satisfies both of the following two conditions.

- $f(x) = x - 10$ for any $x > 100$, and
- $f(x) = f(f(x + 11))$ for any $x \leq 100$.

Answer the following questions.

- (a) Show that $f(100) = 91$.
- (b) Show that $f(x) = f(x + 1)$ holds for any integer x such that $90 \leq x < 100$.
- (c) Compute the value of $f(90)$. Show your steps.
- (d) Compute the value of $f(0)$. Show your steps.
- (e) Prove that $f(100 - n) = f(0)$ holds for any non-negative integer n .

Answer **Problem Math** 1 (2) with **Math** 1 (1). If you select **Math** 1, then do not select **CS** 1. Clearly label the answer sheet at the top of the page as “**Math** 1 (2)” and do not write answers to other problems on the answer sheet.

Problem Math 1 (2)

Let $S = \{1, 2, 3, 4\}$. Define a binary relation R on the Cartesian product $S \times S$ of the set S by

$$R = \{((a, b), (c, d)) \mid a \equiv nc \pmod{5} \text{ and } b \equiv nd \pmod{5} \text{ for some } n \in \mathbb{Z}_{>0}\},$$

where $\mathbb{Z}_{>0}$ is the set of positive integers and $x \equiv y \pmod{m}$ means that m divides $x - y$ for $x, y, m \in \mathbb{Z}_{>0}$.

- (a) Show that $((1, 4), (3, 2)) \in R$.

Note that the relation R is an equivalence relation on the set $S \times S$ since R satisfies the reflexive law, the symmetric law, and the transitive law. For an element (x, y) of the set $S \times S$, the equivalence class $[(x, y)]$ of R containing the element (x, y) is the set defined by

$$[(x, y)] = \{(a, b) \in S \times S \mid ((a, b), (x, y)) \in R\}.$$

- (b) Find the equivalence class $[(1, 1)]$ of R .
- (c) Find the equivalence class $[(1, 2)]$ of R .
- (d) Find the number of the mutually distinct equivalence classes of R .

Let $T = \{1, 2, 3\}$. Define a binary relation R' on the Cartesian product $T \times T$ of the set T by

$$R' = \{((a, b), (c, d)) \mid a \equiv nc \pmod{4} \text{ and } b \equiv nd \pmod{4} \text{ for some } n \in \mathbb{Z}_{>0}\}.$$

- (e) Show that the relation R' is not an equivalence relation on the set $T \times T$.

Answer **Problem Math** 2 (1) with **Math** 2 (2). Clearly label the answer sheet at the top of the page as “**Math** 2 (1)” and do not write answers to other problems on the answer sheet.

Problem Math 2 (1)

Answer the following questions.

- (a) Find the inverse matrix of A .

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

- (b) Find the determinant of matrix B .

$$B = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 1 & 2 \\ 1 & 2 & 4 \end{pmatrix}$$

- (c) Let E_n be the n -th identity matrix. Let $P_n(i, j; c)$ be the matrix obtained by adding c times j -th row of E_n to the i -th row of E_n . Let $Q_n(i; c)$ be the matrix obtained by c times i -th row of E_n , where $c \neq 0$. Let $R_n(i, j)$ be the matrix obtained by interchanging the i -th row and the j -th row of E_n , where $i \neq j$. In this case, express $Q_n(j; -1)R_n(i, j)$ in the form of the product of $P_n(k, l; c)$.

Answer **Problem Math** 2 (2) with **Math** 2 (1). Clearly label the answer sheet at the top of the page as “**Math** 2 (2)” and do not write answers to other problems on the answer sheet.

Problem Math 2 (2)

The curve defined by the following equation in Cartesian coordinates x and y is known as a lemniscate.

$$(x^2 + y^2)^2 = (x^2 - y^2) \quad (1)$$

Answer the following questions.

- (a) Prove that we can transform Equation (1) into Equation (2) in polar coordinates which defines the same curve

$$r^2 = \cos 2\theta, \quad (2)$$

where r is a distance from the origin and θ is an angle from the x -axis (the positive side). Note that the polar-Cartesian transformation from polar coordinates (r, θ) to Cartesian coordinates (x, y) is defined by the following equations:

$$x = r \cos \theta,$$

$$y = r \sin \theta.$$

- (b) Sketch this lemniscate curve in the two-dimensional x - y plane and write the maximum and minimum values of x and y coordinates of the points on the curve in the sketch.
- (c) Find the area of the region bounded by this lemniscate curve. Show your steps. You can use the following equation regarding the area S in Figure 1 defined by a polar equation $r = f(\theta) \geq 0$.

$$S = \frac{1}{2} \int_{\alpha}^{\beta} (f(\theta))^2 d\theta$$

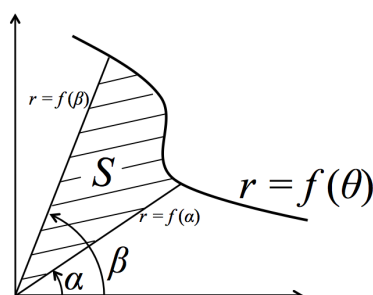


Figure 1

Answer **Problem CS** 1 (1) with **CS** 1 (2). If you select **CS** 1, then do not select **Math** 1. Clearly label the answer sheet at the top of the page as “**CS** 1 (1)” and do not write answers to other problems on the answer sheet.

Problem CS 1 (1)

Let \mathbb{Z} be the set of integers. Suppose that there exists a function $f : \mathbb{Z} \rightarrow \mathbb{Z}$ which satisfies both of the following two conditions.

- $f(x) = x - 10$ for any $x > 100$, and
- $f(x) = f(f(x + 11))$ for any $x \leq 100$.

Answer the following questions.

- (a) Show that $f(100) = 91$.
- (b) Show that $f(x) = f(x + 1)$ holds for any integer x such that $90 \leq x < 100$.
- (c) Compute the value of $f(90)$. Show your steps.
- (d) Compute the value of $f(0)$. Show your steps.
- (e) Prove that $f(100 - n) = f(0)$ holds for any non-negative integer n .

Answer **Problem CS** 1 **(2)** with **CS** 1 **(1)**. If you select **CS** 1, then do not select **Math** 1. Clearly label the answer sheet at the top of the page as “**CS** 1 **(2)**” and do not write answers to other problems on the answer sheet.

Problem CS 1 **(2)**

Define a Boolean function $f(x, y, z)$ with three variables on the Boolean algebra $\{0, 1\}$ by

$$f(x, y, z) = (\bar{x} \cdot \bar{y} + y) \cdot \bar{z} + \bar{x} \cdot y + y + y \cdot z.$$

Answer the following questions.

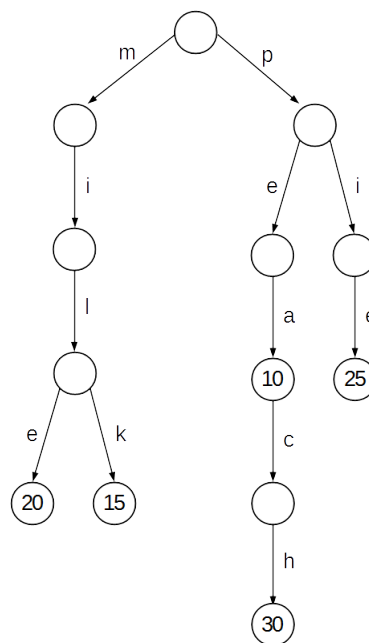
- (a) Draw the truth table of $f(x, y, z)$.
- (b) Find the conjunctive normal form of $f(x, y, z)$.
- (c) Find the disjunctive normal form of $f(x, y, z)$.
- (d) Simplify the disjunctive normal form of $f(x, y, z)$ obtained in (c) by using a Karnaugh map.
- (e) Represent $f(x, y, z)$ by a logic circuit where the only logic gates are NOT gates, AND gates, and OR gates so that the number of logic gates is as few as possible.

Answer **Problem CS 2 (1)** with **CS 2 (2)**. Clearly label the answer sheet at the top of the page as “CS 2 (1)” and do not write answers to other problems on the answer sheet.

Problem CS 2 (1)

Let us consider the implementation of a kind of associative array in which pairs of key and value are stored in a tree structure written in the C language. Assume that keys are strings consisting of lower-case letters a to z, and that values are positive integers.

The tree structure used is the rooted tree as illustrated below.



This figure illustrates an example of the rooted tree in which the five pairs listed in the following table are stored.

Keys	Values
mile	20
milk	15
pea	10
peach	30
pie	25

For representing nodes of the tree illustrated above, we use the following structure `node`.

```

1 struct node {
2     struct node *c[26];
3     int isEnd;
4     int value;
5 };
  
```

We store pointers to child nodes for the letters a to z in the entries of the array `c`. When there is no child node, `NULL` is stored. `TRUE` is stored in `isEnd` if the node is for the last letter of a key, and `FALSE` is stored otherwise. In case `TRUE` is stored in `isEnd`, the value paired with the key is stored in `value`.

With this structure, let us consider the implementation of the following functions.

- `struct node *new_node()`

Creates a new node, initializes it, and returns a pointer to the node.

- `void insert(struct node *root, char *key, int value)`

Inserts the pair of `key` and `value` into the rooted tree pointed by `root`. If `key` exists in the rooted tree already, replace the previous value with `value`.

- `int search(struct node *root, char *key)`

Checks whether a pair, whose key is `key`, is stored in the rooted tree pointed by `root`, and returns the value paired with the key. This function returns `-1` when such a pair is not stored.

- (a) The following program implements the functions `new_node`, `insert`, and `search`, using the structure `node`.

Fill in the blanks (A) to (F) to complete this program.

```

1  #define TRUE  1
2  #define FALSE 0
3
4  struct node *new_node()
5  {
6      int i;
7      struct node *n = (struct node *) malloc(sizeof(struct node));
8
9      for (i = 0; i < 26; i++) {
10         n->c[i] = NULL;
11     }
12     n->isEnd = FALSE;
13     return n;
14 }
15
16 void insert(struct node *root, char *key, int value)
17 {
18     int l;
19     int len = strlen(key);
20     struct node *n = root;
21
22     for (l = 0; l < len; l++)
23     {
24         int i = key[l] - 'a';
25         if ( (A) ) {
26             n->c[i] = new_node();
27         }
28         n = (B) ;
29     }
30     n->isEnd = (C) ;
31     (D) ;
32 }
33
34 int search(struct node *root, char *key)
35 {
36     int l;
37     int len = strlen(key);
38     struct node *n = root;
39
40     for (l = 0; l < len; l++)
41     {
42         int i = key[l] - 'a';
43
44         if (n->c[i] == NULL)
45             return -1;
46
47         n = (E) ;
48     }

```

```

49     if (           (F)           ) {
50         return n->value;
51     } else {
52         return -1;
53     }
54 }

```

- (b) Find the asymptotic time complexity in the worst case when the function `search` is called with a rooted tree pointed by `root` in which n pairs are stored and a key `key` of length k , and give the reason.

Answer **Problem CS** 2 (2) with **CS** 2 (1). Clearly label the answer sheet at the top of the page as “**CS** 2 (2)” and do not write answers to other problems on the answer sheet.

Problem CS 2 (2)

Let us consider the program list written in C language in the following page. Answer the following questions.

- (a) This program implements a buffer for variable-length character strings by using a fixed-length array. The function defined from line 8 to line 22 in the program list stores a given character string to a buffer. Now suppose there is a `struct buffer` named `buf`, whose `head` and `tail` are set to 12 and whose `store` are entirely initialized with `'\0'` (a NULL character). Show the content of `buf.store` when the following sequence of function executions has been performed.

```
put_str(&buf, "ten");
put_str(&buf, "six");
put_str(&buf, "three");
put_str(&buf, "four");
put_str(&buf, "seven");
put_str(&buf, "two");
put_str(&buf, "eight");
```

- (b) The function defined from line 24 to line 35 in the program list retrieves a character string from a buffer. The function returns 0 when there are no strings to retrieve. Otherwise, the function copies the string which starts at `head` to the memory region pointed by `dest` and returns 1. Fill in the blanks (A) to (D) to complete the function. Do not use function calls within the blanks.
- (c) Is the data structure implemented by this program FIFO (First-In, First-Out) or LIFO (Last-In, First-Out)?
- (d) This program might result in a malfunction if an overflow occurs on the value of `head` or `tail`. To fix the issue, let us consider adding the following three lines of code between line 33 and line 34. Fill in the blank (E) to prevent the overflow. Do not use function calls within the blank.

```
i = (E);
buf->head -= i;
buf->tail -= i;
```

```

1  #define BUFSIZE 24
2
3  struct buffer {
4      char store[BUFSIZE];
5      int head, tail;
6  };
7
8  int put_str(struct buffer *buf, char *str) {
9      int i = buf->tail;
10
11     while (i - buf->head < BUFSIZE) {
12         buf->store[i++ % BUFSIZE] = *str;
13         if (*str == '\0') {
14             buf->tail = i;
15             return 1;
16         } else {
17             str++;
18         }
19     }
20
21     return 0;
22 }
23
24 int get_str(struct buffer *buf, char *dest) {
25     int i = buf->head;
26
27     if (i == ) return 0;
28
29     do {
30         *dest = ;
31     } while () != '\0');
32
33      = i;
34     return 1;
35 }

```