

7 枚のうち 1

受験番号 MC-

【注意】 1 は必ず解答すること。2 から 6 までについては、そのうちの 3 問を選び解答せよ。解答用紙は対応する問題番号のものを用い、選択しなかったすべての問題の解答用紙全体に、大きく×を付すこと。裏面を用いる場合は表面の最下行に、その旨を明記すること。解答の指示を守らないときには、本科目の採点を行わない場合がある。

1

図 1-1 および図 1-2 は循環・双方向リストとその操作を行う C 言語のプログラムの一部である。以下の問いに答えよ。

- [1] リストにおけるノードの挿入と削除に関して、単方向リストと比較した場合の双方向リストの利点を 40 字以内で説明せよ。
- [2] 図 1-1、図 1-2 の空欄 (A)~(E) を埋めよ。ただし、空欄 (B)、(C) は複数行になっても構わない。

```
#include <stdio.h>
#include <stdlib.h>

/* 循環・双方向リストの構造体 */
typedef struct LISTNODE {
    int data; /* リストの要素 */
    struct LISTNODE *pNext; /* 次のノードを指すポインタ */
    struct LISTNODE *pPrev; /* 前のノードを指すポインタ */
} ListNode;

/* 先頭のノードを指すダミーノード。rootNode のメンバである pNext
   および pPrev は rootNode 自身を指すように初期化されている。 */
ListNode rootNode = {0, &rootNode, &rootNode};

/* rootNode から数えて num 番目 (rootNode->pNext が指すノードを
   1 番目とする) のノードを取得する。失敗した場合に NULL を、成功した場
   合に取得したノードへのポインタを返す。 */
ListNode *getNode(int num) {
    int i;
    ListNode *curNode = &rootNode;
    for (i = 0; i < num; i++) {
        (A);
        if (curNode == &rootNode) return NULL;
    }
    return curNode;
}
```

(次ページの図 1-2 に続く)

図 1-1 双方向リストの操作を実現するプログラムの一部

```
/* rootNode から数えて num 番目 (rootNode->pNext が指すノードを 1 番目とする) のノードを追加する。失敗した場合に 0 を、成功した場合に 1 を返す */
int addNode(int num, int val) {
    ListNode *prevNode = getNode(num - 1);
    ListNode *newNode;
    if (prevNode == NULL) return 0;
    newNode = (ListNode*)malloc(sizeof(ListNode));
    newNode -> data = val;
```

(B)

return 1;

}

```
/* rootNode から数えて num 番目 (rootNode->pNext が指すノードが 1 番目) を削除する。失敗した場合に 0 を、成功した場合に 1 を返す */
int deleteNode(int num) {
```

ListNode *targetNode = getNode(num);

if (targetNode == NULL) return 0;

(C)

free(targetNode);

return 1;

}

```
/* リストの要素を先頭から順に表示する */
```

void showList(void) {

ListNode *currentNode = rootNode.pNext;

do {

printf("%d ", currentNode -> data);

(D)

} while (currentNode != &rootNode);

}

```
/* リストの末尾から逆順で各要素を表示する */
```

void reverseShowList(void) {

ListNode *currentNode = rootNode.pPrev;

do {

printf("%d ", currentNode -> data);

(E)

} while (currentNode != &rootNode);

}

図 1-2 双方向リストの操作を実現するプログラムの一部

7 枚のうち 3

受験番号	MC-
------	-----

2

以下の問いに答えよ。

二つの集合 X と Y に対して、差集合を $X \setminus Y$ 、対称差を $X \Delta Y = (X \setminus Y) \cup (Y \setminus X)$ 、集合の要素数を $|X|$ とする。 $A = \{a, b, c, d\}$ 、 $B = \{b, c, e\}$ 、 $C = \{1, 2, 3\}$ とするとき、以下の問いに答えよ。ただし、答えのみでよい。

[1] 差集合 $A \setminus B$ と対称差 $A \Delta B$ を求めよ。

[2] べき集合 2^C を求めよ。

[3] 2^C 上の同値関係

$$R_1 = \{(x, y) : x, y \in 2^C, |x| = |y|\}$$

に関して、同値類を全て列挙せよ。

[4] 2^C 上の同値関係

$$R_2 = \{(x, y) : x, y \in 2^C, x \Delta y \in \{\emptyset, C\}\}$$

に関して、同値類を全て列挙せよ。

整理番号

2021 年度 4 月入学 (2020 年度 10 月入学含む) 東京農工大学工学府博士前期課程

10

問題用紙

情報基礎

情報工学
専攻

7 枚のうち 4

受験番号 MC-

3

ある感染症が流行している地域では、全住民 10 万人のうち 100 人が感染しているが、誰が感染しているかは未知であるとする。この感染症に対する検査キットは、感染している人に対して 70% の確率で陽性反応を示し、感染していない人に対しては 99% の確率で陰性反応を示す。このとき以下の問いに答えよ。ただし、答えのみでよく、答えが分数となる場合には、約分しない形の分数表記のままで構わない。

- [1] この検査キットを使って全住民 10 万人を検査したとき、陽性反応を示す人数の期待値を答えよ。
- [2] 無作為に選んだ 1 人の被験者の検査結果は陽性であった。このとき、この被験者が感染している確率を答えよ。
- [3] 無作為に選んだ 1 人の被験者の検査結果は陰性であった。このとき、この被験者が感染している確率を答えよ。
- [4] 無作為に選んだ 1 人の被験者に対して検査を 2 回実施した結果、1 回目に陽性反応を示し、2 回目も陽性反応を示した。このとき、この被験者が感染している確率を答えよ。

7枚のうち5

受験番号 MC-

4

以下の文章で示されたソートのアルゴリズム X について考える。ただし、ソートすべき n 個の要素は `int` 型配列 `a` の `a[0]`, `a[1]`, ..., `a[n-1]` に格納されているものとし、アルゴリズム X はそれらの要素を小さい順に並べかえるものとする。

アルゴリズム X

`a[0]`, `a[1]`, ..., `a[n-1]` の最小値を選び、`a[0]` の要素と交換する
(ただし、最小値が複数存在する場合は添字が小さい方の要素を選ぶものとする。以下同様)
`a[1]`, `a[2]`, ..., `a[n-1]` の最小値を選び、`a[1]` の要素と交換する
`a[2]`, `a[3]`, ..., `a[n-1]` の最小値を選び、`a[2]` の要素と交換する
...

これを最後まで繰り返す

[1] アルゴリズム X を C 言語の関数として記述せよ。関数の名前は `sortX` とし、関数 `sortX` は配列 `a` および整数 `n` を受け取り、戻り値 (戻り値) は返さない関数とせよ。

[2] ソートすべき要素の個数を n としたとき、アルゴリズム X の最悪時間計算量と平均時間計算量をオーダーで示せ。ただし答えのみでよい。

[3] アルゴリズム X が安定なソートならばその理由を、そうでないならば反例を述べよ。

以下に示す例題および図 5-1 に示す解答例にならって〔1〕～〔3〕の問いに答えよ。但し、各問いにおいて解答は最終的な決定性有限オートマトン (DFA) だけを示せばよい。また、受理状態は図 5-1 のように二重丸で明示すること。なお、「言語 L を受理 (受容) する DFA」とは言語 L に含まれる記号列は全て受理 (受容) し、それ以外の記号列は全く受理 (受容) しない DFA を指す。

(例題) 言語 $L = \{w \in (0+1)^* \mid w \text{ は } 0 \text{ で終わる}\}$ を受理 (受容) する DFA を示せ。

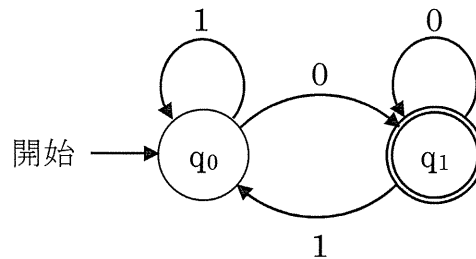


図 5-1 「言語 $L = \{w \in (0+1)^* \mid w \text{ は } 0 \text{ で終わる}\}$ を受理 (受容) する DFA」の解答例

〔1〕 言語 $L = \{w \in (0+1)^* \mid w \text{ は長さが奇数}\}$ を受理 (受容) する DFA を示せ。

〔2〕 言語 $L = \{w \in (0+1)^* \mid w \text{ は } 00 \text{ で終わる}\}$ を受理 (受容) する DFA を示せ。

〔3〕 言語 $L = \{w \in (0+1)^* \mid w \text{ は } 000 \text{ も } 111 \text{ も部分記号列として含まない}\}$ を受理 (受容) する DFA を示せ。

7 枚のうち 7

受験番号 MC-

6

1 から 6 までの 3 ビット 2 進数の数値を昇順に繰り返しカウントする同期型 6 進アップカウンタの設計について以下の問いに答えよ。

- [1] カウンタの現在の出力を Q_2 、 Q_1 、 Q_0 とする。ただし、 Q_2 を MSB とする。アップカウント後の出力を Q_2' 、 Q_1' 、 Q_0' とするとき、表 6-1 の真理値表を完成させよ。ただし、1~6 以外の場合の出力は考慮せず、Don't Care (*) として扱うこととする。

表 6-1

Q_2	Q_1	Q_0	Q_2'	Q_1'	Q_0'

- [2] 前問 [1] で完成させた真理値表をもとに、出力 Q_2' 、 Q_1' 、 Q_0' それぞれのカルノー図を作成し、簡単化した論理式をそれぞれ示せ。また、簡単化の際に、カルノー図においてどのようにまとめたかを、枠で囲むなどにより、図上で示すこと。ただし、本問題における簡単化した論理式とは、項数およびリテラル数が最も少ない積和形の論理式を指すものとする。
- [3] 前問 [2] で示した論理式をもとに、3 ビット同期型 6 進アップカウンタを、D フリップフロップを 3 個用いて実現した場合の回路図を示せ。

D フリップフロップの真理値表を表 6-2 に示す。

表 6-2

D	CLK	Q
L	↑	L
H	↑	H
*	その他	前の値

ただし、CLK はクロックを意味し、↑はクロックの立ち上がりを意味する。

- [4] 3 ビットのリングカウンタでは、000 や 111 の数値となった場合、この状態が永遠に続く。そのため、初期値がこれらの数値にならないように、特定の初期値を与える回路を付加するのが一般的である。前問 [3] で得られた 3 ビット同期型 6 進アップカウンタにおいては、初期値を与える必要はない。その理由を簡潔に述べよ。