

令和7年10月入学, 令和8年4月入学(第1回)  
 山口大学大学院創成科学研究科(工学系)博士前期課程入学試験  
 受験区分コード54 専門科目(データ構造とアルゴリズム(その1))

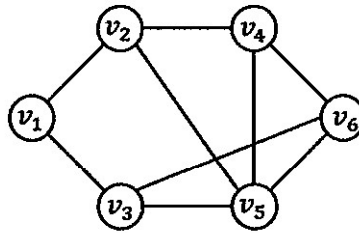
以下の間に答え, 解答を解答用紙(データ構造とアルゴリズム(その1 解答))に記入しなさい。

問1(配点50点)

枠内のアルゴリズムは, 無向グラフ $G=(V, E)$ と色集合 $C$ が与えられたときに接続する頂点は異なる色であるという制約のもとになるべく少ない色数で頂点を彩色するアルゴリズムである。このアルゴリズムを適用し, あるグラフの頂点彩色を行うとき, 以下の間に答えよ。ただし, 次数とは頂点に接続する辺の本数のことである。

1. 頂点集合 $V = \{v_1, v_2, \dots, v_n\}$ を次数が大きい順にソートする。
2. 1.でソートされた各頂点に対して, 順に以下の操作を行う。
  - 2.1 現在の頂点 $v_i$ の接続する頂点で使われていない最も小さい色番号で $v_i$ に色を塗る。

- 1) 下図のようなグラフに対して, 色集合 $C = \{c_1, c_2, c_3, c_4, c_5, c_6\} = \{\text{赤, 青, 黄, 緑, 橙, 紫}\}$ が与えられたとき, 上記の頂点彩色アルゴリズムを適用した場合の各頂点 $\{v_1, v_2, \dots, v_6\}$ に塗られた色をそれぞれ答えよ。ただし, 次数が等しい場合は頂点番号の小さいものから順に操作を行うものとする。



- 2) 上記のアルゴリズムの単純な実装では, 頂点ごとに毎回全ての接続する頂点の色を比較し, 使われていない最も小さい色番を探索することになる。このように実装した場合, グラフの頂点数を $n$ , ソートの時間計算量を $O(n \log n)$ とすると, アルゴリズム全体の時間計算量をオーダー表記で答えよ。
- 3) 上記のアルゴリズムは常に最小の色数で頂点を彩色することができるか否かをその理由とともに答えよ。
- 4) グラフの頂点彩色に関して以下の選択肢の中で内容が誤っているものを選び。
  - a. グラフがどのような場合でも最大次数と同じ色数があれば頂点を彩色することができる
  - b. グラフが木構造の場合は, 2色あれば頂点を彩色することができる
  - c. グラフの頂点数が $n$ であり, かつ $k$ 色( $k < n$ )で頂点彩色可能な場合, そのようなグラフは $k+1$ 色でも頂点を彩色することができる
  - d. グラフが頂点数 $k$ の完全グラフ(すべての頂点間に辺が存在するグラフ)を部分グラフとして含む場合は, 頂点を彩色するためには少なくとも $k$ 色の色が必要である

以下の問に答え, 解答を解答用紙 (データ構造とアルゴリズム(その2 解答)) に記入しなさい。

問2 (配点 50点)

枠内は, 有向非巡回グラフ  $G = (V, E)$  が与えられたときに, 図1から図2を得るように, 頂点を直線上に並べ, 各頂点からの辺の向きが一方向となるように並べる順を得るトポロジカルソートアルゴリズムである。ここでは, 指示された順に頂点を選択するとして, このアルゴリズムをグラフ  $G$  に適用し, 以下の問に答えよ。

1. 2つの空のリスト  $X, Y$  を用意し, リスト  $X$  には, 入力辺を持たない全ての頂点  $v$  を追加する。
2.  $X$  が空になるまで, 以下を繰り返す。
  - 2.1  $X$  から アルファベット順に頂点  $v_i$  を1つ取り出し,  $Y$  の最後に追加する。
  - 2.2 頂点  $v_i$  から出る全ての辺  $e$  が削除されるまで, 以下を繰り返す。
    - 2.2.1 頂点  $v_i$  が接続する頂点  $v_j$  を アルファベット順に 選択し, 頂点  $v_i$  と  $v_j$  を結ぶ辺  $e$  をグラフ  $G$  より除去する。
    - 2.2.2 この時点で頂点  $v_j$  が入力辺を持っていないならば,  $X$  に追加する。
3.  $Y$  に追加された頂点  $v$  を 追加順に並べ替えて 描いたグラフがトポロジカルソートの結果である。このとき, グラフ  $G$  に辺が残っている場合は, グラフ  $G$  で要件を満たす順が見つけれないことを意味する。

例として, 図1のグラフ  $G_0$  のトポロジカルソートを適用する場合,  $X = \{a, d\}$  として初期化され, 処理が開始される。最終的に  $Y = \{a, d, c, b\}$  が得られたとき, 図2のようにグラフとして描ける。

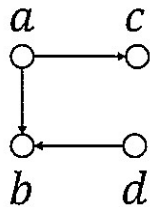


図1 グラフ  $G_0$

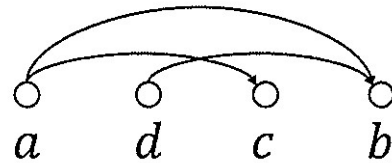


図2 グラフ  $G_0$  のトポロジカルソート結果

- 1) 図3のグラフ  $G_1$  にトポロジカルソートを適用し, その手順2. の繰り返し過程の中で, 各繰り返し回の 手順2.1の実行直前のタイミングでの, リスト  $X, Y$  の要素を示し, グラフ  $G_1$  を図示しなさい。ただし, 手順2.1が実行されない繰り返し回の枠内に, 大きくバツ (X) を書きなさい。

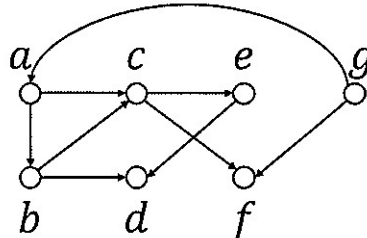


図3 グラフ  $G_1$

- 2) 解答欄1)に記載の8回の繰り返しの中で, トポロジカルソートアルゴリズムが要件を満たす順序を見つけられたか, 見つけられなかったかを選択肢にチェックして答えなさい。見つけられた場合は, 図2のように結果を, 頂点を 左から順に並べるとして, 頂点名と辺を図示せよ。見つけられなかった場合は, 図示の必要はない。

令和7年10月入学, 令和8年4月入学 (第1回)  
山口大学大学院創成科学研究科 (工学系) 博士前期課程入学試験  
受験区分コード54 専門科目 (プログラミング(その1))

以下の問に答え, 解答を解答用紙 (プログラミング (その1 解答)) に記入しなさい。

問1 (配点30点)

プログラム1を実行すると, 出力1を得た。空欄(A)~(F)を埋め, プログラムを完成しなさい。

・プログラム1

```
#include <stdio.h>

int main(void)
{
    int i, j;
    int a[2][5] = {{1, 0, 3, 0, 5},
                  {0, 2, 0, 4, 0}};

    printf("Input\n");

    for (i = 0; i < (A); i++) {
        for (j = 0; j < (B); j++) {
            printf("%d ", (C));
        }
        printf("\n");
    }
    printf("\n");

    printf("Result1\n");

    for (i = 0; i < (B); i++) {
        for (j = 0; j < (A); j++) {
            printf("%d ", (D));
        }
        printf("\n");
    }
    printf("\n");

    printf("Result2\n");

    for (i = 0; i < (A); i++) {
        for (j = 0; j < (B); j++) {
            printf("%d ", (E));
        }
        printf("\n");
    }
    printf("\n");
```

```
printf("Result3\n");

for (i = 0; i < (B); i++) {
    for (j = 0; j < (A); j++) {
        printf("%d ", (F));
    }
    printf("\n");
}
printf("\n");

return (0);
}
```

・出力1

```
Input
1 0 3 0 5
0 2 0 4 0

Result1
0 1
2 0
0 3
4 0
0 5

Result2
0 4 0 2 0
5 0 3 0 1

Result3
5 0
0 4
3 0
0 2
1 0
```

令和7年10月入学, 令和8年4月入学(第1回)  
山口大学大学院創成科学研究科(工学系)博士前期課程入学試験  
受験区分コード54 専門科目(プログラミング(その2))

以下の問に答え, 解答を解答用紙(プログラミング(その2 解答))に記入しなさい.

問2(配点20点)

プログラム2を実行すると, 出力2を得た. 空欄(A)~(D)を埋め, プログラムを完成しなさい.

・プログラム2

```
#include <stdio.h>

int c(int n, int r)
{
    if (r == 0 (A) r == n)
        return (B);
    else
        return c(n - 1, (C)) + c((D), r);
}

int main(void)
{
    int i, j;

    for (i = 0; i < 8; i++) {
        for (j = 0; j <= i; j++) {
            printf("%4d", c(i, j));
        }
        printf("\n");
    }

    return (0);
}
```

・出力2

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
```

令和7年10月入学, 令和8年4月入学 (第1回)  
山口大学大学院創成科学研究科 (工学系) 博士前期課程入学試験  
受験区分コード 54 専門科目 (プログラミング (その3))

以下の問に答え, 解答を解答用紙 (プログラミング (その3 解答)) に記入しなさい。

問3 (配点 25点)

プログラム3を実行したときの出力を答えなさい。

・プログラム3

```
#include <stdio.h>

int p = 5;

void update1(int p, int q)
{
    int r = 100;

    p = 50;
    q = q + 10;
    printf("update1: p = %d, q = %d, r = %d\n", p, q, r);
}

void update2(int *q, int *r)
{
    p = 20;
    *q = *q + 5;
    *r = p + *q;
    printf("update2: p = %d, *q = %d, *r = %d\n", p, *q, *r);
}

int main(void)
{
    int q = 10, r = 15;

    printf("main: p = %d, q = %d, r = %d\n", p, q, r);
    update1(p, r);
    printf("main: p = %d, q = %d, r = %d\n", p, q, r);
    update2(&q, &r);
    printf("main: p = %d, q = %d, r = %d\n", p, q, r);

    return (0);
}
```

令和7年10月入学, 令和8年4月入学(第1回)  
山口大学大学院創成科学研究科(工学系)博士前期課程入学試験  
受験区分コード54 専門科目(プログラミング(その4))

以下の問に答え, 解答を解答用紙(プログラミング(その4 解答))に記入しなさい。

問4(配点25点)

プログラム4を実行したとき, 出力4が得られる。空欄(A)~(E)を埋め, プログラムを完成しなさい。

・プログラム4

```
#include <stdio.h>

#define MAX_NAME 50
#define NUM_DATA 3

typedef struct {
    char name[MAX_NAME];
    int score;
} Record;

void process_data(Record *arr, int n)
{
    int i, j;
    Record temp;

    for (i = 0; i < n - 1; i++)
        for (j = 0; j < n - 1 - i; j++)
            if (arr[j].score  arr[j + 1].score) {
                ;
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
}

int main(void)
{
    int i;
    Record data[NUM_DATA];
    FILE *fp;

    fp = fopen("students.txt", "r");
    for (i = 0; i < NUM_DATA; i++)
        fscanf(fp, "%s %d", , );
    fclose(fp);

    process_data(data, NUM_DATA);

    for (i = 0; i < NUM_DATA; i++)
        printf("Name: %s, Score: %d\n",
               , );

    return (0);
}
```

・students.txt

```
Taro 80
Jiro 65
Hanako 90
```

・出力4

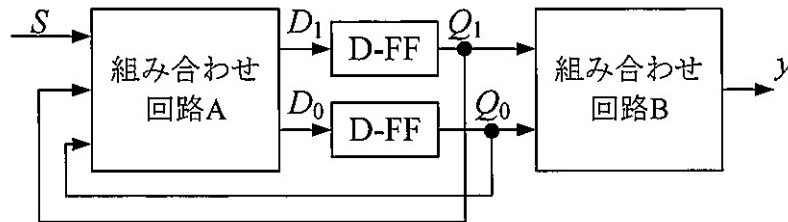
```
Name: Hanako, Score: 90
Name: Taro, Score: 80
Name: Jiro, Score: 65
```

令和7年10月入学, 令和8年4月入学 (第1回)  
 山口大学大学院創成科学研究科 (工学系) 博士前期課程入学試験  
 受験区分コード 54 専門科目 (計算機アーキテクチャ(その1))

以下の問に答え, 解答を解答用紙 (計算機アーキテクチャ(その1 解答)) に記入しなさい.

問1 (配点 50点)

以下の図に示すように, 入力  $S$  の値に従い 2 通りの状態遷移を行う順序回路を, 組み合わせ回路 A と 2 個の D フリップフロップ (D-FF) を用いて設計する. ここで, 状態  $(Q_1 Q_0)$  は,  $S=0$  のときに  $(00) \rightarrow (01) \rightarrow (11)$  の順で遷移を繰り返し,  $S=1$  のときに  $(11) \rightarrow (01) \rightarrow (00)$  の順で遷移を繰り返すものとする. また, 組み合わせ回路 B は,  $(Q_1 Q_0) = (00), (01), (11)$  において,  $Q_1 = Q_0$  のときに  $y=1$  を出力し,  $Q_1 \neq Q_0$  のときに  $y=0$  を出力するように設計する.



- (1-1) 上記の順序回路について, 入力  $S$ , 現在の状態  $(Q_1 Q_0)$ , 次の状態  $(Q'_1 Q'_0)$ , 及び各状態遷移に必要な D-FF の入力  $(D_1 D_0)$  からなる状態遷移表を作成せよ. また, 入力  $(Q_1 Q_0)$  に対する出力  $y$  の真理値表を作成せよ. なお, 想定外の入力に対する出力はドント・ケア (\*) とする.
- (1-2) 上記(1-1)の状態遷移表と真理値表に基づいて,  $D_1 = f_1(S, Q_1, Q_0)$  と  $D_0 = f_0(S, Q_1, Q_0)$  で定義される状態遷移関数, 及び  $y = g(Q_1, Q_0)$  で定義される出力関数のカルノー図を作成し, それぞれの関数の最簡形を求めよ. ただし, 使用できる論理ゲートは AND, OR, NOT のみとする.
- (1-3) 上記(1-2)で求めた状態遷移関数, 及び出力関数を満たす組み合わせ回路 A と B を図示せよ.
- (1-4) 上記(1-3)の回路を NOR ゲートのみを用いて図示せよ.

令和7年10月入学, 令和8年4月入学 (第1回)  
 山口大学大学院創成科学研究科 (工学系) 博士前期課程入学試験  
 受験区分コード 54 専門科目 (計算機アーキテクチャ(その2))

以下の問に答え, 解答を解答用紙 (計算機アーキテクチャ(その2 解答)) に記入しなさい.

問2 (配点 50 点)

右に示す CASL II のプログラムに関して以下の問に答えなさい.  
 なお, CASL II は仮想ハードウェア COMET II 上で動作するアセンブラ言語であり, 数値は 16 ビットの 2 進数で表現される.

(2-1) プログラムが終了するまでに, ラベル LP, L0, L1, LE の行が実行される回数を 10 進数で答えよ.

(2-2) プログラム終了時に, S に保存される数値を 10 進数および 16 進数で答えよ. さらに, R に保存される数値を 10 進数および 16 進数で答えよ.

(2-3) 25 行から 26 行を以下のように変更した場合を考える. プログラム終了時に, S に保存される数値を 10 進数および 16 進数で答えよ. さらに, R に保存される数値を 10 進数および 16 進数で答えよ.

- A DC 15, 32, 08, 10, 23
- B DC 52, 17, 42, 34, 01

(2-4) 25 行から 26 行の A, B を入力, 27 行から 28 行の S, R を出力とした場合, それらの意味と関係が分かるように, このプログラムの動作を文章で説明せよ.

行数	プログラム
1	LM START
2	LD GR1, N
3	LD GR2, =0
4	LD GR3, =0
5	LP CPA GR2, GR1
6	JZE LE
7	LD GR4, A, GR2
8	LD GR5, B, GR2
9	ST GR4, R, GR2
10	CPA GR4, GR5
11	JZE L0
12	JPL L1
13	SUBA GR3, =1
14	ST GR5, R, GR2
15	ADDA GR2, =1
16	JUMP LP
17	L0 ADDA GR2, =1
18	JUMP LP
19	L1 ADDA GR3, =1
20	ADDA GR2, =1
21	JUMP LP
22	LE ST GR3, S
23	RET
24	N DC 5
25	A DC 10, 21, 30, 41, 50
26	B DC 10, 20, 30, 40, 50
27	S DS 1
28	R DS 5
29	END

[命令形式の例]

- LD GRn, address, GRm : GRn ← M(address+GRm)
- ST GRn, address, GRm : GRn → M(address+GRm)
- ADDA GRn, address : GRn ← GRn+M(address)
- SUBA GRn, address : GRn ← GRn-M(address)
- CPA GRn, GRm : GRn = GRm の時に ZF = 1, GRn < GRm の時に SF = 1
- JPL address : SF = 0 かつ ZF = 0 すなわち直前の演算結果が正の時に PR ← address
- JZE address : ZF = 1 すなわち直前の演算結果が零 (zero) の時に PR ← address
- JUMP address : PR ← address
- DC N : 定数 N で指定したデータをメモリに保存
- DS N : 指定したワード N の領域を確保

ここで, M(address) は address 番地のメモリの内容, PR は Program Register (Program Counter), 「LD GRn, = 0」等の「= 0」は定数, ZF は Zero Flag, SF は Sign Flag を表す.