

令和 6 年度

千葉大学先進科学プログラム入学者選考課題

課題論述

情報

解答例

1

(出題意図)「方程式の解法」をテーマにプログラムを読む能力および書く能力を問うことを意図する。

問1では二次方程式の解の公式から、式の係数による場合分けを行え、それをプログラムで表せるかを問う。

問2は二分法と呼ばれるアルゴリズムに関する問である。問2-1, 問2-2は探索アルゴリズムとしての二分法であり、問2(1)でプログラムを読み、理解する能力、問2(2)では再帰表現で書ける能力を問う。問2(3)は方程式を解くための二分法であり、問2(1), 問2(2)をヒントにして実現する創造力を問う。

問1の解答（プログラム）例：

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

int main() {
    double a, b, c;
    double root, root1, root2;
    double discriminant, realPart, imaginaryPart;

    printf("方程式  $ax^2 + bx + c = 0$  の解を求めます。 ¥n");

    printf("係数 a を入力してください。 :");
    scanf("%lf", &a);

    printf("係数 b を入力してください。 :");
    scanf("%lf", &b);

    printf("係数 c を入力してください。 :");
    scanf("%lf", &c);

    if (a == 0) {
        if (b == 0) {
            if (c == 0) {
                printf("無限の解があります。 ¥n");
            } else {
                printf("解は存在しません。 ¥n");
            }
        } else {
            double root = -c / b;
            printf("1 次方程式の解: %lf¥n", root);
        }
    } else {
        double discriminant = b * b - 4 * a * c;

        if (discriminant > 0) {
            root1 = (-b + sqrt(discriminant)) / (2 * a);
            root2 = (-b - sqrt(discriminant)) / (2 * a);
            printf("実数解 1: %lf¥n", root1);
            printf("実数解 2: %lf¥n", root2);
        } else if (discriminant == 0) {
            root = -b / (2 * a);
            printf("重解: %lf¥n", root);
        } else {
            realPart = -b / (2 * a);
            imaginaryPart = sqrt(-discriminant) / (2 * a);
            printf("虚数解 1: %lf + %lfi¥n", realPart, imaginaryPart);
            printf("虚数解 2: %lf - %lfi¥n", realPart, imaginaryPart);
        }
    }
}
```

```

    }

    return 0;
}

```

解答例

```

ア a == 0
イ b == 0
ウ c == 0
エ 解は存在しません。
オ if (discriminant > 0)
カ, キ (-b + sqrt(discriminant)) / (2 * a), (-b - sqrt(discriminant)) / (2 * a)
ク else if (discriminant == 0)
ケ -b / (2 * a)
コ else
サ -b / (2 * a)
シ sqrt(-discriminant) / (2 * a)
ス, セ %lf + %lf, %lf - %lf

```

==

問2 (1) の解答例

入力した target と一致する arr[] の配列番号を表す。
target が配列にない場合は -1 を出力する。

==

問2 (2) のプログラム例

```

int bisection(int arr[], int left, int right, int target) {
    int mid
    if (left > right) {
        return -1;
    }

    mid = left + (right - left) / 2;

    if (arr[mid] == target) {
        return mid;
    } else if (arr[mid] < target) {
        return bisection (arr, mid+1, right, target);
    } else {
        return bisection (arr, left, mid-1, target);
    }
}

```

解答例

ア -1

イ mid

ウ arr, mid+1, right, target

エ arr, left, mid-1, target

==

問2 (3) のプログラム例

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

```
#include <math.h>
```

```
double f(double x){
```

```
    return(cos(x)-0.3); ←解答としては求めていない。
```

```
}
```

```
double binarySearch(double a, double b, double tolerance) {
```

```
    double mid = (a + b) / 2;
```

```
    if (fabs(f(mid)) < tolerance) {
```

```
        return mid;
```

```
    }else if (f(a)*f(mid)<0) {
```

```
        return binarySearch(a, mid, tolerance);
```

```
    } else {
```

```
        return binarySearch(mid, b, tolerance);
```

```
    }
```

```
}
```

```
int main() {
```

```
    double a, b;
```

```
    double tolerance = 0.0001;
```

```
    printf("探索範囲の最小値を入力してください。:");
```

```
    scanf("%lf", &a);
```

```
    printf("探索範囲の最大値を入力してください。:");
```

```
    scanf("%lf", &b);
```

```
if (f(a)*f(b) > 0){
    printf("探索範囲が不適切です。解を導出できません。¥n");
}else{
    printf("方程式の解は x = %.6f です。¥n",
        binarySearch(a, b, tolerance));
}
return 0;
}
```

==

解答例

ア f(mid)

イ mid

ウ f(a)*f(mid)

エ a, mid, tolerance

オ mid, b, tolerance

カ f(a)*f(b)

キ a, b, tolerance

2

(出題意図)

関数の再起呼び出しを理解できているかを問う。C 言語のプログラムを読み取り動作を理解できるか、および説明できるかを問うている。

解答

問 1

(1)

f (403, 221);

f (221, 182);

f (182, 39);

f (39, 26);

f (26, 13);

の 5 回

(2) A,B の最大公約数を求めている。

問 2

(1)

(0,9)、(0,4)、(0,2)、(0,1)、(0,0)、(1,1)、(2,2)、(3,4)、(3,3)、(4,4)、(5,9)、(5,7)、
(5,6)、(5,5)、(6,6)、(7,7)、(8,9)、(8,8)、(9,9)

(2)

前半と後半を独立にソートした結果を統合している。

問 3

(1) 3 (w を挿入、o を挿入、a を e に置換)

(2) 3 個の引数の最小値を返す。

(3) ①削除, ②置換, ③挿入

(4) 最後の 1 文字が一致する場合は置換の必要がないので cost=0, 最後の 1 文字が一致しない場合は置換を 1 回行うので cost=1 となる。

(5) (ア) s b (イ) s a

3

(出題意図) 身近な話題で考えられる組合せ最適化問題の C 言語プログラムの理解を問う。具体的には、最大マッチング問題 (問 1) と安定結婚問題 (問 2-4) を扱う。

問 1

```
ア: pair[v] == -1          //女子 v に対するペアがない場合
イ: matrix, pair[v], wariate, pair    //女子 v とペアになる男子 pair[v] がペア
    を作れるとわかった場合 (matching 関数が, true が返す場合)
```

問 2

組合せ 3 不安定 理由: A と X
組合せ 4 不安定 理由: A と X
組合せ 5 不安定 理由: B と Z, C と Z
組合せ 6 不安定 理由: B と Z

問 3

```
ウ: wmyorimom1(wPrefer, w, m, m1) == false // 入替えの必要があるのは, 現在の
    ペアの m1 よりも m のほうが希望順位が高い場合, よって, false が返ってきた場合。
エ: wPair[w] = m; // 女子の w のペアとして m を格納。
オ: mFree[m] = true; // m のペアが存在するので, true に変更。
カ: mFree[m1] = false; // m1 のペアが存在しなくなったので, false に変更。
(エ, オ, カは, 順番が変わっても正解。)
```

問 4

安定な組合せ:
男子 2 が, 女子 4 とペア。
男子 0 が, 女子 5 とペア。
男子 3 が, 女子 6 とペア。
男子 1 が, 女子 7 とペア。

4

(出題意図) 本問の目的は 2次元画像処理に関する理解の確認である。

問 1

コード 3 では輝度値を反転する処理が書かれているため、入力画像で暗い画素は明るくなり、明るい画素は暗くなるように変換される。従って正解答はイである。

問 2

注目する画素の画素値が閾値以上の場合は画素値を 255、閾値未満の場合は 0 に設定する。

```
# 画素値を閾値に応じて二値化する
if (canvas[h][w] >= th): canvas[h][w] = 255
else: canvas[h][w] = 0
```

問 3

(1) まず全画素の画素値を合計し、最後に画素数で割って平均値を算出する。

```
for h in range(height):
    for w in range(width):
        # 画素値の合計を算出
        ave += canvas[h][w]
# 画素数で割って平均値を算出
ave = ave / (width * height)
```

(2) まず、前問の関数を使って平均値を算出したのち、定義に従って分散を算出する。

```
# 前問の関数を使って平均値を計算
ave = average(canvas, width, height)

for h in range(height):
    for w in range(width):
        # 各画素値と平均値との差の二乗和を計算
        var = var + (canvas[h][w] - ave)**2
# 画素数で割って分散を算出
var = var / (width * height)
```

- (3) 注目する画素の画素値に応じて、リストの要素に 1 を加えて度数分布を算出する

```
for h in range(height):
    for w in range(width):
        #各画素値に応じた階級に度数を 1 つ増やす
        val = canvas[h][w]
        hist[val] += 1
```

問 4

- (1) 与えられた閾値に応じて、 m_1 , m_2 , n_1 , n_2 を計算する。 n_1 ないし n_2 が 0 の場合は不適となることに注意する。

```
for h in range(height):
    for w in range(width):
        val = canvas[h][w]
        if 0 <= val < th:
            m1 += val
            n1 += 1
        else:
            m2 += val
            n2 += 1
# 画素数が 0 の場合は不適
if((n1 == 0) or (n2 == 0)):
    return -1

m1 = m1 / n1
m2 = m2 / n2
```

- (2) 前問の関数を用いて D を最大化する最適な閾値 opt_th を探索する。

```
D = calculate_D(canvas, width, height, th)

if D > max_D:
    max_D = D
    opt_th = th
```