

# アルゴリズム入門 共通問題 (2022 年度 A セメスター試験)

試験日時: 2023 年 1 月 31 日 第 5 限 (60 分) 答案用紙: 1 部 持ち込み一切不可

裏面にも問題があるので確認すること

内容に関する質問は受け付けない。問題の記述が曖昧な場合は、適切な仮定をおいて回答し、どのような仮定をおいたのか明記せよ。

## 問題 1

以下では、変数 `seq` は整数の配列で、互いに異なる 3 つ以上の要素からなるとする。

(1) `seq` の要素の最小値を求める関数 `minimum` (図 1) を書いた。空欄 `A`・`B` に適切な式を埋めよ。

(2) 図 1 の `minimum` に最小限の変更を行い、`seq` の要素の最大値を求める関数 `maximum` を実装したい。変更箇所と変更内容を簡潔に説明せよ。

(3) 関数 `minimum` を実行した際に、5 行目で変数 `m` を書き換えた回数を考える。 $N = \text{len}(\text{seq})$  として、回数の最大値を  $N$  の式で見積もれ。また `seq` がどのようなときに、書き換え回数が最大となるのか簡潔に述べよ。

(4) `seq` の要素の、2 番目の最小値を求める関数を考える。太郎さんは図 2 のように実装した。`seq = [3, 2, 1]` に対してこの関数を実行した際の、変数 `m2` の値の変化を答えよ。

(5)  $N = \text{len}(\text{seq})$  として、太郎さんの実装の漸近計算量を  $N$  に関するオーダーで回答せよ。なお、主要な計算量として、`seq` の要素の値を `seq[i]` で読み取った回数を考える。

(6) `seq` の要素の、2 番目の最小値を求める関数を、花子さんは図 3 のように実装した。`seq = [3, 2, 1]` に対してこの関数を実行した際の、変数 `m1`、`m2` の値の変化を答えよ。

(7) 花子さんの実装は、`seq` によっては 2 番目の最小値とは異なる値を返すことがある。そのような `seq` の具体例を一つ回答せよ。

(8) 花子さんの実装を修正し、どんな `seq` でも正しい答えを返すように図 4 の空欄 `C` ~ `E` を埋めよ。なお、空欄 `C` は複数行のプログラムで埋めてよい。`seq` の要素の値を読み取った回数がないものが望ましい。

```
def minimum(seq):
    m = seq[0]
    for i in range(1, len(seq)):
        if A:
            m = B
    return m
```

図 1: 関数 `minimum`

```
def second_minimum_a(seq):
    m2 = maximum(seq)
    for i in range(0, len(seq)):
        if m2 > seq[i] > minimum(seq):
            m2 = seq[i]
    return m2
```

図 2: 太郎さんのプログラム

```
def second_minimum_b(seq):
    m1 = seq[0]
    m2 = seq[0]
    for i in range(0, len(seq)):
        if seq[i] < m1:
            m2 = m1
            m1 = seq[i]
        elif seq[i] < m2:
            m2 = seq[i]
    return m2
```

図 3: 花子さんのプログラム

```
def second_minimum_b(seq):
    C
    for i in range(D, E):
        if seq[i] < m1:
            m2 = m1
            m1 = seq[i]
        elif seq[i] < m2:
            m2 = seq[i]
    return m2
```

図 4: 花子さんのプログラムの修正

## 問題 2

ある講義は  $n$  クラス ( $n > 1$ ) 開講しており、どのクラスでも同じ試験を行う。答えはクラスごとに高得点順に整列されている。これを全クラス分まとめ、高得点から順に整列したい。

入力は二次元配列  $s$  であり、配列  $s[i]$  は  $i$  番目のクラス ( $0 \leq i \leq n - 1$ ) の点数を高得点から順に並べたものである。なお、各クラスとも少なくとも 1 枚答案があるものとする。

(1) 2 クラスの点数を併せて高得点から順に並べる関数 `merge` (図 5) を作成した。空欄を埋めよ。なお、同じ記号の欄の内容は同じである。引数  $x$  と  $y$  はそれぞれ 1 クラス分の点数を高得点から順に並べたものである。また、`ita.array.make1d(n)` は  $n$  個の 0 が並んだ配列を返す関数である。

(2) 関数 `merge` の 2 つの引数  $x$  と  $y$  の長さをそれぞれ  $m_1, m_2$  とする。このときの関数 `merge` の漸近計算量を  $m_1$  と  $m_2$  を用いて表せ。

(3) 二次元配列  $s$  中の全クラスの成績を関数 `merge` を使って高得点から順に並べる関数 `mergeAll1` (図 6) を作成した。

(a) どのクラスもちょうど  $m$  人が試験を受けたとする。関数 `mergeAll1` の漸近計算量を  $m$  とクラス数  $n$  を用いて表わせ。

(b) 全クラスで合計  $N$  人が試験を受けたときの、 $N$  に関する計算量について考える。関数 `mergeAll1` の漸近計算量が最悪となるのは、クラス数  $n$  と合計人数  $N$  がどのような関係にあり、各クラスでの受験者がどのようになっている時か。また、その時の漸近計算量を  $N$  を用いて表わせ。

(4) 関数 `mergeAll1` の計算量を改善するため、図 7 の関数 `mergeAll2` を作成した。ただし、`s[x : y]` は配列  $s$  の  $x$  番目から  $y - 1$  番目までの要素を順に並べた配列を表す。

(a) `mergeAll2(s)` を実行すると関数 `mergeAll2` は合計何回呼ばれるか。配列  $s$  の長さ  $n$  を用いて表せ。

(b) 全クラスで合計  $N$  人が試験を受けたとする。関数 `mergeAll2` の漸近計算量を  $N$  とクラス数  $n$  を用いて表わせ。

```
def merge(x, y):
    r = ita.array.make1d(len(x) + len(y))
    i = 0
    j = 0
    for k in range(0, len(r)):
        if (i >= len(x) or
            (j < len(y) and x[i] < y[j])):
            r[k] = 
             =  + 1
        else:
            r[k] = 
             =  + 1
    return r
```

図 5: 関数 `merge`

```
def mergeAll1(s):
    r = s[0]
    for i in range(0, len(s) - 1):
        r = merge(r, s[i + 1])
    return r
```

図 6: 関数 `mergeAll1`

```
def mergeAll2(s):
    n = len(s)
    if n == 1:
        return s[0]
    else:
        return merge(mergeAll2(s[0 : n // 2]),
                    mergeAll2(s[n // 2 : n]))
```

図 7: 関数 `mergeAll2`

### 問題 3

二次方程式  $ax^2 + bx + c = 0$  ( $a \neq 0, b^2 - 4ac \geq 0$ ) の解を求めるプログラムについて考える。

(1) A 君は図 8 の関数 f1 を作成した。このプログラムは  $ax^2 + bx + c = 0$  の 2 つの解が  $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$  であることを利用している。

(a) 関数 f1 は長さ 4 の配列を返す。このうち前半 2 つの要素は二次方程式の解に対応する。後半 2 つの要素は何のためのもので、どのような結果が期待されているか。1~2 行程度で説明せよ。

(b) f1(1, -10000, 1) を実行すると次の結果が得られた。

[9999.999899999999, 0.000100000000111176632, 0.0, -1.1176630732023796e-09]

この結果では、単なる丸め誤差とは思えないほどの大きな誤差が生じている。この主な理由を 1~2 行程度で説明せよ。桁落ち・情報落ちなどとの関係があれば述べること。

(2) B 君は A 君のプログラムを参考にして図 9 の関数 f2 を作成した。なお、このプログラムは  $ax^2 + bx + c = 0$  の 2 つの解を  $x_1, x_2$  としたとき、 $x_1 x_2 = \frac{c}{a}$  であることを利用している。

(a) 関数 f2 には関数 f1 にはない処理が追加されている。この部分にはどのような効果が期待されているか。1~2 行程度で説明せよ。

(b) f2(1, -10000, 1) の結果は、f1(1, -10000, 1) の結果を比べたとき、どの部分は等しく、どの部分はどのように異なると予想できるか。簡単に説明せよ。

```
def f1(a, b, c):
    d = (b ** 2 - 4.0 * a * c) ** 0.5
    x1 = (-b + d) / a / 2
    x2 = (-b - d) / a / 2
    r1 = a * x1 ** 2 + b * x1 + c
    r2 = a * x2 ** 2 + b * x2 + c
    return [x1, x2, r1, r2]
```

図 8: 関数 f1

```
def f2(a, b, c):
    d = (b ** 2 - 4.0 * a * c) ** 0.5
    x1 = (-b + d) / a / 2
    x2 = (-b - d) / a / 2
    if b > 0:
        x1 = c / a / x2
    else:
        x2 = c / a / x1
    r1 = a * x1 ** 2 + b * x1 + c
    r2 = a * x2 ** 2 + b * x2 + c
    return [x1, x2, r1, r2]
```

図 9: 関数 f2