

2022年度 大学院入学試験問題

先端数理科学研究科 先端メディアサイエンス専攻（博士前期課程）

科目：専門科目

注意事項

- 1 試験時間は90分です。
- 2 外国人留学生志願者のみ、英語による解答を認めます。
- 3 問題は、〔I〕数学、〔II〕情報、〔III〕プログラミングの3分野から出題されています。3分野のうち2分野を選択し解答しなさい。
- 4 選択した2分野の解答用紙の選択欄に「○」を、選択しなかった1分野の解答用紙の選択欄に「×」を、それぞれ必ず記入しなさい。選択欄にこれら以外の記載がある場合や、空欄がある場合、すべての答案を無効とします。
- 5 選択しなかった分野を含め、すべての解答用紙および問題用紙に受験番号と氏名を記入しなさい。
- 6 問題用紙は10ページあります（表紙は含まず）。
- 7 解答は、解答用紙に記入しなさい。
- 8 問題用紙、解答用紙は試験終了後すべて回収します。

受験番号 _____

氏 名 _____

2021

工
期

正
期

工
期

正
期

[1] 数学

【問題1】線型写像 f と $a_1, a_2, a_3, b_1, b_2, b_3, b_4$ を

$$f : \begin{pmatrix} x \\ y \\ z \end{pmatrix} \longrightarrow \begin{pmatrix} x+y \\ y+z \\ z+x \\ x+y+z \end{pmatrix}$$

$$a_1 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, a_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, a_3 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, b_1 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, b_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, b_3 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, b_4 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

とする。 \mathbb{R}^3 の基底を $\{a_1, a_2, a_3\}$ とし、 \mathbb{R}^4 の基底を $\{b_1, b_2, b_3, b_4\}$ とするととき、これらの基底に関する線型写像 f の表現行列を求めよ。ただし、 $\{a_1, a_2, a_3\}$, $\{b_1, b_2, b_3, b_4\}$ がそれぞれ $\mathbb{R}^3, \mathbb{R}^4$ の基底になることの証明はしなくてよい。

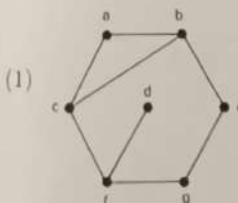
【問題2】

(1) 次の定積分の値を求めよ。

$$\int_{-1}^1 \frac{dx}{x^2 + 4x + 7}$$

(2) $D = \{(x, y) \mid x^2 + y^2 \leq 4, x \leq 0, 0 \leq y\}$ のとき、次の2重積分の値を求めよ。

$$\iint_D \frac{xy^2}{\sqrt{x^2 + y^2}} dx dy$$

【問題3】 v をグラフの点とするとき、 v から最も遠い点までのグラフ上の距離を v の離心値といい、離心値が最小となる点をそのグラフの中心という。次のグラフの中心を求め、中心となる点の記号を答えよ。

[11] 情報

【問題1】

- (1) 逆ポーランド記法で表された式
 $ABC + D / EF * +$
 を中置記法(通常の表記)で記述せよ。

- (2) 中置記法により表された式

$$\frac{A + 3B + C}{A - B/2}$$

を逆ポーランド記法で記述せよ。

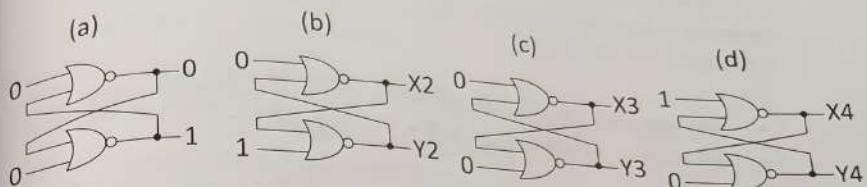
- (3) 逆ポーランド表記法の演算に必要なデータ構造は何か答えよ。

【問題2】

- (1) 10進小数において、2進数で表すと無限小数となるための条件を説明せよ。
- (2) 8ビットの符号付き整数の2進数 11011100 を右へ1ビット算術シフトした結果を2進数で示せ。
- (3) (2)で示した2進数 11011100 と算術シフトした結果それぞれについて、符号付き整数の数に変換した結果を示せ。

【問題3】

- (1) 次の回路において、(a), (b), (c), (d)の順に入力を与えた時、状態(b), (c), (d)における出力 $X_2, Y_2, X_3, Y_3, X_4, Y_4$ の値を求めよ。



- (2) この回路の名称は何か？

- (3) この回路の状態遷移図を書け。ただし、状態(a), (b)をそれぞれ、 S_a, S_b 、(a)から(d)の各状態の入力を $(0, 0), (0, 1), (0, 0), (1, 0)$ で表す。

2021

工
期

【問題4】

Web サーバ A、ブラウザ B、認証局 C が、それぞれ公開鍵 PK_A, PK_B, PK_C 、秘密鍵 SK_A, SK_B, SK_C を管理している。次のそれぞれで必要となる鍵を示せ。

- (1) A の公開鍵証明書を発行する時、C が使う鍵。
 (2) A の公開鍵証明書の中に格納されている全ての鍵。
 (3) B が A にアクセスを要求する時に用いる鍵。
 (4) A が (3) の B からの要求に答えてデータを送信する時に用いる鍵。

[III] プログラミング

この問題では Processing という言語でプログラムが記述されている。Processing は Java を基にしたプログラミング言語である。以下のプログラムの仕様を前提に各設問に答えよ。

- ・ 関数 `setup()` はプログラム実行開始時に一度だけ呼び出される。
- ・ 関数 `size(w, h)` は画面サイズを横 w ピクセル、縦 h ピクセルの大きさに設定する。
- ・ 関数 `line(x1, y1, x2, y2)` は座標 $(x1, y1)$ と座標 $(x2, y2)$ とを結ぶ線を描画する。
- ・ 関数 `background(255)` は画面を白色で塗りつぶす。
- ・ 関数 `print(x)` は引数 x で指定された内容を標準出力（コンソール）に表示する。
- ・ 関数 `println(x)` は引数 x で指定された内容を標準出力（コンソール）に表示した後、改行する。
- ・ 関数 `sqrt(x)` は x の平方根を求める。
- ・ 関数 `sin(a)` は正弦（サイン）を求める（ a の単位はラジアン）。
- ・ 関数 `floor(x)` は x の小数点以下を切り捨てた整数值を出力する。
- ・ 関数 `random(m, n)` は m 以上 n 未満の実数の乱数を出力する。
- ・ 配列 `array` に対し、`array.length` は、`array` の要素数を表す変数である。
- ・ 定数 `PI` は円周率として用いることができる。
- ・ 整数同士の演算では、その計算結果の小数点以下の値は切り捨てられる。

Processing における主な算術演算子

- + 足す（加算）
- 引く（減算）
- * 掛ける（乗算）
- / 割る（除算）
- % 割った数の余り（剰余）

【問題 1】

長さ N の配列 `a` には値 $0, 1, 2, \dots, N - 1$ が格納されている。いまこの配列の中身をシャッフルする、つまり順番をランダムに入れ替えるために、以下のアルゴリズムを用いるものとする。解答用紙のプログラムの空白部分を埋め、これを実現するプログラムを完成させよ。

変数 i を 0 から $N - 2$ まで 1 ずつ増加させながら以下を実行する：
 j に i 以上 N 未満のランダムな整数を代入する
 $a[i]$ と $a[j]$ の値を交換する

```
int[] a = {0, 1, 2, ..., N - 1};  
for (int i = 0; i <= N - 2; i++) {
```

```
}
```

2021

工
期

二
期

2021

工
期

工
期

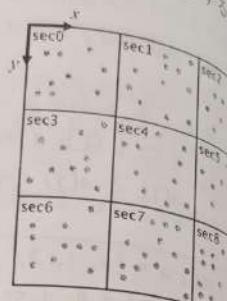
II
期

【問題2】

2次元平面領域内に散らばる点が100個ある。ただし、 i 番目 ($0 \leq i < 100$) の点の座標値 (x_i, y_i) が取り得る範囲は $0 \leq x_i < 600$, $0 \leq y_i < 600$ とする。

ここで、この平面領域を右図のように9個の矩形領域に均等に分割し、各領域内にある点の個数を数えて標準出力に表示する関数 `countPoints()` を作成して標準出力に表示する。解答用紙の空白部分を埋めプログラムを完成させよ。

ただし、各点の x 座標と y 座標はそれぞれ配列 `px` と `py` により与えられるものとする。なお、图形を描画する必要はない。



```
// 配列 px, py にはそれぞれ 100 個ずつ値が入っているものとする
int[] px = {10, 380, 190, ..., 250, 550};
int[] py = {590, 23, 150, ..., 400, 142};

void countPoints() {
    int[] count = new int[9];

    for (int i = 0; i < 100; i++) {
        int sectionIndex = ...;
        count[sectionIndex]++;
    }

    for (int i = 0; i < 9; i++) {
        System.out.println("sec" + i + " の点の個数：" + count[i]);
    }
}
```

【問題3】

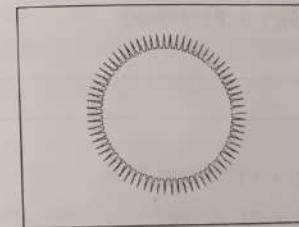
以下はある波形を画面に描画するプログラムである。`(A)～(D)`の実行結果に對応する処理①と処理②の組み合わせを(ア)～(オ)の中から選べ。

```
int N = 360, w = 400, h = 300;
float[] px = new float[N];
float[] py = new float[N];
```

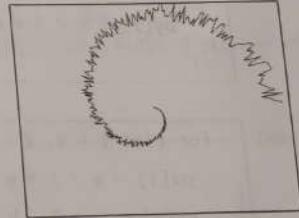
```
void setup() {
    size(400, 300);
    処理①
    処理②
}
```

```
background(255);
for (int i = 0; i < N - 1; i++) {
    line(px[i], py[i], px[i+1], py[i+1]);
}
}
```

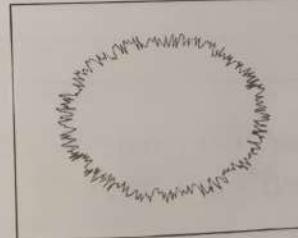
(A)



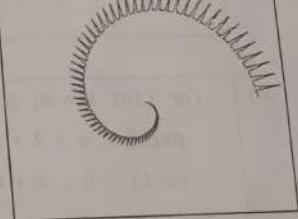
(B)



(C)



(D)



(ウ)

```

for (int i = 0; i < N; i++) {
    float dx = px[i] - w / 2;
    float dy = py[i] - h / 2;
    float d = 0.9;
    if (i % 5 == 0) d = 1.1;
    px[i] = d * dx + w / 2;
    py[i] = d * dy + h / 2;
}

```

(イ)

```

for (int i = 0; i < N; i++) {
    float dx = px[i] - w / 2;
    float dy = py[i] - h / 2;
    float d = random(0.9, 1.1);
    px[i] = d * dx + w / 2;
    py[i] = d * dy + h / 2;
}

```

(ウ)

```

for (int i = 0; i < N; i++) {
    px[i] = w / 2 + 0.5 * i * cos(i * PI / 180);
    py[i] = h / 2 + 0.5 * i * sin(i * PI / 180);
}

```

(エ)

```

for (int i = 0; i < N; i++) {
    px[i] = w / 2 + w / 3 * cos(i * PI / 180);
    py[i] = h / 2 + h / 3 * sin(i * PI / 180);
}

```

(オ)

```

for (int i = 0; i < N; i++) {
    px[i] = w / 2 + 0.25 * w * cos(i * PI / 180);
    py[i] = h / 2 + 0.25 * w * sin(i * PI / 180);
}

```

【問題4】

都市 0 (ゼロ) から都市 3 までの 4 都市を巡る巡回セールスマン問題 (traveling salesman problem) を絶当たり方式で解くことを考える。すなわち、都市 i から都市 j への移動距離が $d_{i,j}$ (ただし距離は整数とする) で与えられるとき、都市 0 から出発し、すべての都市を 1 回ずつ通ってふたたび都市 0 に戻ってくるまでの総移動距離が最小となるような経路を求める。

この問題を解くための以下のプログラムにおいて、関数 setup() から呼ばれている関数 checkRoute() を完成させよ。解答にあたって以下の注意を参照せよ。

- 移動経路の結果は配列 answer_route に格納すること
- $d_{i,j}$ の値を得るには関数 distance(i, j) を使用せよ (以下のプログラムには示されていない)
- 配列 route の要素に、通る都市の番号を順に格納した状態で関数 calcTotalDistance() を呼ぶと、都市 0 から出発してふたたび都市 0 に戻ってくるまでの総移動距離を返り値として得ることができる
- 配列 route の 0 番目から $n-1$ 番目までの要素中に、都市 c が含まれているかどうかを確認し、含まれている場合は true を、含まれていない場合は false を出力する関数 checkConflict(n, c) を使用してよい
- 追加で関数を定義してよいが、必ず解答欄中にそのプログラムを記すこと
- 簡単のため、総移動距離が 9999 を超えないことを想定してよい
- 総移動距離が最小となる経路が複数ある場合は、そのうちのいずれか一つが配列に格納されればよい

```

int[] route, answer_route;
int max_total_distance;

void setup() {
    route = new int[4];
    answer_route = new int[4];
    max_total_distance = 10000;
    route[0] = 0;
    checkRoute();
}

```

```
    println("Result:");
    for (int i = 0; i < 4; i++) {
        println(answer_route[i]);
    }
}

int calcTotalDistance() {
    int total_distance = 0;
    for (int i = 0; i < 3; i++) {
        total_distance += distance(route[i], route[i+1]);
    }
    total_distance += distance(route[3], route[0]);
    return total_distance;
}

boolean checkConflict(int n, int c) {
    for (int i = 0; i < n; i++) {
        if (route[i] == c) return true;
    }
    return false;
}

// 以下に関数 checkRoute()ほか必要な関数の定義を記せ
```

2022年度 大学院入学試験問題

先端数理科学研究科 先端メディアサイエンス専攻(博士前期課程)

科目: 専門科目

注意事項

- 1 試験時間は90分です。
- 2 外国人留学生志願者のみ、英語による解答を認めます。
- 3 問題は、(I) 数学、(II) 情報、(III) プログラミングの3分野から出題されています。3分野のうち2分野を選択し解答しなさい。
- 4 選択した2分野の解答用紙の選択欄に「○」を、選択しなかった1分野の解答用紙の選択欄に「×」を。
それぞれ必ず記入しなさい。選択欄にこれら以外の記載がある場合や、空欄がある場合、すべての答案を無効とします。
- 5 選択しなかった分野を含め、すべての解答用紙および問題用紙に受験番号と氏名を記入しなさい。
- 6 問題用紙は11ページあります(表紙は含まず)。
- 7 解答は、解答用紙に記入しなさい。
- 8 問題用紙、解答用紙は試験終了後すべて回収します。

2021

工
期Ⅱ
期

受験番号 _____

氏名 _____