

令和6年度(令和6年4月入学)
博士前期課程(修士課程)一般入試(第I期)
令和5年度(令和5年秋入学)
博士前期課程(修士課程)外国人留学生特別入試
情報工学専攻

数 学 (120分)

〔注意事項〕

1. 監督者の指示があるまで、問題冊子(この冊子)を開いてはいけません。
2. 配布物は、この問題冊子1部、解答用紙3枚と計算用紙1枚です。
3. 解答用紙には志望専攻名、受験番号を記入する欄がそれぞれ1箇所ずつあります。監督者の指示に従って、すべての解答用紙(合計3枚)の志望専攻名欄と受験番号欄に志望専攻名と受験番号を記入しなさい。
4. 解答は、問題番号に対応する解答用紙の指定された場所に書きなさい。解答を解答用紙の裏面に書いてはいけません。解答用紙、計算用紙の追加、交換はしません。
5. 問題は全部で3問あり、2ページにわたって印刷されています。落丁・乱丁および印刷の不鮮明な箇所などがあれば、手をあげて監督者に知らせなさい。
6. 問題冊子の白紙と余白は、計算などに使用してもよろしい。
7. 解答用紙は、持ち帰ってはいけません。
8. 問題冊子と計算用紙は、持ち帰りなさい。

1

4 次対称行列

$$A = \begin{pmatrix} 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 \end{pmatrix}$$

について、次の問いに答えよ。

- (1) A の固有値をすべて求めよ。
- (2) A の固有値のうち、最小のものに属する固有ベクトルをすべて求めよ。

2

(1) 次の極限を求めよ。

$$\lim_{x \rightarrow 0} \frac{x \cos x - \sin x}{x^3 \cos x}$$

(2) 2 変数関数 $f(x, y)$ を

$$f(x, y) = x^3 + y^3 - 3xy$$

と定める。

- (i) $x < 0, y < 0, f(x, y) = 0$ を満たす実数 x, y は存在しないことを示せ。
- (ii) $x = \sqrt[3]{2}$ の近くで定義された C^2 級関数 $\varphi(x)$ が $f(x, \varphi(x)) = 0, \varphi(\sqrt[3]{2}) = \sqrt[3]{4}$ を満たすとする。このとき、 $\varphi'(\sqrt[3]{2}), \varphi''(\sqrt[3]{2})$ の値をそれぞれ求めよ。

- 3 実数 a は, $a \leq \frac{1}{2}$ を満たすとする。2次元確率変数 (X, Y) は, 次で定義される関数 $f(x, y)$ を同時密度関数としてもつとする。

$$f(x, y) = \begin{cases} \frac{1}{1-a}(1-ax-ay) & (0 \leq x \leq 1, 0 \leq y \leq 1 \text{ のとき}) \\ 0 & (\text{それ以外のとき}) \end{cases}$$

- (1) X の期待値 $E(X)$ を求めよ。
(2) XY の期待値 $E(XY)$ および, X, Y の共分散 $\text{Cov}(X, Y)$ の値を求めよ。ただし, 共分散公式

$$\text{Cov}(X, Y) = E(XY) - E(X)E(Y)$$

を証明なしに用いてよい。

- (3) $\text{Cov}(X, Y) = 0$ を満たす a の値をすべて求めよ。また, a がそのような値である場合に, X, Y が独立かどうかを判定せよ。

(以上)

令和6年4月入学 一般入試（第I期）

April Admission for the 2024 Academic Year for Japanese Students (1st period)

令和5年秋入学 外国人留学生特別入試 Fall Admission for the 2023 Academic Year for International Students

京都工芸繊維大学 大学院工芸科学研究科 博士前期課程（修士課程） 情報工学専攻 試験問題

Question booklet of Entrance Examination for Master's Program of Information Science,

Graduate School of Science and Technology, Kyoto Institute of Technology (KIT)

専門科目 Special Subject

[注意事項 Cautions]

1. この問題冊子は合図があるまで中を開かないでください。この中身は以下の2題であり、2題とも必須です。落丁・乱丁および印刷の不鮮明な箇所などがあれば、手を挙げて監督者に知らせなさい。

Do not open this question booklet until permitted by the proctor. Answer all two subject parts listed below.

Raise your hand and inform the proctors of any missing pages, disarranged pages, unclear printing, etc.

プログラミング（1） Programming (1)

プログラミング（2） Programming (2)

2. 配布物は、この問題冊子1部、解答用紙2枚、および下書き用紙1枚です。

The proctors distribute this question booklet, two answer sheets, and one memo sheet.

3. 机の上には受験票以外に、次のものを置いてもよろしい。

You can put the following goods in addition to your exam admission ticket.

(1) 黒鉛筆とシャープペンシル Black pencils and mechanical pencils

(2) プラスチック製の消しゴム Plastic erasers

(3) 電動でない小型の鉛筆削り Small-sized non-electric pencil sharpeners

(4) 秒針音がしない小型の時計（辞書、電卓、通信等の機能があるものは不可）

Small-sized silent watches or clocks without any additional dictionary, calculator, communication, etc.

(5) 無地のマスク Plain masks

(6) 眼鏡、ハンカチ、目薬、ティッシュペーパー（袋又は箱から中身だけを取り出したもの）

Glasses, handkerchiefs, eye drops, tissues without package

これら以外については監督者の了解を受けてください。

Ask the proctors for permission to use any goods other than the above.

4. プログラミング（1）とプログラミング（2）を別の解答用紙に解答してください。解答用紙2枚すべての上欄指定枠内に、問題科目名、志望専攻名、受験番号を忘れずに記入してください。解答用紙の裏面に解答を書いても構いません。解答用紙と下書き用紙の追加配布はしません。

Use a separate answer sheet for each subject part. Fill in the subject-part name, the major of Master's Program, and your examinee's number in the designated boxes on all two answer sheets. You can use both sides of the answer sheet. No additional sheet is available.

5. この問題冊子はバラしても構いません。 You can unbind this booklet.

6. 試験終了後も退出の許可があるまで退室はできません。中途退室できません。

Do not leave the room after the exam until permitted by the proctor. Also, you do not during the exam.

7. 問題冊子と下書き用紙は持ち帰ってください。

Bring this question booklet and the memo sheets when you leave the room after the exam.

プログラミング(1) [1/3]

問 1 C 言語で記述された Program1 は、非負の整数(integer)の階乗(factorial)を求めるプログラムである。下記の(あ)、(い)を埋めて、Program1 を完成せよ。また、入力 num の値が 4 のとき、標準出力(standard output)に出力される内容を示せ。

Program1

```
#include <stdio.h>
int factorial(int n){
    static int cnt=0;
    printf("cnt=%d\n", ++cnt);
    if (n>0)    return (あ) ;
    else       return (い) ;
}
int main(void){
    int num;
    scanf("%d", &num);
    printf("Factorial of %d is %d", num, factorial(num));
    return 0;
}
```

問 2 C 言語で記述された Program2 を実行したとき、標準出力(standard output)に出力される内容を示せ。

Program2

```
#include <stdio.h>
int main(void){
    struct c{
        char x;
        char *y;
    }*p;
    char b[3]={'d', 'e', 'f'};
    struct c a[3]={{'g', b}, {'j', (b+1)}, {'i', (b+2)}};
    p=a;
    printf("%c\n", p++->x);
    printf("%c\n", ++p->x);
    printf("%c\n", (p-1)->x);
    printf("%c\n", *(p->y));
    return 0;
}
```

[次ページに続く]

プログラミング(1) [2/3]

問 3 時間計算量(time complexity)について、以下の問(a), (b), (c)に答えよ。

- (a) 計算時間が、入力サイズ n の関数として、次の式で表される 3 つのアルゴリズム(algorithm)A, B, C がある。それぞれの時間計算量のオーダー(order)を求めよ。解答用紙には最もタイト(tight)な答えだけを記すこと。

注意：例えば、 $f(n) = n^2$ のとき、 $O(n^2)$, $O(n^3)$, $O(n^4)$, ... 等のいずれも解答となりうるが、最もタイトな(すなわち、最小(minimum)の) $O(n^2)$ を解答とすること。

A. $f_A(n) = 3.1n\sqrt{n} + 3.5n\log n$

B. $f_B(n) = n\log n + n^2$

C. $f_C(n) = 3e^n + 2^n$

- (b) n が十分大きくなったとき、上記の 3 つのアルゴリズム A, B, C の中で一番高速に実行できるものはどれか。また、2 番目に高速なのはどれか。解答用紙にはそれぞれの答え(A, B, C のいずれか)だけを記すこと。

- (c) $f(n) = 3n^2 + 2n + 4$ のときのオーダーが $O(n^2)$ であることを、以下に示すオーダーの定義に従い、証明せよ。

オーダーの定義：

2 つの関数 $f(n)$ と $g(n)$ に対して、ある正の定数 c と n_0 が存在し、 $n \geq n_0$ なる全ての n に対して、常に $f(n) \leq c \cdot g(n)$ となるとき、 $f(n)$ はオーダー $g(n)$ であるという。

問 4 数列(numerical sequence)を単方向連結リスト(singly linked list)で管理するプログラムについて考える。Program3 は、初期状態が {12, 23, 45, 50} という数列の場合の C 言語で記述されたプログラムである。下記の条件(a), (b)を満たすように、(あ), (い), (う)を埋めて、Program3 を完成せよ。

(a) 要素 12 と要素 23 の間に要素 20 を追加する。

(b) 要素 45 を削除する。

[次ページに続く]

プログラミング(1) [3/3]

Program3

```
#include <stdio.h>

struct list {
    int data;
    struct list *next;
};

void printlist(struct list *ptr) {
    while(ptr != NULL) {
        printf("%d ", ptr->data);
        ptr = ptr->next;
    }
    printf("%n");
}

int main() {
    struct list p0, p1, p2, p3, p4;
    struct list *head;
    p0.data = 12;    p0.next = &p1;
    p1.data = 23;    p1.next = &p2;
    p2.data = 45;    p2.next = &p3;
    p3.data = 50;    p3.next = NULL;
    head = &p0;
    printlist(head);

    p4.data=20;
    (あ) ;
    (い) ;
    printlist(head);

    (う) ;
    printlist(head);

    return 0;
}
```

プログラミング(2) [1/3]

問1 C言語で記述された下記 Program1 は、モンテカルロ法(Monte Carlo method)を用いて円周率(circular constant) π の近似値(approximate value)を求める関数(function) pi である。プログラム中の空欄(あ)、(い)、(う)を埋めて、関数 pi を完成せよ。なお、引数(argument) n は試行回数(number of trials)である。また、解答には、後述する関数 rand と定数(constant) RAND_MAX を用いてよい。

Program1

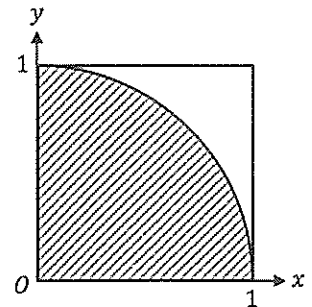
```
double pi(int n) {
    int i;
    int m = 0;
    double x, y;

    for(i=0; i<n; i++) {
        x = (あ);
        y = (あ);
        if( (い) < 1)
            m++;
    }

    return (う);
}
```

モンテカルロ法により円周率 π の近似値を求めるアルゴリズムの概要

- 0 以上 1 以下の一様乱数(uniform random number)を独立に(independently)2つ生成して、それぞれを x 、 y とする。
- (x, y) を xy 平面(xy -plane)上の1点としてプロット(plot)する。
- 1、2の試行(trial)を n 回繰り返す。したがって、プロットされた点の総数(total number)は n となる。
- 原点(origin)を中心(center)とした半径(radius)1の円(circle)の領域(area) (右図の斜線の領域) 内にプロットされた点の数を m とする。
- n と m の比率(ratio) $n:m$ は、おおよそ(approximately) $1:\pi/4$ となることから、 π の近似値を得る。



Program1 の外部で定義されているライブラリ関数の説明

int rand(void);

機能：0 以上 RAND_MAX 以下の一様乱数を生成する。

RAND_MAX は、関数 rand の戻り値(return value)の最大値(maximum value)を表す

0 より大きい整数(integer)として定義されている。

戻り値：生成された乱数。

[次ページに続く]

プログラミング(2) [2/3]

問2 C 言語で書かれた下記 Program2 は、文字列(string) str を double 型の実数(real number)に変換(convert)した値を返す関数 to_double である。プログラム中の空欄(あ)、(い)、(う)を埋めて、関数 to_double を完成せよ。なお、文字列 str は、1つ以上の数字('0'~'9')と1つ以下の小数点('.')の組み合わせから構成され、10進数(decimal number)として扱われるものとする。

Program2

```
double to_double(const char *str) {
    double value = 0.0;
    const char *point = NULL;

    while(*str != '\0') {
        if(*str == '.')
            point = str;
        else if(point == NULL)
            value = value * (あ) + ((い));
        else
            value += pow((あ), (う)) * ((い));
        str++;
    }

    return value;
}
```

Program2 の外部で定義されているライブラリ関数の説明

```
double pow(double x, double y);
```

機能 : x の y 乗(y-th power of x)の値を求める。

戻り値 : x の y 乗の値。

問3 Java 言語で記述された Program3 について、以下の問(a)~(c)に答えよ。

- Program3 を実行したときの、標準出力(standard output)への出力結果を示せ。
- main メソッド(method)内の for ループ(loop)の実行が完了した時点での front と rear の値をそれぞれ示せ。
- Program3 では、配列(array) a をリングバッファ(ring buffer)として用いている。リングバッファとはどのようなデータ構造(data structure)か、簡潔に説明せよ。

[次ページに続く]

プログラミング(2) [3/3]

Program3

```
class MyArray {
    int a[];
    int n, front, rear;

    public MyArray(int size) {
        n = size;
        a = new int[n];
        front = 0;
        rear = n - 1;
    }

    public void push(int value) {
        a[front++] = value;
        if(front > n - 1)
            front = 0;
    }

    public int pop() {
        if(--front < 0)
            front = n - 1;
        return a[front];
    }

    public void unshift(int value) {
        a[rear--] = value;
        if(rear < 0)
            rear = n - 1;
    }

    public int shift() {
        if(++rear > n - 1)
            rear = 0;
        return a[rear];
    }

    public static void main(String[] args) {
        MyArray array = new MyArray(10);

        for(int i=0; i<10; i++) {
            switch(i % 5) {
                case 0:
                    array.push(i); break;
                case 2:
                    System.out.println(array.pop()); break;
                case 4:
                    System.out.println(array.shift()); break;
                default:
                    array.unshift(i);
            }
        }
    }
}
```