令和3年度(令和3年4月入学)
博士前期課程(修士課程)一般入試(第Ⅰ期)
令和2年度(令和2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 (1) a, b を実数とする。$\mathbf{R}^4$ のベクトルの組

$$\vec{x} = \begin{pmatrix} 3 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \quad \vec{y} = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 1 \end{pmatrix}, \quad \vec{z} = \begin{pmatrix} 1 \\ 0 \\ -1 \\ -2 \end{pmatrix}, \quad \vec{w} = \begin{pmatrix} 0 \\ 1 \\ a \\ 2 \end{pmatrix}$$

は，$\mathbf{R}^4$ の基底であるとする。$\mathbf{R}^4$ から $\mathbf{R}^4$ への線形写像 T は，ベクトル $\vec{x}$, $\vec{y}$, $\vec{z}$, $\vec{w}$ をそれぞれ $\mathbf{R}^4$ のベクトル

$$\begin{pmatrix} 3 \\ 2 \\ 0 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 4 \\ 1 \\ -1 \end{pmatrix}, \quad \begin{pmatrix} 2 \\ 0 \\ 4 \\ b \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 0 \\ -1 \\ 5 \end{pmatrix}$$

に写すとする。

(i) $\mathbf{R}^4$ のベクトル $\vec{p} = \begin{pmatrix} 5 \\ 0 \\ -a+4 \\ 4 \end{pmatrix}$ を $\vec{x}$, $\vec{y}$, $\vec{z}$, $\vec{w}$ の線形結合として表せ。

また，$T(\vec{p})$ を求めよ。

(ii) a の取りうる値の範囲を求めよ。

(2) 3 次正方行列 $A = \begin{pmatrix} -4 & 0 & -6 \\ -3 & 2 & -3 \\ 3 & 0 & 5 \end{pmatrix}$ が対角化可能であるかどうか調べよ。

2 (1) (i) $\dfrac{d}{dx}\left(\dfrac{x}{x^2+1}\right) + \dfrac{1}{x^2+1}$ を計算し，その計算を用いて不定積分 $\displaystyle\int \frac{dx}{(x^2+1)^2}$ を求めよ。

(ii) a を正の定数とする。広義積分 $\displaystyle\int_0^\infty \frac{dx}{(x^2+a^2)^2}$ を求めよ。

(2) a および s を正の定数とする。広義積分 $\displaystyle\int_0^\infty \frac{x}{(x^2+a^2)^s} dx$ を求めよ。

3 確率変数 X, Y は独立であり，それぞれの密度関数 $f_X(x)$, $f_Y(y)$ は

$$f_X(x) = \begin{cases} 1 & (0 \leqq x \leqq 1) \\ 0 & (x < 0 \text{ または } x > 1) \end{cases}, \quad f_Y(y) = \begin{cases} 1 & (0 \leqq y \leqq 1) \\ 0 & (y < 0 \text{ または } y > 1) \end{cases}$$

で与えられている。

(1) X の期待値 $E[X]$, 分散 $V[X]$ を求めよ。

(2) X, Y の積 XY の期待値 $E[XY]$, 分散 $V[XY]$ を求めよ。

(3) $0 < a < 1$ を満たす実数 a に対し，確率 $P[X \geqq a]$ と $P[XY \geqq a]$ を求めよ。

(以上)

2021年4月入学 一般入試I期 April Admission for the 2021 Academic Year for Japanese Students (1st period)
2020年秋入学 外国人留学生特別入試 Fall Admission for the 2020 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 Specialized 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 may 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 functions such as dictionary, calculator, communication, etc.
   (5) 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、問題科目名と問題番号、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) [1/2]

問 1 C 言語で記述された Program 1 について考える。①、②、③の行にある printf 文による出力結果 (output) をそれぞれ示せ。

Program 1

```
int func(int x, int y) {
  return x > y ? x : (x < y ? y : 0);
}
int main() {
  int x, y;
/* ① */  x = 10; y = 11; printf("%d\n", func(x / y, y % x));
/* ② */  x = 10; y = 11; printf("%d\n", func(x, y--));
/* ③ */  printf("%d\n", func(x, y));
  return 0;
}
```

問 2 C 言語で記述された Program 2 について考える。なお、strcpy(char *dst, char *src) は src の文字列 (string) を dst にコピー (copy) する関数 (function) である。

Program 2

```
void func(char ①s) {
  ②s = (char *)malloc(sizeof(char) * 10);
  strcpy(③s,"Hello");
}
int main() {
  char *str;
  func(④str);
  printf("%s\n",str);
  return 0;
}
```

このプログラムでは、main 関数の printf 文において、Hello という文字列を表示させたい。このとき、空欄 ① ～ ④ を埋めるべき記号を以下の記号群より選び (a)～(f) で答えよ。なお、同じ記号を複数回使っても良い。空欄に何も記入する必要が無い場合は、(f) を選択せよ。

(a) & (b) * (c) ** (d) ++ (e) % (f) 何もなし

[次ページに続く]

プログラミング (1) [2/2]

問 3 C 言語で記述された Program 3 においては、int 型の配列 array、int 型の変数 pos、void 型の関数 push、int 型の関数 pop を定義している。これらについて、(a)〜(c) の問いに答えよ。

Program 3

```
int array[10] = {0};
int pos = 0;
void push(int x) { array[pos++] = x; }
int pop() { return array[--pos]; }
```

(a) これらの定義に対して、
push(10); push(20); pop(); push(30); pop();
を実行した後の配列 array の内容を答えよ。

(b) Program 3 で実現したデータ構造 (data structure) を何と呼ぶか。名称を答えよ。

(c) このプログラムは pos の値の下限 (lower limit) と上限 (upper limit) を決めていないので、push() や pop() を続けると不具合 (fault) が発生する可能性がある。どのような不具合が発生するか簡潔に述べよ。

問 4 Java 言語で記述された Program 4 について、これらのクラス定義 (class definition) をした上で、(a)〜(e) の問いに答えよ。

Program 4

```
class Fruit {
    String name;
    Fruit() { name = "fruit"; }
    void squeeze() { System.out.println(name + " juice"); }
}
class Orange extends Fruit {
    Orange() { name = "orange"; }
    void squeeze() { super.squeeze(); }
}
class Apple extends Fruit {
    Apple() { name = "apple"; }
    void bake() { System.out.println(name + " pie");}
}
```

(a) 次の文を実行したときの出力を示せ。 Fruit f = new Fruit(); f.squeeze();

(b) 次の文を実行したときの出力を示せ。 Apple a = new Apple(); a.bake();

(c) 次の文を実行したときの出力を示せ。 Orange o = new Orange(); o.squeeze();

(d) 次の文をコンパイル (compile) すると、コンパイルエラー (complie error) になった。理由を簡潔に記せ。Fruit a = new Apple(); a.bake();

(e) クラス Fruit とクラス Orange におけるメソッド (method) squeeze() のように、親クラス (parent class) のメソッドを子クラス (child class) で実装する (implement) ことを何と呼ぶか。

プログラミング(2) [1/3]

問1 C 言語で記述された下記 Program1 に関する以下の問に答えよ。プログラム中の左端の数字は行番号である。qsort は、大きさ n の整数型配列 (array) を整列 (sort) するクイックソートアルゴリズム (quick sort algorithm) を実装した関数である。

Program1

```
1  int part(int a[], int left, int right) {
2      int i, j, pivot, tmp;
3      i = left - 1; j = right; pivot = a[right];
4      for (;;) {
5          while(a[++i] < pivot)
6              ;
7          while(i < --j && pivot < a[j])
8              ;
9          if (i >= j)
10             break;
11         tmp = a[i]; a[i] = a[j]; a[j] = tmp;
12     }
13     tmp = a[i]; a[i] = a[right]; a[right] = tmp;
14     return i;
15 }
16
17 void qsort_1(int a[], int left, int right) {
18     int v;
19     if (left >= [ ① ] )
20         return;
21     v = part(a, left, right);
22     qsort_1(a, left, [ ② ]);
23     qsort_1(a, [ ③ ], right);
24 }
25
26 void qsort(int a[], int n) {
27     qsort_1(a, 0, n-1);
28 }
```

(a) Program1 中の空欄 ① ～ ③ を埋めて、整列を行う関数を完成せよ。

(b) 安定 (stable) な整列アルゴリズムとはどのようなものか説明し、クイックソートが安定な整列かどうか答えよ。

(c) qsort の時間計算量 (time complexity) のオーダ (order) を、平均的な場合 (average case) と最悪の場合 (worst case) それぞれについて答えよ。

(d) 時間計算量が最悪となる場合を避ける工夫として、プログラム 3 行目の変数 pivot の値設定を変更することが考えられる。どのように変更すれば最悪の場合を回避できる可能性が高くなるか答えよ。

[次ページに続く]

プログラミング(2)　[2/3]

問2　C 言語で記述された下記 Program2 に関する以下の問に答えよ。プログラム中の左端の数字は行番号である。Program2 は、動的計画法 (dynamic programming) を用いてナップザック問題 (knapsack problem) の解を求めるプログラムである。ナップザック問題とは、N 種類の品物 (items) があり、それぞれの大きさ (size) と価値 (value) が与えられているとき、品物の大きさの合計がナップザックの大きさ K 以下で、価値の合計が最大になるような品物の組み合わせとその価値を求めるものである。品物の数は整数で、それぞれの品物をいくつ選んでもよい。

Program2

```
 1  #include <stdio.h>
 2  int size[] = {2, 3, 5};
 3  int value[] = {3, 5, 7};
 4  #define N (sizeof(size) / sizeof(size[0]))
 5  #define K 8                                  // size of knapsack
 6
 7  int main() {
 8    int i, j, total[K+1], choice[K+1], repack;
 9    for (i=0; i<=K; i++) {
10      total[i] = 0;
11      choice[i] = -1;
12    }
13    for (i=0; i<N; i++) {
14      for (j=size[i]; j<=K; j++) {
15        repack = total[j-size[i]] + value[i];
16        if (repack > total[j]) {
17          total[j] = repack;
18          choice[j] = i;
19        }
20      }
21      printf("i = %d (size=%d)¥n", i, size[i]);
22      printf(" size  = ");
23      for (j=0; j<=K; j++)
24        printf("%3d", j);
25      printf("¥n total  = ");
26      for (j=0; j<=K; j++)
27        printf("%3d", total[j]);
28      printf("¥n choice = ");
29      for (j=0; j<=K; j++)
30        printf("%3d", choice[j]);
31      printf("¥n");
32    }
33    for ( [ ① ]; choice[i]>=0; [ ② ] )
34      printf("Pack item %d (value %d)¥n", choice[i], value[choice[i]]);
35    printf("sum=%d¥n", total[ [ ③ ] ]);
36    return 0;
37  }
```

[次ページに続く]

プログラミング(2)　[3/3]

解を求める手順の概略は、以下のようになっている。

1. まず一番小さい品物に着目する。この品物の大きさから大きさ K までのナップザックそれぞれに対して、この品物のみを使ったときの最大価値を求める。このとき、最後に入れた品物としてこの品物の番号も記録しておく。
2. 次に、二番目に小さい品物に着目する。1.で得られた結果を利用しながら、この品物の大きさから大きさ K までのナップザックそれぞれに対して、ここまでに利用可能な品物を使った組み合わせの最大価値を求め、最後に入れた品物の番号も記録しておく。
3. 2.の手順で使用する品物を順に増やしてゆき、最後の品物を用いて求めた大きさ K のナップザックの最大価値が、品物の組み合わせの最大価値である。品物の組み合わせを求める際は、最後に入れた品物を取り出し、その品物の大きさの分だけナップザックを小さくするということを、取り出す品物がなくなるまで繰り返す。

以下は、Program2 を動かしたときの、32 行目までの出力結果である。

```
i = 0 (size=2)
 size   =   0  1  2  3  4  5  6  7  8
 total  =   0  0  3  3  6  6  9  9 12
 choice = -1 -1  0  0  0  0  0  0  0
i = 1 (size=3)
 size   =   0  1  2  3  4  5  6  7  8
 total  =   0  0  3  5  6  8 10 11 13
 choice = -1 -1  0  1  0  1  1  1  1
i = 2 (size=5)
 size   =   0  1  2  3  4  5  6  7  8
 total  =   0  0  3  5  6  8 10 11 13
 choice = -1 -1  0  1  0  1  1  1  1
```

(a) Program2 中の空欄 ① ～ ③ を埋めて、プログラムを完成せよ。

(b) Program2 の 33 行目以降の出力結果を示せ。

(c) 問題文中の記号を使って、ナップザック問題の時間計算量のオーダを求めよ。