令 和 5 年 度

名古屋大学大学院情報学研究科
知能システム学専攻
入学試験問題（専門）

令和４年８月８日

注意事項

１．試験開始の合図があるまでは、この問題冊子を開いてはならない。

２．試験終了まで退出できない。

３．外国人留学生の志願者は、日本語と日本語以外の１言語間の辞書１冊に限り使用してよい。電子辞書の持ち込みは認めない。

４．日本語または英語で解答すること。

５．問題冊子、解答用紙４枚、草稿用紙３枚が配布されていることを確認すること。

６．問題は解析・線形代数、確率・統計、プログラミングの３科目がある。これらの全てについて回答すること（プログラミングのみ、大問１問ごとに１枚の計２枚で、他の２科目は１枚で解答すること）。なお、解答した科目名を解答用紙の指定欄に記入すること。プログラミングには大問[1]か[2]かを明記すること。

７．全ての解答用紙の所定の欄に受験番号を必ず記入すること。解答用紙に受験者の氏名を記入してはならない。

８．解答用紙に書ききれない場合は、裏面を使用してもよい。ただし、裏面を使用した場合は、その旨、解答用紙表面右下に明記すること。

９．解答用紙は試験終了後に４枚とも提出すること。

１０．問題冊子、草稿用紙は試験終了後に持ち帰ること。

解析・線形代数

(解の導出過程も書くこと)

[1] 行列 $\boldsymbol{M} = \begin{bmatrix} 8 & 1 & -6 \\ 1 & 8 & -6 \\ 3 & 3 & -3 \end{bmatrix}$ について，次の問いに答えよ．

(a) $\boldsymbol{M}$の固有値をすべて求めよ．また，各固有値に対応する単位固有ベクトルをそれぞれ求めよ．

(b) $\boldsymbol{M}$の異なる二つの単位固有ベクトルを基底とする平面をPとする．Pの単位法線ベクトル $\boldsymbol{n}$を求めよ．

(c) $\boldsymbol{n}$を用いてPの方程式 を求めよ．

(d) P上の3点A$(0, a, z_a)$，B$(b, 0, z_b)$，C$(0, 0, z_c)$を 頂点 とする三角形の面積をa，bで表せ．

[2] 複素数 $z = e^{i(a+\frac{b}{10}i)x}$ について，次の問いに答えよ．ただし，i は虚数単位である．

(a) z の実部 $\mathrm{Re}(z)$ と虚部 $\mathrm{Im}(z)$ をそれぞれ示せ．

(b) $a = 0$，$b = 1$のとき，$\mathrm{Re}(z)$ のグラフの概形を描け．ただし，横軸を x とし，$0 \leq x \leq 20$ とする．

(c) $a = 1$，$b = 1$のとき，$\mathrm{Im}(z)$ のグラフの概形を描け．ただし，横軸を x とし，$0 \leq x \leq 20$ とする．

[3] 関数 $f(x, y) = (x + y)e^{-x^2-y^2}$ について，次の問いに答えよ．

(a) f の停留点をすべて求めよ．

(b) (a) で求めた停留点ごとにヘッセ行列$\boldsymbol{H} = \begin{bmatrix} \dfrac{\partial^2 f(x,y)}{\partial x^2} & \dfrac{\partial^2 f(x,y)}{\partial x \partial y} \\ \dfrac{\partial^2 f(x,y)}{\partial x \partial y} & \dfrac{\partial^2 f(x,y)}{\partial y^2} \end{bmatrix}$とその固有値を求めよ．

(c) (b) で求めた固有値を用いて，(a) で求めた停留点がそれぞれ極大点，極小点，鞍点のいずれかであるかを示せ．

Translation of technical terms

行列	matrix	固有値	eigenvalue
単位固有ベクトル	unit eigenvector	基底	basis
平面	plane	単位法線ベクトル	unit normal vector
方程式	equation	頂点	vertex
三角形	triangle	面積	area
複素数	complex number	虚数単位	imaginary unit
実部	real part	虚部	imaginary part
グラフ	graph	概形	approximate shape
横軸	horizontal axis	関数	function
停留点	stationary point	ヘッセ行列	Hessian matrix
極大点	local maximum point	極小点	local minimum point
鞍点	saddle point		

確率・統計

解の導出過程も書くこと．

[1] 以下の問いに答えよ．サイコロ (dice) は 1 から 6 の数字が同じ確率で出るものとする．

(1) サイコロを 3 回振り，出た数字の積を得点とするとき，得点の期待値を求めよ．

(2) サイコロを 2 回振り，異なる数字が出た場合はそれらの積を，それ以外の場合は 0 を得点とするとき，得点の期待値を求めよ．

(3) サイコロを 3 回振り，すべて異なる数字が出た場合はそれらの積を，それ以外の場合は 0 を得点とするとき，得点が 15 点以上となる確率を求めよ．

[2] 確率変数X の確率密度関数$f(x)$ に関して，以下の問いに答えよ．

(1) $f(x)$ が満たすべき性質を 2 つ答えよ．

(2) $f(x)$ が次式で与えられる連続関数であるとき，定数a，b，c の値を求めよ．

$$f(x) = \begin{cases} a^2x^3 + bx^2 + 4ax + c & (0 \leq x \leq 1) \\ 0 & (\text{otherwise}) \end{cases}$$

(3) $f(x)$ が (2) と同様に与えられるとき，確率変数 X の期待値 $E(X)$ を求めよ．

[3] 統計的仮説検定に関して，以下の問いに答えよ．

(1) 帰無仮説が実際には真であるのに棄却してしまう過誤のことを第一種過誤という．有意水準0.05 で検定した場合に第一種過誤が発生する確率を答えよ．

(2) 既存手法に対し 3 つの提案手法がある場合を考える．各提案手法と既存手法の実験結果の間に有意な差があるかを有意水準 0.05 で検定した場合に，計 3 回の検定のうち，少なくとも 1 回において第一種過誤が発生する確率を有効数字2桁で答えよ．

Translation of technical terms

- 確率: probability
- 積: product
- 期待値: expectation
- 確率変数: random variable
- 確率密度関数: probability density function
- 連続関数: continuous function
- 定数: constant
- 統計的仮説検定: statistical hypothesis testing
- 帰無仮説: null hypothesis
- 真: true
- 棄却: rejection
- 第一種過誤: type I error
- 有意水準: significance level
- 有効数字2桁: 2 significant digits

プログラミング

[1] 以下の全ての問いに答えよ．ただし，コンパイル時の最適化は行われないものとし，オーバーフローエラーが起こらない範囲での実行のみを考えることとする．

(1) ソースコード 1 に掲げた C 言語のプログラム `binsearch.c` について，以下の全ての問いに答えよ．

(a) `binsearch(x,i,j)` が「昇順にソートされた配列 `a[i],...,a[j]` に対し，値 `x` の存在を二分探索によって判定する」関数になるように，ソースコード 1 の A ～ C を埋めよ．

(b) `binsearch_loop` が，`binsearch` と同じ二分探索を再帰呼出しを用いずに記述したものになるように，ソースコード 1 の D ～ F を埋めよ．

(2) ソースコード 2 に掲げた C 言語のプログラム `function.c` について，以下の全ての問いに答えよ．

(a) `main()` を実行した時，標準出力に出力される実行結果を書け．

(b) 任意の非負整数 `n` に対して，`f2(n,0,1)` の返り値が，`f1(n)` の返り値と常に同じになるように，ソースコード 2 の G を埋めよ．

(c) 以下の回数をそれぞれ答えよ．

i. `main()` を実行したとき，`f1` が呼び出される回数．

ii. 137 行目の `f1(7)` を，`f2(7,0,1)` に置き換えて `main()` を実行したとき，`f2` が呼び出される回数．

(d) `f_loop` が，`f2` と同じ結果を返す関数を再帰呼出しを用いずに記述したものになるように，ソースコード 2 の H ～ K を埋めよ．

(3) `binsearch` や `f1`，`f2` のような再帰呼出しを含むプログラムよりも，`binsearch_loop` や `f_loop` のように再帰呼出しを用いない方が，多くの場合，実行時間が短くなる．この理由を，50 字以内（英語の場合，30 words 以内）で説明せよ．

Translation of technical terms

コンパイル	compile	二分探索	binary search
最適化	optimization	関数	function
オーバーフローエラー	overflow error	再帰呼出し	recursive call
昇順	ascending order	標準出力	standard output
ソート	sort	非負整数	non-negative integer
配列	array	返り値	return value

ソースコード 1: binsearch.c

```
1  #include <stdio.h>
2  #define TRUE 1
3  #define FALSE 0
4
5  int a[10] = { 1, 4, 6, 10, 11, 13, 15, 20, 30, 32};
6
7  int binsearch (int x, int i, int j) {
8    int k;
9    if(i>j)
10     return FALSE;
11   else {
12     k=(i+j)/2;
13     if ( [ A ] )
14       return binsearch(x,k+1,j);
15     else if ( [ B ] )
16       return binsearch( [ C ] );
17     else
18       return TRUE;
19   }
20 }
21
22 int binsearch_loop (int x, int i, int j) {
23   int k;
24   while(1) {
25     if(i>j) {
26       return FALSE;
27     } else {
28       k=(i+j)/2;
29       if ( [ D ] )
30         i=k+1;
31       else if ( [ E ] )
32         j= [ F ] ;
33       else {
34         return TRUE;
35       }
36     }
37   }
38 }
39
40 int main (void) {
41   if(binsearch(14,0,9)) printf("found\n");
42   else printf("not found\n");
43
44   if(binsearch_loop(14,0,9)) printf("found\n");
45   else printf("not found\n");
46
47   return 0;
48 }
```

ソースコード 2: function.c

```
100 #include <stdio.h>
101
102 int f1 (int x) {
103   if (x <= 0)
104     return 0;
105   else if (x == 1)
106     return 1;
107   else
108     return f1(x-2) + f1(x-1);
109 }
110
111 int f2 (int x, int y, int z) {
112   if (x <= 0)
113     return y;
114   else if (x == 1)
115     return z;
116   else
117     return f2(x-1, z, [ G ]);
118 }
119
120 int f_loop (int x, int y, int z) {
121   int tmp;
122   while (1) {
123     if (x<=0) {
124       return y;
125     } else if (x == 1) {
126       return [ H ] ;
127     } else {
128       tmp = y;
129       x = [ I ] ;
130       y = [ J ] ;
131       z = [ K ] ;
132     }
133   }
134 }
135
136 int main (void) {
137   printf("%d\n", f1(7));
138
139   return 0;
140 }
```

[2] 5ページ目のソースコード3に掲げたC言語のプログラム repdec.c 中で定義された関数 repdec(n, d, base) は，正の整数 n，d が与えられたとき，除算 n/d の結果を表示するプログラムである．ただし，結果が循環小数になる場合は，繰り返される数字列の部分を{ }で表示する．ここで，base は N 進法の基数 N を指定する引数であり，base = 10 のとき 10 進法で出力される．また，d の値はプログラム3行目の MAX 以上にならないものとする．

37行目から40行目までの記述は，プログラムの動作確認のために，変数 p の内容を表示するものである．

このプログラムを実行した結果は以下の通りである．

```
0.{3}
p = [0 1 0 ]
```

このとき以下の問いに答えよ．

1. 45行目の引数を (22, 7, 10) に変更したときの実行結果を書け．

2. 45行目の引数を (1, 10, 10) に変更したときの実行結果は以下の通りである．

   ```
   0.1
   p = [2 1 0 0 0 0 0 0 0 0 ]
   ```

   この引数を (1, 10, 2) に変更して実行した場合は，同じ 1/10 の結果を2進法で出力することになる．その結果を書け．

3. 変数 p[n] の値が k のとき，それは何を示しているのか，50字以内（英語の場合，30 words 以内）で説明せよ．

4. 22行目はループからの脱出条件を示している．なぜこの条件で良いのか，その理由を100字以内（英語の場合，60 words 以内）で説明せよ．

5. 6ページ目のソースコード4に掲げたC言語のプログラム sum.c は，0.1 を100回足した結果を出力するプログラムである．しかし，その出力結果は10にならない．その理由を50字以内（英語の場合，30 words 以内）で説明せよ．

Translation of technical terms

定義する	define	引数	argument
関数	function	10 進法	decimal numeral system, base-ten numeral system
正の整数	positive integer		
除算	division	変数	variable
循環小数	repeating decimal	2 進法	binary numeral system, base-two numeral system
数字列	digit sequence		
N 進法	base-N numeral system	ループ	loop
基数	base	脱出条件	exit condition

ソースコード 3: repdec.c

```c
#include <stdio.h>
#include <stdlib.h>
#define MAX 1000

void repdec(unsigned int n, unsigned int d, unsigned int base)
{
  unsigned int i, k;
  unsigned int a[MAX + 1], p[MAX];

  for (i = 0; i < MAX; i++)
    p[i] = 0;

  a[0] = n / d;
  n = n % d;
  k = 0;

  while (1){
    p[n] = ++k;
    n = n * base;
    a[k] = n / d;
    n = n % d;
    if (p[n] != 0)
      break;
  }

  printf("%u.", a[0]);
  for (i = 1; i < p[n]; i++)
    printf("%u", a[i]);
  if (p[n] < k || a[k] != 0){
    printf("{");
    for (i = p[n]; i<= k; i++)
      printf("%u", a[i]);
    printf("}");
  }
  printf("\n");

  printf("p = [");
  for (i = 0; i < d; i++)
    printf("%u ", p[i]);
  printf("]\n");
}

int main (void)
{
  repdec(1, 3, 10);
  return 0;
}
```

ソースコード 4: sum.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main (void)
5  {
6    float f = 0.1, sum = 0;
7    unsigned int i;
8
9    for (i = 0; i < 100; i++)
10     sum += f;
11   printf("%f\n", sum);
12
13   return 0;
14 }
```