

令和2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）

令和元年8月7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. 外国人の留学生に限り、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3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解答すること。なお、選択した科目名を解答用紙の指定欄に記入すること。
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3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] 複素数 $z = (1+i)^n$ (n は実数, i は虚数単位) について, $z = u+vi$ と表すことにする. このとき, 以下の問いに答えよ.

(a) $n = 8$ のとき, u, v を求めよ.

(b) $|z| = 8$ のとき, u, v を求めよ.

[2] 次の二つの行列 P, Q について, 以下の問いに答えよ.

$$P = \frac{1}{5} \begin{pmatrix} 7 & 4 \\ 4 & 13 \end{pmatrix}, \quad Q = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

(a) P の固有値と単位固有ベクトルを求めよ.

(b) $q = (x, y)^t$ とするとき,

$$q^t P q < 1 < q^t Q q$$

を満たす座標平面上の点 (x, y) の領域を図示せよ. ただし, q^t は q の転置を表す.

[3] 座標平面上の動点 P の時刻 t における座標が次の式で与えられているとき, 以下の問いに答えよ.

$$\begin{cases} x = \sin^3 t \\ y = \cos^3 t \end{cases}$$

(a) P の軌跡の長さを求めよ. ただし, $0 \leq t \leq 2\pi$ とする.

(b) P の速度の大きさが最大となるときの P の座標を求めよ. ただし, $0 < t < \frac{\pi}{2}$ とする.

Translation of technical terms

複素数 complex number, 実数 real number, 虚数単位 imaginary unit, 行列 matrix, 固有値 eigenvalue, 単位固有ベクトル unit eigenvector, 座標平面 coordinate plane, 領域 region, 転置 transpose, 動点 moving point, 座標 coordinates, 軌跡 trajectory, 速度 velocity, 最大 maximum.

確率・統計

解の導出過程も書くこと。

[1] 表 (head) が出る確率と裏 (tail) が出る確率がともに $1/2$ であるコイン (coin) 1 枚を連続して投げ、『裏表表』、『表表表』のような事前に決めた系列 (sequence) が出現したら終了するものとする。このとき、以下の問いに答えよ。

- (1) 『裏表表』という系列に決めた場合に 3 回で終了する確率、すなわち、1 回目が裏、2 回目が表、3 回目が表となる確率を求めよ。
- (2) 『裏表表』という系列に決めた場合にちょうど 4 回で終了する確率を求めよ。
- (3) 『表表表』という系列に決めた場合にちょうど 4 回で終了する確率を求めよ。
- (4) 初めて表が出るまでにコインを投げる回数の期待値を $E(\text{表})$ と書くとき $E(\text{表})$ を求めよ。
- (5) 初めて 2 回続けて表が出るまでにコインを投げる回数の期待値を $E(\text{表表})$ と書くとき $E(\text{表表})$ を求めよ。なお、初めて表が出た直後に表が出た場合はその時点で 2 回続けて表が出たことになり、裏が出た場合はその後 2 回続けて表が出るまでにコインを投げる回数の期待値が $E(\text{表表})$ と等しくなることから以下の式が成立する。

$$E(\text{表表}) = \frac{E(\text{表}) + 1}{2} + \frac{E(\text{表}) + 1 + E(\text{表表})}{2}$$

- (6) 『表表表』という系列に決めた場合のコインを投げる回数の期待値を求めよ。

[2] 確率変数 X, Y の同時確率密度関数 $f_{X,Y}(x, y)$ が次式で与えられている。ただし、 a は定数である。このとき、以下の問いに答えよ。

$$f_{X,Y}(x, y) = \begin{cases} a(x^2 - y^2)e^{-x} & (0 \leq x, -x \leq y \leq x) \\ 0 & (\text{otherwise}) \end{cases}$$

- (1) $f_{X,Y}(x, y)$ が最大となる x, y の値を求めよ。
- (2) 周辺密度関数 $f_X(x)$ を定数 a を用いて表せ。
- (3) 定数 a の値を求めよ。
- (4) 確率変数 X の期待値 μ_x を求めよ。

Translation of technical terms

- | | |
|-------------------------|---|
| ● 確率: probability | ● 期待値: expectation |
| ● 確率変数: random variable | ● 同時確率密度関数: joint probability density function |
| ● 定数: constant | ● 周辺密度関数: marginal probability density function |

プログラミング

プログラム A は、ハッシュ表を用いて正整数の集合を操作するための C 言語プログラムである。ハッシュ値ごとの正整数のリスト構造を、3つの配列 element, next, hfirst を用いて実現している。element 配列は、正整数の集合を格納するための配列である。next 配列は、リストの次の要素が格納されている element 配列の添え字を格納するための配列である。hfirst 配列はハッシュ表のための配列であり、hfirst[h] はハッシュ値 h を持つ正整数のリストの先頭の要素が格納されている element 配列の添え字を表す。ここで、hfirst[h] が -1 のときは、ハッシュ値 h を持つ正整数のリストが空であることを表す。

hashfunc 関数は、ハッシュ関数を表す。search 関数、insert 関数、delete 関数は、それぞれ element 配列に対する正整数の探索、挿入、削除を行う関数である。initarrays 関数は、hfirst 配列、element 配列、next 配列をそれぞれ初期化する関数である。outputarrays 関数は、hfirst 配列、element 配列、next 配列をそれぞれ標準出力に出力する関数である。

プログラム A について、以下の問いに答えよ。

- (1) プログラム A を実行した際に、74 行目の outputarrays 関数の呼び出しにより標準出力に出力される文字列を書け。
- (2) プログラム A の 65 行目の `int data[] = {1,2,3,5,7};` を `int data[] = {1,2,18,19,20};` に置き換えて実行した際に、74 行目の outputarrays 関数の呼び出しにより標準出力に出力される文字列を書け。
- (3) data 配列に格納される正整数の集合がどのような特徴を持つときに、search 関数の実行時間が集合のサイズに対して長くなるか答えよ。
- (4) (ア) と (イ) を適切な式で埋めて、delete 関数を完成させよ。
- (5) プログラム A を実行した際に、78 行目の outputarrays 関数の呼び出しにより標準出力に出力される文字列を書け。ただし、プログラム A は (2) の置き換えを行っていないものとする。

Translation of technical terms:

プログラム	program	関数	function
ハッシュ表	hash table	ハッシュ関数	hash function
正整数	positive integer	探索	search
集合	set	挿入	insertion
C 言語	C programming language	削除	deletion
ハッシュ値	hash value	初期化	initialization
リスト	list	標準出力	standard output
構造	structure	出力	output
配列	array	実行	execution
格納する	store	呼び出し	call
要素	element	文字列	character string
添え字	index	実行時間	execution time
空	empty	式	expression

プログラム A

```

1 #include <stdio.h>
2 #define MAXSIZE          1000
3 #define P                17
4 #define SENTINEL        -1
5 #define NOTFOUND        -2
6 int hfirst[P];
7 int element[MAXSIZE];
8 int next[MAXSIZE];
9 int avail=-1;
10 int maxnode=0;
11 int hashfunc(int data){
12     return data%P;
13 }
14 int search(int h, int data){
15     int pred=-1;
16     if (hfirst[h]==-1) return NOTFOUND;
17     if (element[hfirst[h]]==data) return pred;
18     pred=hfirst[h];
19     while (next[pred]!=SENTINEL){
20         if (element[next[pred]]==data) return pred;
21         pred=next[pred];
22     }
23     return NOTFOUND;
24 }
25 void insert(int h, int data){
26     int u;
27     if (avail!=-1) {
28         u=avail;
29         avail=next[avail];
30     } else {
31         u=maxnode;
32         maxnode=maxnode+1;
33     }
34     element[u]=data;
35     next[u]=hfirst[h];
36     hfirst[h]=u;
37 }
38 void delete(int h, int pred){
39     int u;
40     if (pred!=-1){
41         u=next[pred];
42         next[pred]= (ア) ;
43     } else {
44         u=hfirst[h];

```

```

45     hfirst[h]= (1);
46     }
47     next[u]=avail; avail=u;
48 }
49 void initarrays(){
50     int i;
51     for (i=0; i<P; i++) {hfirst[i]=-1;}
52     for (i=0; i<MAXSIZE; i++) {element[i]=0;}
53     for (i=0; i<MAXSIZE; i++) {next[i]=SENTINEL;}
54 }
55 void outputarrays(int maxnode){
56     int i;
57     for (i=0; i<P; i++) {printf("%d,",hfirst[i]);}
58     printf("\n");
59     for (i=0; i<maxnode; i++) {printf("%d,",element[i]);}
60     printf("\n");
61     for (i=0; i<maxnode; i++) {printf("%d,",next[i]);}
62     printf("\n");
63 }
64 int main(){
65     int data[] = {1,2,3,5,7};
66     int h;
67     int i;
68     int pred;
69     initarrays();
70     for (i=0; i<5; i++){
71         h=hashfunc(data[i]);
72         if (search(h,data[i])==NOTFOUND) insert(h,data[i]);
73     }
74     outputarrays(maxnode);
75     h=hashfunc(1);
76     pred=search(h,1);
77     if (pred!=NOTFOUND) delete(h,pred);
78     outputarrays(maxnode);
79     return 0;
80 }

```