

平成31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）

平成30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. 外国人留学生は、語学辞書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. 外国人留学生は、英語での解答を可とする。
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、解答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] a_i ($i = 1, 2, \dots, N$) を定数とするとき、次の関数 $f(x)$ の値が最大になるときの x の値を求めよ。

$$f(x) = \prod_{i=1}^N e^{\frac{-(x-a_i)^2}{2}}$$

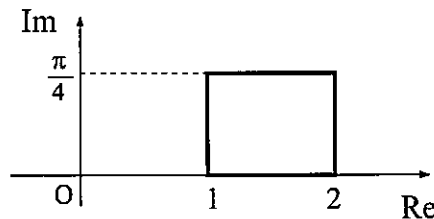
- [2] 次の行列 P について、以下の問いに答えよ。

$$P = \begin{pmatrix} 5 & 2 \\ 2 & 2 \end{pmatrix}$$

- (a) 行列 P の固有値と単位固有ベクトルを求めよ。
(b) 行列 P を対角化せよ。
(c) $q = (x, y)^t$ とするとき、 $q^t P q = 1$ を満たす座標平面上の点 (x, y) の軌跡を描け。ただし、 q^t は q の転置を表す。

- [3] 複素数 z について、以下の問いに答えよ。ただし、 i は虚数単位を表す。

- (a) $z = e^{\frac{\pi}{3}i}$ とし、 $n = 0, 1, 2, \dots, 5$ とするとき、複素平面上での z^n の座標をすべて求めよ。
(b) 複素関数 $w = e^z$ を考える。下図のように z が複素平面上で、4点 $(1, 0)$, $(1, \frac{\pi}{4})$, $(2, \frac{\pi}{4})$, $(2, 0)$ を頂点とする四角形上を移動したとき、複素平面上での w の軌跡を描け。



Translation of technical terms

定数 constant, 関数 function, 最大 maximum, 行列 matrix, 固有値 eigenvalue,

単位固有ベクトル unit eigenvector, 対角化 diagonalization, 座標平面 coordinate plane, 軌跡 trajectory,

転置 transpose, 複素数 complex number, 虚数単位 imaginary unit, 複素平面 complex plane,

座標 coordinates, 複素関数 complex function, 頂点 vertex, 四角形 rectangle.

確率・統計

解の導出過程も書くこと。

[1] 赤玉が4つ、青玉が1つ、白玉が1つ入った箱 A と、赤玉が2つ、青玉が2つ、白玉が2つ入った箱 B があるとき、以下の問いに答えよ。ただし2つの箱は外からは区別が付かず、玉を取り出すとき、箱の中は見えないものとする。

- (1) 無作為に1つの箱を選び、その箱から玉を1つ取り出したとき、それが赤玉である確率を求めよ。
- (2) 無作為に1つの箱を選び、その箱から玉を2つ取り出したとき、それらの色が異なる確率を求めよ。
- (3) 1つの箱から玉を1つ取り出したとき、それが赤玉であったとする。その玉を箱に戻さずに、さらに玉を1つ取り出すとき、最初に玉を取り出したのと同じ箱から取り出す場合と、別の箱から取り出す場合、それぞれについて新たに取り出した玉が赤玉である確率を求め、どちらの場合の方が赤玉である確率が大きいか答えよ。

[2] 確率変数 X の確率密度関数 $f(x)$ が次式で与えられている。ただし a は定数とする。

$$f(x) = \begin{cases} 12x^2(a-x) & (0 \leq x \leq 1) \\ 0 & (\text{上記以外}) \end{cases}$$

- (1) a の値を求めよ。
- (2) 確率変数 X の値が $1/3$ 以下となる確率を求めよ。
- (3) 確率変数 X の平均 μ と分散 σ^2 を求めよ。
- (4) 表が出る確率 θ の確率分布が $f(\theta)$ で与えられたコインがあるとする。このコインを n 回投げたときにすべて裏が出る確率の期待値を求めよ。ただし、 n 回の試行の間、 θ の値は変動しないものとする。

Translation of technical terms

- | | |
|--|-------------------------|
| ● 無作為: random | ● 確率変数: random variable |
| ● 確率密度関数: probability density function | ● 定数: constant |
| ● 平均: mean | ● 分散: variance |
| ● 確率分布: probability distribution | ● 期待値: expectation |

プログラミング

プログラム P は与えられた文字列を操作する C 言語プログラムである。プログラム P で扱う文字は、1 バイト、または、3 バイトで構成される。1 バイトで構成される文字は最上位ビットが 0 であり、残りのビットにより文字を指定する。3 バイトで構成される文字は、その 1 バイト目の最上位ビットが 1 であり、1 バイト目の残りのビットと 2 バイト目、ならびに、3 バイト目により文字を指定する。文字列は 8 ビット長の char 型の配列に格納され、'¥0' を終端とする。16 進数は、0x41 のように先頭に 0x をつけて表す。すなわち、0x41 は 10 進表記の 65 の 16 進表記である。

プログラム P において、27 行目の関数 concat は、ポインタ s1、ポインタ s2 がこの順で引数として与えられたとき、s1 と s2 が参照する文字列をこの順に連結し、得られた文字列の先頭を参照するポインタを返す。59 行目の関数 replace は、ポインタ s1、ポインタ s2、正の整数 k がこの順で引数として与えられたとき、s1 が参照する文字列の k 文字目を s2 が参照する文字列で置き換えて得られる文字列の先頭を参照するポインタを返す。なお、関数 replace の引数 s2 が参照する文字列の文字数は 1 とし、1 文字は 1 バイトまたは 3 バイトで構成されることに注意せよ。

このとき以下の問いに答えよ。なお、ポインタが参照する文字列の内容を解答する場合、下の記述例のようにアドレスと値を対応づけて文字列を表すこと（終端も含めること）。例えば、80 行目の $r = \text{str1}$; の実行直後のポインタ r が参照する文字列の内容は、下の記述例のように解答する。

アドレス	r	r+1	r+2	r+3
値	0x41	0x42	0x43	'¥0'

- (1) 空欄 A, B, C, D, E にあてはまる式を書いて、関数 concat を完成させよ。
- (2) 76 行目の $r = \text{index}(\text{str1}, 1)$; を実行した直後の r が参照する文字列の内容を記述例にならって書け。
- (3) 77 行目の $r = \text{index}(\text{str2}, 2)$; を実行した直後の r が参照する文字列の内容を記述例にならって書け。
- (4) 78 行目の $r = \text{index}(\text{str3}, 3)$; を実行した直後の r が参照する文字列の内容を記述例にならって書け。
- (5) 79 行目の $r = \text{replace}(\text{str3}, \text{str4}, 3)$; の実行時に呼び出される 62 行目の $\text{tmp2} = \text{concat}_p(\text{string1}, \text{string2}, \text{tmp1})$; を実行した直後の tmp2 の内容を記述例にならって書け。
- (6) 48 行目では下線部のように固定長の領域を result に割り当てている。しかしながら、関数 concat_p に与えられる引数によっては領域が不足する。与えられる引数に応じて必要な領域が割り当てられるように下線部を適切な式に変更せよ。
- (7) 空欄 F, G, H にあてはまる式を書いて関数 replace を完成させよ。

プログラム P

```
1  #include <stdlib.h>
2
3  char* index(char* string, int p){
4      char *tmp;
5      int i;
6      i = 1;
7      tmp = string;
8
9      while ( (*tmp != '\0') && (i < p) ){
10         i++;
11         if ((*tmp & 0x80) == 0)
12             tmp++;
13         else
14             tmp += 3;
15     }
16     return tmp;
17 }
18
19 int length_b(char* string){
20     int len;
21     len = 0;
22     while (string[len] != '\0')
23         len++;
24     return len;
25 }
26
27 char* concat(char* string1, char* string2){
28     char *result, *tmp;
29     result = (char*)malloc(sizeof(char) * (length_b(string1) + length_b(string2) + 1));
30     tmp = result;
31     while (*string1 != '\0'){
32         A = B;
33         tmp++;
34         string1++;
35     }
36     while (*string2 != '\0'){
37         C = D;
```

```

38     tmp++;
39     string2++;
40 }
41 *tmp = ;
42 return result;
43 }
44
45 char* concat_p(char* string1, char* string2, char* l){
46     char *result, *tmp1, *tmp2;
47     tmp1 = string1;
48     result = (char*)malloc(sizeof(char) * 100 );
49     tmp2 = result;
50     while ((tmp1 != 1) && (*tmp1 != '\0')){
51         *tmp2 = *tmp1;
52         tmp1++;
53         tmp2++;
54     }
55     *tmp2 = '\0';
56     return concat(result, string2);
57 }
58
59 char* replace(char* string1, char* string2, int p){
60     char *tmp1, *tmp2;
61     tmp1 = index(string1, p);
62     tmp2 = concat_p(string1, string2, tmp1);
63     if (  == 0)
64         ;
65     else
66         ;
67     return concat(tmp2, tmp1);
68 }
69
70 void main(){
71     char str1[4] = {0x41, 0x42, 0x43, '\0'};
72     char str2[10] = {0xE3, 0x81, 0x82, 0xE3, 0x81, 0x84, 0xE3, 0x81, 0x86, '\0'};
73     char str3[9] = {0xE3, 0x81, 0x82, 0x41, 0xE3, 0x81, 0x84, 0x42, '\0'};
74     char str4[4] = {0xE3, 0x81, 0x86, '\0'};
75     char *r;

```

```

76     r = index(str1, 1);
77     r = index(str2, 2);
78     r = index(str3, 3);
79     r = replace(str3, str4, 3);
80     r = str1;
81 }

```

Translation of technical terms

プログラム	program	関数	function
文字列	string	ポインタ	pointer
C 言語	C programming language	引数	argument
バイト	byte	参照する	refer
最上位ビット	most significant bit	アドレス	address
配列	array	式	expression
16 進数	hexadecimal	固定長	fixed length
10 進表記	decimal notation	割り当てる	allocate
16 進表記	hexadecimal notation		