

平成30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29年8月3日(木)
12:30~15:30

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. 試験終了まで退出してはならない。
3. 英語で解答してもよい。外国人留学生は、和英辞書などの辞書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. 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。
5. 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。
6. 解答用紙の指定欄に受験番号を必ず記入すること。なお、解答用紙に受験者の氏名を記入してはならない。
7. 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、その旨を解答用紙表面右下に明記すること。
8. 解答用紙は試験終了後に3枚とも提出すること。
9. 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] 次の関数 f について、以下の問いに答えよ。

$$f(x, y) = \sin x + \sin y + \sin(x + y) \quad (0 < x < \pi, 0 < y < \pi)$$

- (a) f の停留点を求めよ。
 (b) (a) で求めた停留点に対して極値をとるかどうかを判定し、 f の極値を求めよ。

[2] 以下の問いに答えよ。ただし、 $\text{Im}(z)$ は複素数 z の虚部を表す。

(a) z 平面上の直線 $\text{Im}(z) = \frac{1}{2}$ が複素関数 $w = \frac{1}{z}$ によって写される w 平面上の図形を求め、図示せよ。

(b) z 平面上の領域 $\text{Im}(z) > 0$ が 1 次分数変換 $w = \frac{\alpha z + \beta}{z + \gamma}$ によって w 平面上の領域 $|w| < 1$ に写されるとき、複素数 α, β, γ を求めよ。

[3] 次の対称行列 A について、以下の問いに答えよ。

$$A = \begin{pmatrix} 5 & -3 \\ -3 & 5 \end{pmatrix}$$

- (a) A のすべての固有値および各固有値に対する単位固有ベクトルを求めよ。
 (b) A を対角化する直交行列 $P = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ のうち $a = d$ のものを求め、 A を対角化せよ。
 (c) ある 1 次変換 $\begin{pmatrix} x \\ y \end{pmatrix} = U \begin{pmatrix} x' \\ y' \end{pmatrix}$ により、2 次形式 $Q(x, y) = 5x^2 - 6xy + 5y^2$ を $x'y'$ の項を含まない 2 次形式に変形せよ。
 (d) 2 次曲線 $5x^2 - 6xy + 5y^2 = 8$ の概形を図示せよ。

Translation of technical terms

関数	function	停留点	stationary point
極値	extreme value	複素数	complex number
虚部	imaginary part	平面	plane
直線	line	複素関数	complex function
図形	shape	領域	region
1 次分数変換	linear fractional transformation	対称行列	symmetric matrix
固有値	eigenvalue	単位固有ベクトル	unit eigenvector
対角化	diagonalization	直交行列	orthogonal matrix
1 次変換	linear transformation	2 次形式	quadratic form
項	term	2 次曲線	quadratic curve
概形	rough sketch		

確率・統計

解の導出過程も書くこと。

[1] ある事象が2分間に平均1回発生する場合、10分間でその事象が3回発生する確率を有効数字2桁で求めなさい。ただし、その事象の生起回数の確率はポアソン分布に従うものとし、 $e^{-1} = 0.37, e^{-2} = 0.14, e^{-3} = 0.050, e^{-4} = 0.018, e^{-5} = 0.0067$ とする。

[2] 統計学的仮説検定の手順を、帰無仮説および有意水準という用語を説明した上で、それらを用いて300字程度 (about 250 words for English) で説明しなさい。

[3] 互いに独立な確率変数 X と Y がともに正規分布 $N(\mu, \sigma^2)$ に従うとする。この確率密度関数は $\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ で表される。このとき、以下の問いに答えなさい。

- (1) 確率変数 X のモーメント母関数 (積率母関数) は e^{tX} の期待値、つまり $E[e^{tX}]$ で定義される。 X のモーメント母関数を μ と σ を使って表しなさい。
- (2) 確率変数 Z を $Z = X + Y$ とする。 Z のモーメント母関数を求め、 Z も正規分布に従うことを示しなさい。

Translation of Technical Terms

- 事象 event
- 平均 average
- 有効数字 significant digit
- 生起回数 number of occurrences
- ポアソン分布 Poisson distribution
- 統計学的仮説検定 statistical hypothesis testing
- 帰無仮説 null hypothesis
- 有意水準 level of significance
- 互いに独立 mutually independent
- 確率変数 random variable
- 正規分布 normal distribution
- 確率密度関数 probability density function
- モーメント母関数 (積率母関数) moment-generating function
- 期待値 expected value

プログラミング

ソースコード1は整数の集まりを操作するためのデータ構造を扱うC言語プログラムである。このプログラムについて以下の問いに答えよ。

- (1) ソースコード1を実行した際に標準出力に出力される文字列を書け。
- (2) ポインタheadから参照される整数の集まりは、関数insert, top, eliminateが実現している操作の下でどのようなデータ構造をなしているか？ 下記の中から最も適切なものを1つ選択せよ。

(a) LIFO (b) FIFO (c) ヒープ (d) 2分木 (e) DAG

- (3) 関数topはポインタheadの値がNULLであるときに-1を返す。この実装では、保持されているデータが存在するかどうかを関数topが返す値からは必ずしも正しく判断できない。正しく判断できない場合を30文字（英語の場合は20単語）以内で説明せよ。
- (4) ソースコード1では関数insertによってデータを追加するときすべてのデータを走査する必要があり効率が悪い。これを改善する方法として、ポインタheadとは別に末端の要素のアドレスを保持するポインタtailを導入して以下のように改変することが考えられる。以下の (ア) ~ (ス) を埋めて関数insert, eliminateを完成させよ。

- 11行目に以下の文を追加する。

```
CELL tail = NULL;
```

- 関数insert, eliminateのソースコードを以下のように変更する。

```
void insert(int i) {  
    CELL c = (CELL) malloc(sizeof(struct cell));  
    c->num = i; c->next = NULL;  
    if(tail != NULL) {  
        (ア) = (イ);  
    } else {  
        (ウ) = (エ);  
    }  
    (オ) = (カ);  
}
```

```
void eliminate() {  
    if(head != NULL) {  
        if( (キ) != NULL) {  
            (ク) = (ケ);  
        } else {  
            (コ) = (サ);  
            (シ) = (ス);  
        }  
    }  
}
```

- (5) ソースコード1の関数 eliminate ではメモリ領域^{りょういき かいほう}を解放すべきところで解放していない。適当な2行を追加して、不要になったメモリ領域を解放するように修正せよ。なお、解答は「〇〇行目と次の行の間に△△を挿入」という形式で記し、追加する2行が連続していない場合は追加する行ごとに「〇〇行目と次の行の間に△△を挿入」という形式で記すこと。また、メモリ解放には標準^{ひょうじゆん}ライブラリ関数である free を使用すること。
- (6) 保持するデータの個数の最大数を決められる場合には、ソースコード1のような動的^{どうてき}にサイズが変化するデータ構造ではなく、配列^{はいれつ}を用いても同様の処理^{しゆり}を実現できる。ソースコード1の実行結果^{じっこうけつか}と一致するように、ソースコード2の中の (セ) ~ (タ) を埋めよ。

Translation of technical terms

ソースコード	source code	値	value
整数	integer	返す	return
データ構造	data structure	実装	implementation
C言語	C programming language	走査する	scan
プログラム	program	保持する	store
実行する	execute	文	statement
標準出力	standard output	メモリ領域	memory space
出力する	output	解放する	free
文字列	string	標準ライブラリ	standard library
ポインタ	pointer	動的に	dynamically
参照する	refer to	配列	array
関数	function	処理	processing
ヒープ	heap	実行結果	result of execution
2分木	binary tree		

ソースコード 1:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct cell {
5     int num;
6     struct cell *next;
7 };
8
9 typedef struct cell *CELL;
10 CELL head = NULL;
11
12 void insert(int i) {
13     CELL c = (CELL) malloc(sizeof(struct cell));
14     CELL tmp = head;
15     c->num = i; c->next = NULL;
16
17     if(head != NULL) {
18         while(tmp->next != NULL) tmp = tmp->next;
19         tmp->next = c;
20     } else {
21         head = c;
22     }
23 }
24
25 int top() {
26     if(head != NULL) {
27         return head->num;
28     } else {
29         return -1;
30     }
31 }
32
33 void eliminate() {
34     if(head != NULL) {
35         head = head->next;
36     }
37 }
38
39 void display() {
40     CELL tmp = head;
41     while(tmp != NULL) {
42         printf("%d;", tmp->num);
43         tmp = tmp->next;
44     }
45     printf("\n");
46 }
47
48 int main() {
49     insert(0); insert(4); insert(9); insert(3);
50     display();
51     printf("%d\n", top());
52     eliminate(); eliminate(); insert(7); insert(2);
53     display();
54     return 0;
55 }
```

ソースコード 2:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAX 5
5
6 int ar[MAX];
7
8 int head = 0;
9 int cnt = 0;
10
11 void insert(int i) {
12     if(cnt >= MAX) {
13         printf("error\n");
14         exit(1);
15     }
16     ar[ (七) ] = i;
17     cnt++;
18 }
19
20 int top() {
21     if(cnt > 0) {
22         return ar[head];
23     } else {
24         return -1;
25     }
26 }
27
28 void eliminate() {
29     if(cnt > 0) {
30         head = (ノ) ;
31         cnt--;
32     }
33 }
34
35 void display() {
36     int i = 0;
37     while(i < cnt) {
38         printf("%d;", ar[ (タ) ]);
39         i++;
40     }
41     printf("\n");
42 }
43
44 int main() {
45     insert(0); insert(4); insert(9); insert(3);
46     display();
47     printf("%d\n", top());
48     eliminate(); eliminate(); insert(7); insert(2);
49     display();
50     return 0;
51 }
```

