

# 知能システム学 I

次の3問すべてを解答せよ。問題毎に別の解答用紙を用いること。解答用紙の追加は認めない。

## 1

以下の問い1), 2) に答えよ。

1) 以下の小問 a), b) に答えよ。

a)  $\int_0^{\infty} \frac{\sin x}{x} dx$  が収束することを証明せよ。

b)  $\int_0^{\infty} \frac{|\sin x|}{x} dx$  が発散することを証明せよ。

2) 2次元平面上の極座標表示  $(r, \theta)$  において、 $r = 1 + \cos \theta$  ( $0 \leq \theta \leq \pi$ ) で表される曲線  $C$  を考える。以下の小問 a) ~ d) に答えよ。

a) 曲線  $C$  の概形を、直交座標  $(x, y)$  上で描け。また、この曲線  $C$  と  $y$  軸との交点、および  $\theta = \frac{2}{3}\pi$  の点の直交座標をそれぞれ求めよ。

b) 曲線  $C$  の長さ  $L$  を求めよ。

c) 曲線  $C$  と  $x$  軸で囲まれた内部の面積  $S$  を求めよ。

d) 曲線  $C$  と  $x$  軸で囲まれた部分が、 $x$  軸の周りに1回転してできる回転体の体積  $V$  を求めよ。

2

以下の問い 1), 2) に答えよ。ただし,  $T$  は転置を表す。

- 1) 3次元ベクトル  $l = (l_x, l_y, l_z)^T$ ,  $m = (m_x, m_y, m_z)^T$ ,  $n = (n_x, n_y, n_z)^T$  が一次従属となる必要十分条件は, 零行列でない  $3 \times 3$  行列  $M$  を用いて  $l^T M n = 0$  と表すことができる。これを満たす行列  $M$  を一つ求めよ。

- 2)  $A = \begin{pmatrix} 2 & 0 \\ 0 & 2 \\ 2 & 1 \end{pmatrix}$ ,  $x = (x_1, x_2)^T$ ,  $b = (2, 6, 4)^T$  とする。次の小問 a) ~ f) に答えよ。

- a)  $J = \|Ax - b\|^2$  のとき,  $\frac{\partial J}{\partial x} = \left( \frac{\partial J}{\partial x_1}, \frac{\partial J}{\partial x_2} \right)^T$  を求めよ。ただし,  $\|\cdot\|$  はユークリッドノルムを表す。
- b)  $J$  を最小化する  $x$  を  $\hat{x}$  と表す。  $\hat{x}$  を求めよ。
- c)  $B = A^T A$  とする。  $B$  の固有値  $\lambda_1, \lambda_2$  ( $\lambda_1 \geq \lambda_2$ ) を求めよ。また,  $\lambda_1, \lambda_2$  に対応し, 正規直交系をなす  $B$  の固有ベクトル  $v_1, v_2$  をそれぞれ求め, 行列  $V = \begin{pmatrix} v_1 & v_2 \end{pmatrix}$  を求めよ。
- d)  $C = AA^T$  とすると,  $\lambda_1, \lambda_2$  が  $C$  の固有値となることを示せ。
- e) これらの固有値  $\lambda_1, \lambda_2$  に対応し, 正規直交系をなす  $C$  の固有ベクトル  $u_1, u_2$  をそれぞれ求め, 行列  $U = \begin{pmatrix} u_1 & u_2 \end{pmatrix}$  を求めよ。
- f)  $D = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$ ,  $\sigma_i = \sqrt{\lambda_i}$ , ( $i=1,2$ ) とすると,  $A = UDV^T$  と表すことができる。このとき,  $(A^T A)^{-1} A^T = VD^{-1}U^T$  を導出せよ。

## 3

プログラム1は二分木のなぞり（走査）を行うC言語プログラムである。二分木は構造体 `binarytree` で定義される。 `create_binary_tree` は標準入力から、英文大文字を各ノードのメンバ `data` とする二分木をつくる関数である。 `preorder_traverse` と `inorder_traverse` は、それぞれ行きがけ順（前順）と通りがけ順（中順）で再帰的に二分木をなぞり、ノードのデータを標準出力に書き出す関数である。 `preorder_loop_traverse` は、スタックとループ（繰り返し）処理を使って二分木を行きがけ順でなぞり、ノードのデータを標準出力に書き出す関数である。スタックは構造体 `stack` で定義される。 `push_stack` はスタックにデータを追加する関数である。 `pop_stack` はスタックからデータを取り出す関数である。このプログラムについて、以下の問い1)~5)に答えよ。

- 1) AB1234C##D##E##を標準入力したときつくられる二分木を図示せよ。ただし、各ノードは、メンバ `data` の英文大文字を○で囲むことで表し、メンバ `lchild`, `rchild` が指すノードを、それぞれ左下、右下に直線で接続すること。
- 2) 関数 `inorder_traverse` の中の空欄  を適切な処理が行われるように埋めよ。ただし、文が複数となっても良い。
- 3) 関数 `push_stack` の中の空欄  と、関数 `pop_stack` の中の空欄  を適切な処理が行われるように埋めよ。ただし、文が複数となっても良い。
- 4) 関数 `preorder_loop_traverse` の中の空欄 ,  を適切な処理が行われるように埋めよ。ただし、文が複数となっても良い。
- 5) プログラム1を実行したとき、標準入力 AB##CE###に対して、問題なく動作したとする。このとき、87行目、88行目、89行目がそれぞれ実行されたときに出力される標準出力を示せ。

## プログラム1

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  struct binarytree {
4      char data;
5      struct binarytree *lchild;
6      struct binarytree *rchild;
7  };
8  typedef struct binarytree btree;
9  typedef struct binarytree *pbtree;
10 void create_binary_tree(pbtree *t) {
11     char chin;
12     while(1) {
13         scanf("%c", &chin);
14         if(chin>='A' && chin<='Z') break;
15         if(chin=='#') { *t=NULL; return; }
16     }
17     *t = (pbtree)malloc(sizeof(btree));
18     if(!*t) {
19         printf("ERROR: Failed to allocate mem for a.btree node.\n");

```

## —問題3の続き—

```
20     exit(1);
21     }
22     (*t)->data = chin;
23     create_binary_tree(&((*t)->lchild));
24     create_binary_tree(&((*t)->rchild));
25     }
26 void preorder_traverse(pbtree t) {
27     if(t!=NULL) {
28         printf("%c", t->data);
29         preorder_traverse(t->lchild);
30         preorder_traverse(t->rchild);
31     } else return;
32     }
33 void inorder_traverse(pbtree t) {
34     if(t!=NULL) {
35         
36     } else return;
37     }
38 #define STACKSIZE 100
39 struct stack {
40     pbtree *base;
41     pbtree *top;
42     int stacksize;
43 };
44 typedef struct stack btrestack;
45 typedef struct stack *pbtrestack;
46 void init_stack(pbtrestack s) {
47     s->base = (pbtree*)malloc(STACKSIZE*sizeof(pbtree));
48     if(!s->base) {
49         printf("ERROR: Failed to allocate mem for a stack.\n");
50         exit(1);
51     }
52     s->top = s->base;
53     s->stacksize = STACKSIZE;
54 }
55 void push_stack(pbtrestack s, pbtree e) {
56     if(s->top - s->base >= s->stacksize) {
57         printf("ERROR: Stack is overflowed.\n"); exit(1);
58     }
59     
60 }
61 pbtree pop_stack(pbtrestack s) {
62     if(s->top == s->base) {
63         printf("ERROR: Stack is empty.\n"); exit(1);
64     }
65     
66 }
67 int isempty_stack(pbtrestack s) {
68     if(s->top == s->base) return 1; else return 0;
69 }
70 void preorder_loop_traverse(pbtree t) {
71     pbtree p = t;
72     btrestack s;
73     init_stack(&s);
74     while(p || !isempty_stack(&s)) {
75         if(p) {
```

## —問題3の続き—

```
76     printf("%c", p->data);
77         
78     } else {
79         p = pop_stack(&s);
80         
81     }
82 }
83 }
84 int main() {
85     pbtree t = NULL;
86     create_binary_tree(&t);
87     preorder_traverse(t);
88     inorder_traverse(t);
89     preorder_loop_traverse(t);
90     return 0;
91 }
```