

知能システム学 I

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

1

以下の問い1)~3)に答えよ。

- 1) 室温 10°C の部屋で 90°C のお湯がコップに注がれた。その3分後、コップのお湯は 70°C になった。室温は一定であり、お湯の温度の低下速度は室温との温度差に比例するものとする。 x をお湯の温度、 t を時間として微分方程式を立て、コップにお湯が注がれてから6分後のお湯の温度を求めよ。
- 2) オーストリアの生物学者フォン・ベルタランフィは魚の体重 (y) の時間変化について式 (1) のモデルを提唱した。ただし t は時間、 a 、 b は正定数 ($a > b$) である。

$$\frac{dy}{dt} = ay^{\frac{2}{3}} - by \quad (1)$$

式 (1) 右辺の第1項は栄養分による体重増加、第2項は呼吸による体重減少を表している。生まれた瞬間 ($t = 0$) の体重を1として以下の小問 a)、b) に答えよ。

- a) y を t の関数として求めよ。
 - b) 十分時間が経過したとき、魚の体重がいくらに収束するか求めよ。
- 3) 式 (2) の曲線群について以下の小問 a)、b) に答えよ。ただし C_0 は非ゼロの任意定数とする。

$$x^2 + y^2 = C_0 x \quad (C_0 \neq 0) \quad (2)$$

- a) 式 (2) を満たし、 C_0 を含まない一階微分方程式を求めよ。
- b) 小問 a) で求めた微分方程式から、式 (2) の曲線群と直交する曲線群を導出せよ。

2

以下の問い 1), 2) に答えよ。

- 1) 2×2 の行列 $A = \begin{pmatrix} 1-p & q \\ p & 1-q \end{pmatrix}$ について考える。ただし、 p, q は実数とする。

以下の小問 a) ~ d) に答えよ。

- A の固有値をすべて求めよ。
- A が対角化可能かどうかを判定し、対角化可能である場合は対角化せよ。
- A^n を 2×2 の行列の形で表せ。ただし、 n は正整数である。
- $\lim_{n \rightarrow \infty} A^n$ が、定数のみを要素に持つ行列 A^∞ に収束するための p, q の条件を求めよ。
また、 A^∞ をすべて求めよ。

- 2) 2×3 の行列 $A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ について考える。以下の小問 a) ~ c) に答えよ。

- a) $ABA = A$ を満たす 3×2 の行列 B を $B = \begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix}$ とおく。

ただし、 a, b, c, d, e, f は実数である。 c, d, e, f を a, b を用いて表せ。

- b) $ABA = A$ に加え、次の三つの条件をすべて満たす B を B^+ とする。

$$(i) (BA)^T = BA \quad (ii) (AB)^T = AB \quad (iii) BAB = B$$

ただし、 T は転置を表す。 B^+ を求めよ。

- c) 3次元変数ベクトルを x 、2次元定数ベクトルを c とする連立一次方程式 $Ax = c$ を考える。二乗ノルム $\|x\|^2$ を最小にする解を、 B^+ と c を用いて表せ。

3

以下の問い1), 2)に答えよ。

- 1) プログラム1は、B木の探索のC言語プログラムである。このプログラムでは、B木の各ノードは最大M個のキーとM+1個の子ノードを持つ。ノードは構造体BTNodeで定義している。メンバ変数key_numは、各ノードが保持するキーの個数を格納する。キーの値は配列keyに格納する。また、そのノードの親ノードへのポインタをparentとし、最大M+1個ある子ノードへのポインタを配列childに格納する。関数search_btree(PTree t, int val, Node *r_node, int *r_i)は、B木tにvalをキーとして持つノードが存在するかどうかを調べる関数である。存在する場合、当該ノードを指すポインタをr_nodeに代入する。また、このノードにおいて、valが格納されている配列keyのインデックスの値を、r_iが指すメモリ領域に代入する。存在しない場合、r_nodeをNULL、r_iが指すメモリ領域を-1にする。以下の小問a)~c)に答えよ。
- a) 各ノードが最大M個のキーを持つB木に、N個の要素を登録する。この時、B木のとりうる高さの最小値と最大値を示せ。
- b) プログラム1の20行目のsizeof(Node)の値を示せ。ただし、int型に4バイト、ポインタ型に8バイトを割り当てるものとする。
- c) 空欄 と を埋めよ。ただし、 は、複数の文になってもよい。

プログラム1

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #define M 3
4  typedef struct BTNode {
5      int key_num;
6      int key[M];
7      struct BTNode *parent;
8      struct BTNode *child[M+1];
9  } Node, *PTree;
10 void search_btree(PTree t, int val, Node *r_node, int *r_i) {
11     int i; Node *p = t;
12     while(p != NULL) {
13         i = 0; while(i < p->key_num && val > p->key[i]) i++;
14         if() { r_node = p; *r_i = i; return; }
15         else { p = p->child[i]; }
16     }
17     
18 }
19 int main() {
20     PTree t = (PTree)malloc(sizeof(Node));
21     t->key_num = 2; t->parent = NULL;
22     t->key[0] = 4; t->key[1] = 7;
23     t->child[0] = (PTree)malloc(sizeof(Node));
24     t->child[0]->key_num = 1; t->child[0]->parent = t;
25     t->child[0]->key[0] = 3;
26     t->child[0]->child[0] = NULL;
27     t->child[0]->child[1] = NULL;
28     t->child[1] = (PTree)malloc(sizeof(Node));
29     t->child[1]->key_num = 1; t->child[1]->parent = t;

```

---問題3の続き---

30	t->child[1]->key[0] = 6;
31	t->child[1]->child[0] = NULL;
32	t->child[1]->child[1] = NULL;
33	t->child[2] = (PTree)malloc(sizeof(Node));
34	t->child[2]->key_num = 1; t->child[2]->parent = t;
35	t->child[2]->key[0] = 9;
36	t->child[2]->child[0] = NULL;
37	t->child[2]->child[1] = NULL;
38	Node *r_nd = NULL; int r_id = -1;
39	search_btrees(t, 6, r_nd, &r_id); printf("%d", r_id);
40	return 0;
41	}

- 2) プログラム2はヒープソートを行うC言語プログラムである。作業用領域を効率的に使用するため、整列対象となる配列の中にヒープを構築している。以下の小問a)~d)に答えよ。
- a) 整列対象数を N とするとき、ヒープソートの平均時間計算量をオーダー表記で示せ。また、ヒープソート以外のソートアルゴリズムの中で、平均時間計算量がヒープソートと同じであるアルゴリズムの名前を二つ示せ。
- b) 次の文の下線部 i と j を、それらの直後の括弧内の選択肢から適切に選んで埋めよ。
- ヒープソートは i (安定・不安定) な j (外部・内部) 整列アルゴリズムである。
- c) プログラム2の20行目では2行目の関数 swap を呼び出している。空欄 3 を埋めよ。
- d) プログラム2の実行開始から終了までの標準出力を示せ。

プログラム2

1	#include <stdio.h>
2	void swap(int* a, int* b) { int temp = *a; *a = *b; *b = temp; }
3	void heapdown(int a[], int i, int n) {
4	int child, tmpele;
5	for (tmpele=a[i]; 2*i+1<n; i=child) {
6	child = 2*i+1;
7	if (child!=n-1 && a[child+1]>a[child]) ++child;
8	if (tmpele<a[child]) a[i] = a[child]; else break;
9	}
10	a[i] = tmpele;
11	}
12	void print(int a[], int n) {
13	int i;
14	for(i=0; i<n; i++) printf("%d ", a[i]);
15	printf("\n");
16	}
17	void heapsort(int a[], int n) {
18	int i;
19	for(i=n/2; i>=0; --i) heapdown(a, i, n); print(a, n);
20	for(i=n-1; i>0; --i) { swap(3); heapdown(a, 0, i); }
21	}
22	int main() {
23	int arr[10] = {81, 2, 39, 49, 34, 62, 53, 6, 43, 98};
24	heapsort(arr, 10);
25	return 0;
26	}