

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) 以下のa), b)の不定積分を求めよ。ただし a, b は実数の定数で、 $a^2 + b^2 \neq 0$ とする。

a) $\int (\tan^6 x + \tan^4 x) dx$

b) $\int e^{ax} \sin bx dx$

2) 以下のa), b)の重積分を求めよ。

a) $\iint_D \sin \frac{\pi x}{2y} dx dy$, $D = \{(x, y) \mid y \leq x \leq y^2, 1 \leq y \leq 3\}$

b) $\iint_D 3x^2 y^3 dx dy$, D は直線 $y = x$ と曲線 $y = x^2$ で囲まれた領域

3) 次の xy 平面上の曲線 C ($0 \leq t \leq 2\pi$) について以下の小問a)~c)に答えよ。ただし a は正定数とする。

$$\text{曲線 } C \begin{cases} x = a(t - \sin t) \\ y = a(1 - \cos t) \end{cases}$$

a) xy 平面上の曲線 C の長さを求めよ。

b) 曲線 C と x 軸に囲まれる図形を D とし、 D の面積を求めよ。

c) D を x 軸周りに回転してできる立体の体積を求めよ。

2

n 次正方実行列の全体 $\mathbb{R}^{n \times n}$ は実計量線形空間である。 $M, N \in \mathbb{R}^{n \times n}$ の内積は $(M, N) = \text{tr}(M^T N)$ と、 $M \in \mathbb{R}^{n \times n}$ のノルムは $\|M\| = \sqrt{(M, M)}$ と定義される。ただし、 \mathbb{R} は実数全体の集合を、 $\text{tr}(M)$ と M^T はそれぞれ $M \in \mathbb{R}^{n \times n}$ のトレースと転置を表す。また、 \mathcal{O}_n を n 次実直交行列全体の集合、 \mathcal{S}_n を n 次実対称行列全体の集合とする。以下の問い 1)~7) に答えよ。

1) 次の $M, R(\theta) \in \mathbb{R}^{2 \times 2}$ の内積 $(M, R(\theta))$ を求めよ。ただし、 a, b, c, d, θ は実数である。

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

- 2) $M \in \mathbb{R}^{n \times n}$ に対して、変数 $P \in \mathcal{O}_n$ の関数 $f(P) = (M, P)$ と $g(P) = \|M - P\|$ を考える。 $f(P)$ が最大値をとる P を \hat{P} とおくと、 $g(P)$ は \hat{P} で最小値をとることを示せ。
- 3) 問い 1) の $M, R(\theta)$ に対して、変数 θ ($-\pi \leq \theta < \pi$) の関数 $h(\theta) = \|M - R(\theta)\|$ を考える。 $h(\theta)$ が最小値をとる θ を求めよ。ただし、 $a + d > 0$ 、 $c - b > 0$ とする。
- 4) \mathcal{O}_n と \mathcal{S}_n はそれぞれ $\mathbb{R}^{n \times n}$ の部分空間か。理由とともに答えよ。
- 5) $M \in \mathbb{R}^{n \times n}$ は次のように実対称行列と実交代行列（実歪対称行列）の和で表すことができる。

$$M = \frac{1}{2}(M + M^T) + \frac{1}{2}(M - M^T)$$

次の $M \in \mathbb{R}^{2 \times 2}$ を実対称行列と実交代行列（実歪対称行列）の和で表せ。

$$M = \begin{bmatrix} 1 & -4 \\ 6 & 2 \end{bmatrix}$$

- 6) 実対称行列と実交代行列（実歪対称行列）は $\mathbb{R}^{n \times n}$ において直交することを示せ。
- 7) $M \in \mathbb{R}^{n \times n}$ に対して、変数 $S \in \mathcal{S}_n$ の関数 $v(S) = \|M - S\|$ を考える。 $v(S)$ の最小値を求めよ。

3

以下の問い 1), 2) に答えよ。

- 1) プログラム 1.1 は、二分探索木の C 言語プログラムである。このプログラムの 1~5 行目は、探索キーを val、左右の子ノードへのポインタをそれぞれ left, right とする二分探索木のデータ構造を定義するコードである。6~12 行目の create_node は、左右の子ノードにそれぞれ空の木を持つ二分探索木を作成する関数である。13~20 行目の insert_node は、引数 val をキーとする新しいノードを作成し、それを引数 root が指す二分探索木に挿入する関数である。21~43 行目の delete_node は、引数 root が指す二分探索木から引数 val と一致するキーを持つノードを削除する関数である。44~50 行目の search_node は、引数 root が指す二分探索木から引数 val と一致するキーを持つノードを探し、そのポインタを返す関数である。探索対象のノードが見つからない場合は NULL を返す。insert_node, delete_node, search_node 各関数は再帰的に定義されている。また、プログラム 1.1 は、ヘッダーファイルの参照部分と main 関数部分を省略している。このプログラムに関する以下の小問 a)~d) に答えよ。
- a) 空欄①を埋めよ。また、int のサイズを 4 バイト、ポインタのサイズを 8 バイトと想定する場合、プログラム 1.1 で定義される二分探索木の一つのノードのサイズを答えよ。
- b) プログラム 1.1 の 40 行目は、再帰的に delete_node 関数を呼び出している。この行の空欄②を埋めよ。
- c) プログラム 1.1 の 45 行目は、search_node 関数の再帰呼び出しを停止するかどうかを判断している。この行の空欄③を埋めよ。
- d) プログラム 1.2 はプログラム 1.1 の delete_node 関数を非再帰的な方法で実装したものである。このプログラムの空欄④と空欄⑤を埋めよ。

プログラム 1.1

```

1 struct TreeNode {
2     int val;
3     struct TreeNode *left;
4     struct TreeNode *right;
5 };
6 struct TreeNode* create_node(int val) {
7     struct TreeNode* new_node = ①;
8     new_node->val = val;
9     new_node->left = NULL;
10    new_node->right = NULL;
11    return new_node;
12 }
13 struct TreeNode* insert_node(struct TreeNode* root, int val) {
14     if (root == NULL) return create_node(val);
15     if (val < root->val)
16         root->left = insert_node(root->left, val);
17     else if (val > root->val)
18         root->right = insert_node(root->right, val);
19     return root;
20 }
21 struct TreeNode* delete_node(struct TreeNode* root, int val) {
22     if (root == NULL) return root;

```

---問題3の続き---

```

23     if (val < root->val)
24         root->left = delete_node(root->left, val);
25     else if (val > root->val)
26         root->right = delete_node(root->right, val);
27     else {
28         if (root->left == NULL) {
29             struct TreeNode* temp = root->right;
30             free(root);
31             return temp;
32         } else if (root->right == NULL) {
33             struct TreeNode* temp = root->left;
34             free(root);
35             return temp;
36         }
37         struct TreeNode* temp = root->right;
38         while(temp->left != NULL) temp = temp->left;
39         root->val = temp->val;
40         root->right = ;
41     }
42     return root;
43 }
44 struct TreeNode* search_node(struct TreeNode* root, int val) {
45     if ()
46         return root;
47     if (val < root->val)
48         return search_node(root->left, val);
49     return search_node(root->right, val);
50 }

```

プログラム 1.2

```

1  struct TreeNode* delete_node(struct TreeNode* root, int val) {
2      struct TreeNode *parent = NULL, *current = root;
3      while (current != NULL && current->val != val) {
4          parent = current;
5          if (val < current->val)
6              current = current->left;
7          else
8              current = current->right;
9      }
10     if (current == NULL) return root;
11     if () {
12         struct TreeNode *new_curr;
13         if (current->left == NULL)
14             new_curr = current->right;
15         else
16             new_curr = current->left;
17         if (parent == NULL)
18             return new_curr;
19         if (current == parent->left)
20             parent->left = new_curr;
21         else
22             parent->right = new_curr;
23         free(current);
24     } else {
25         struct TreeNode *p = NULL;
26         struct TreeNode *temp;
27         temp = current->right;
28         while (temp->left != NULL) {
29             p = temp;
30             temp = temp->left;

```

—問題3の続き—

```

31     }
32     current->val = temp->val;
33     if (p != NULL) {
34         ;
35     } else {
36         current->right = temp->right;
37     }
38     free(temp);
39 }
40 return root;
41 }

```

- 2) プログラム2はC言語によるクイックソートの再帰的実装である。ここでは、ヘッダーファイルの参照部分を省略している。以下の小問a)~c)に答えよ。
- a) 整列対象数をNとするとき、クイックソートの平均時間計算量をオーダー表記で示せ。
- b) このプログラムに使われる swap 関数は、引数として受け取る二つのポインタが指す値を入れ替える。空欄⑥と空欄⑦を埋めよ。ただし、空欄⑦は複数行になってもよい。
- c) 24行目の配列 arr を整列したい時、空欄⑧を埋めよ。ただし、arr の要素数が必要な場合は、sizeof 関数によって求めること。

プログラム2

```

1 void swap() {
2     
3 }
4 int partition(int arr[], int low, int high) {
5     int pivot = arr[high];
6     int i = (low - 1);
7     for (int j = low; j <= high - 1; j++) {
8         if (arr[j] < pivot) {
9             i++;
10            swap(&arr[i], &arr[j]);
11        }
12    }
13    swap(&arr[i + 1], &arr[high]);
14    return (i + 1);
15 }
16 void quick_sort(int arr[], int low, int high) {
17     if (low < high) {
18         int pi = partition(arr, low, high);
19         quick_sort(arr, low, pi - 1);
20         quick_sort(arr, pi + 1, high);
21     }
22 }
23 int main() {
24     int arr[] = {10, 7, 8, 9, 1, 5};
25     ;
26     return 0;
27 }

```