

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)~4)に答えよ。

- 1) 次の関数の導関数を求めよ。ただし $x > 0$ とする。

$$y = x^{x^x}$$

- 2) 一辺の長さが1kmの正方形の頂点に建つ工場 A, B, C, D (図1a) 全てに行き来できる道路を設計する。設計例のいくつかを図1bに示す。ただし工場の大きさ、道路の幅は無視できるものとする。総延長が最小となる道路の長さを求めよ。

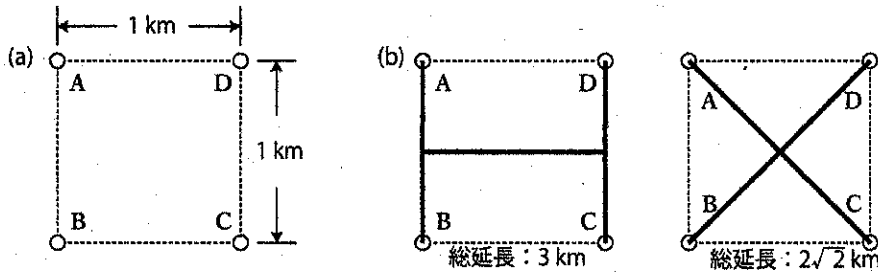


図 1: (a) 工場 A, B, C, D の配置, (b) 道路の設計例

- 3) 直円錐（頂点と底面の中心を結ぶ直線が底面に直交する円錐）の表面積が一定のとき、その体積が最大となる高さと底面の半径の比を求めよ。
- 4) 半径1の円（中心 O）に外接する三角形 ABC (図2) について以下の小問 a), b) に答えよ。

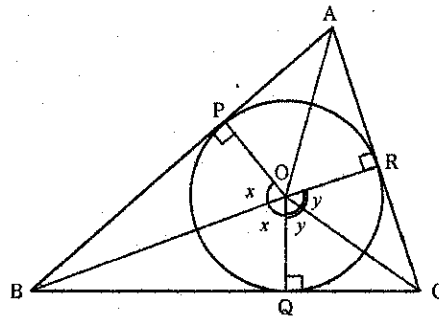


図 2: 三角形 ABC

- a) 3辺 AB, BC, CA と円の接点をそれぞれ P, Q, R とし、 $\angle POQ = 2x$ ($0 < x < \pi/2$), $\angle QOR = 2y$ ($0 < y < \pi/2$) とする。このとき三角形 ABC の面積を x と y の2変数関数で表せ。
- b) 三角形 ABC の面積が最小となる x, y , 並びにそのときの面積を求めよ（導出過程も示すこと）。

2

以下の問い 1), 2) に答えよ.

1) 4次元ベクトル $\mathbf{a}_1, \mathbf{a}_2, \mathbf{b}$ を次のように定める. 以下の小問 a) ~ c) に答えよ.

$$\mathbf{a}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ -2 \end{pmatrix}, \quad \mathbf{a}_2 = \begin{pmatrix} 2 \\ -2 \\ 2 \\ -1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 2 \end{pmatrix}$$

a) $\mathbf{a}_1, \mathbf{a}_2$ で張られる線形空間への \mathbf{b} の4次元直交射影ベクトル \mathbf{p} を求めよ.

b) 4×2 の行列 $\mathbf{A} = (\mathbf{a}_1 \ \mathbf{a}_2)$, 2次元ベクトル $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ に対し, $J = \|\mathbf{Ax} - \mathbf{b}\|^2$ を最小にする \mathbf{x} を $\hat{\mathbf{x}}$ とする. ただし, $\|\cdot\|$ はユークリッドノルムを表す. $\hat{\mathbf{x}}$ を求めよ.

c) $\hat{\mathbf{x}}, \mathbf{A}, \mathbf{p}$ の間に成り立つ関係式を示せ.

2) 3×3 の行列 $\mathbf{A} = \begin{pmatrix} 1 & p & q \\ 2 & q & 4 \\ p & 1 & q \end{pmatrix}$ と3次元ベクトル $\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$ を用いて表される連立一次方程式

$$\mathbf{Ax} = \mathbf{b}$$

を考える. ただし, p, q, b_1, b_2, b_3 は実定数とする. この連立一次方程式の解空間が,

$$\left\{ \begin{pmatrix} \alpha + 2 \\ -3\alpha + 2\beta \\ \alpha - \beta + 1 \end{pmatrix} \mid \alpha \text{ と } \beta \text{ は実定数} \right\}$$

で与えられるとする. 以下の小問 a) ~ c) に答えよ.

a) p, q, b_1, b_2, b_3 の値を定めよ.

b) a) で決定した \mathbf{A} を対角化せよ.

c) a) で決定した \mathbf{A} に対し \mathbf{A}^n を 3×3 の行列の形で表せ. ただし, n は正整数である.

3

以下の問い 1), 2) に答えよ。

- 1) プログラム 1-1 は、再帰呼び出しで実装したクイックソートの C 言語プログラムである。ソート対象となるデータは SortList 構造体を用いて保存する。この構造体はデータ要素を持つ配列 r[MAX] と要素の数を表す変数 length によって構成される。クイックソートの整列対象となるデータは配列の 1 番目から length 番目までの要素に格納される。このプログラムに関する以下の小問 a) ~ f) に答えよ。
- a) プログラム 1-1 の 10 行目の sizeof(SortList) の戻り値を示せ。ただし、int 型は 4 バイトとする。
- b) 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。また、プログラム 1-1 で代入されているデータは整列済みで、時間計算量の観点で最悪のケースである。この場合の時間計算量をオーダー表記で示せ。ただし、いずれもデータ数を n として解答すること。
- c) 最悪のケースを回避するための手法について述べよ。
- d) 空欄 を埋めよ。
- e) 空欄 と を埋めよ。
- f) プログラム 1-2 はプログラム 1-1 の関数 quick_sort(SortList *sl) の非再帰版である。スタック low_stack[MAX] と high_stack[MAX] を用いて整列を行う範囲を繰り返し管理する。このプログラムの空欄 を埋めよ。複数の文になってもよい。

プログラム 1-1

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #define MAX 100
4  typedef struct { int data; } ElementSort;
5  typedef struct { ElementSort r[MAX]; int length; } SortList;
6  int partition(SortList *sl, int low, int high);
7  void q_sort(SortList *sl, int low, int high);
8  void quick_sort(SortList *sl);
9  int main() {
10     SortList *sl = (SortList*)malloc(sizeof(SortList));
11     sl->length=8;
12     sl->r[1].data=108;
13     sl->r[2].data=107;
14     sl->r[3].data=106;
15     sl->r[4].data=105;
16     sl->r[5].data=104;
17     sl->r[6].data=103;
18     sl->r[7].data=102;
19     sl->r[8].data=101;
20     quick_sort(sl);
21     for(int i=1; i<=sl->length; i++) printf("%d ", sl->r[i].data);
22     free(sl); return 0;
23 }
24 int partition(SortList *sl, int low, int high) {

```

—問題3の続き—

```

25     int pivot_data = sl->r[low].data;
26     sl->r[0] = sl->r[low];
27     while(low<high) {
28         while(low<high && sl->r[high].data>=pivot_data) high--;
29         sl->r[low] = sl->r[high];
30         while(low<high && sl->r[low].data<=pivot_data) low++;
31         sl->r[high] = sl->r[low];
32     }
33     sl->r[low] = ;
34     return low;
35 }
36 void q_sort(SortList *sl, int low, int high){
37     if(low<high) {
38         int pivot_index = partition(sl, low, high);
39         q_sort(sl, low, );
40         q_sort(sl, , high);
41     }
42 }
43 void quick_sort(SortList *sl){
44     q_sort(sl, 1, sl->length);
45 }

```

プログラム 1-2

```

1     void quick_sort_loop(SortList *sl) {
2         int sp = 1, low, high, low_stack[MAX], high_stack[MAX];
3         low_stack[0] = 1; high_stack[0] = sl->length;
4         while(sp>0) {
5             sp--; low = low_stack[sp]; high = high_stack[sp];
6             if(low<high) {
7                 int pivot_index = partition(sl, low, high);
8                 
9             }
10        }
11    }

```

2) プログラム2は自然数を対象とする基数ソートのC言語プログラムである。関数 radix_sort(int *a, int n)内では、十進数の各桁について順に分布数え上げソートを行っている。配列 b[MAX]は分布数え上げソートごとの結果を保存する。このプログラムに関する以下の小問 a), b)に答えよ。

- a) 基数ソートの平均時間計算量をオーダー表記で示せ。ただし、データ数を n としして解答すること。
b) 空欄 と を埋めよ。

プログラム 2

```

1     #include <stdio.h>
2     #define MAX 20
3     #define BASE 10
4     void print(int *a, int n);
5     void radix_sort(int *a, int n);
6     int main() {
7         int i, n, arr[MAX];
8         printf("Enter total elements (n <= %d) : ", MAX);
9         scanf("%d", &n); n = n < MAX ? n : MAX;
10        printf("Enter %d Elements : ", n);

```

—問題3の続き—

```
11     for(i=0; i<n; i++) scanf("%d", &arr[i]);
12     printf("\nARRAY : "); print(arr, n);
13     radix_sort(arr, n);
14     printf("\nSORTED : "); print(arr, n);
15     return 0;
16 }
17 void radix_sort(int *a, int n) {
18     int i, b[MAX], m = a[0], exp = 1;
19     for(i=1; i<n; i++) { if(a[i]>m) m = a[i]; }
20     while(D.1) {
21         int count[BASE] = { 0 };
22         for(i=0; i<n; i++) count[(a[i]/exp)%BASE]++;
23         for(i=1; i<BASE; i++) count[i] += count[i - 1];
24         for(i=n-1; i>=0; i--) b[D.2] = a[i];
25         for(i=0; i<n; i++) a[i] = b[i];
26         exp *= BASE;
27         printf("\nProcessing...: "); print(a, n);
28     }
29 }
30 void print(int *a, int n) {
31     int i;
32     for(i=0; i<n; i++) printf("%d\t", a[i]);
33     printf("\n");
34 }
```