

■設問 1 アルゴリズムとデータ構造概説

以下の文章を読んで、 ～ にあてはまる適切な語句や数値を、次の選択肢から選んで、それぞれ答えなさい。

ある問題に対して、これを解決するための方法や計算手順、手続きのことを と呼ぶ。プログラムは、この に具体的なデータを与えることにより成り立つ。また、あるデータ群を効率よく操作するために使用するデータの構造を と呼ぶ。ここで、データ群を効率よく操作する方法とは、上述した に他ならない。よって、 と には、密接な関係がある。

データ群に対して、あるキーの昇順、あるいは降順にデータを並べ替える処理のことをソートという。ソートは、 ソートと外部ソートに大別される。外部ソートとは、磁気テープ等、外部記憶装置へのアクセスを前提としたものである。これに対して、 ソートとは、メモリに記憶された配列をソートするものであり、通常、単にソートと言った場合、この ソートのことを指す。

ソートとして、いくつかの有名な手法が存在する。その代表的なものに、単純選択法やクイックソート法がある。単純選択法においては、そのアルゴリズムからも明白なように、すべてのデータを総当たりで比較している。すなわち、 n 個のデータをソートすることを考えると、1 個のデータと残りの全てのデータを比較するのに $O(\text{ })$ の時間計算量が必要であり、さらに、これを n 回繰り返す必要がある。そのため、このソート全体にかかる時間計算量は $O(\text{ })$ になるといえる。

これに対して、クイックソート法は、基本的には、すべてのデータを総当たりには比較しない巧妙な手法である。具体的には、ピボットを基準として、データを大小 2 つのグループに分割する、という処理を繰り返していく。 n 個のデータをソートすることを考えると、これをそれぞれ 1 つになるまで、大小 2 つのグループへ分割していくのにかかる時間計算量は理想的には $O(\text{ })$ となる。これは、たとえば、16 個のデータをそれぞれ 1 つになるまで分割していくとすると、この回数は理想的には 4 回になるということである。この際、各段階で処理され

るデータ数は最大でも n である。よって、このソート全体にかかる時間計算量は $O(\text{キ})$ になるといえる。

一方で、空間計算量について考える。単純選択法やバブルソート法においては、基本的には、データの置換等に必要なメモリ領域しか余分に消費しないことから、この空間計算量は $O(\text{ク})$ となる。これに対して、クイックソート法は、ふつう、再帰関数による再帰的な処理により実現されることから、これに要するメモリ領域が必要となり、その空間計算量は $O(\text{ケ})$ となる。

上述した手法は、いずれも比較に基づいた相対的なものである。これに対して、比較によらない絶対的なソート手法として、バケットソート法がある。この手法においては、キーの比較を行う必要がなく、各キーをそれが対応する容器に順番に格納し、これを順番に出力するだけである。1つのデータを任意の容器に格納する、あるいは、任意の容器から取り出すのに要する時間計算量は $O(\text{コ})$ となる。これを n 個のデータについて行うので、このソート全体にかかる時間計算量は $O(\text{サ})$ になるといえる。

しかしながら、バケットソート法の空間計算量は $O(\text{シ})$ となり、厳密には、 ウ ソートの要件を満たさないほど、膨大なメモリ領域が必要となる。これをある程度、解消するため、 k 桁の数値など、一定の条件に特化したソート手法が基数ソート法である。この手法においては、最下位桁から最上位桁まで随時 x 桁目の要素をキーとしてバケットソートを繰り返す。よって、この時間計算量は $O(\text{ス})$ になるといえる。

このように、各種アルゴリズムには一長一短があるので、その処理対象、使用用途に合わせ、適切なものを選択する必要がある。

選択肢：

アル＝フワリズム、アルゴリズム、シナリオ、データ構造、ラーメン構造、内部、中部、0, 1, 3, n , $\sqrt{3}n$, $3n$, $\log n$, $n \log n$, n^2 , $3n^2$, 2^n

■設問2 ソート1

9 個のデータからなる以下の配列について，次の問いに答えなさい．並べ替えはすべて昇順に行うものとする．また，各置換結果，挿入結果には，自分自身を入れ替えるような見た目が変化しないものは含めないものとする．

配列：3, 5, 7, 1, 2, 9, 6, 8, 4

- (1) 以下の表は，単純選択法を用いて整列するときのすべての置換結果（配列の状態）を表している．空欄をすべて埋め（ア）～（オ）に入る値を，それぞれ答えなさい．

(0)	3	5	7	1	2	9	6	8	4
(1)	(ア)				(イ)				
(2)									
(3)					(ウ)				
(4)						(エ)			
(5)								(オ)	
(6)	1	2	3	4	5	6	7	8	9

- (2) 以下の表は，単純挿入法を用いて整列するときのすべての挿入結果（配列の状態）を表している．空欄をすべて埋め（ア）～（オ）に入る値を，それぞれ答えなさい．

(0)	3	5	7	1	2	9	6	8	4
(1)		(ア)			(イ)				
(2)									
(3)				(ウ)		(エ)			
(4)								(オ)	
(5)	1	2	3	4	5	6	7	8	9

- (3) 以下の表は，バブルソート法を用いて整列するときのすべての置換結果（配列の状態）を表している．空欄をすべて埋め（ア）～（オ）に入る値を，それぞれ答えなさい．

(0)	3	5	7	1	2	9	6	8	4
(1)									
(2)									
(3)									
(4)									
(5)									
(6)									
(7)						(ア)			
(8)							(イ)		
(9)				(ウ)					
(10)									
(11)									
(12)						(エ)			
(13)							(オ)		
(14)	1	2	3	4	5	6	7	8	9

- (4) 以下の表は，pivot を 5 としてクイックソート法を用いて整列するときの置換結果（配列の状態）を表している．空欄をすべて埋め（ア）～（オ）に入る値を，それぞれ答えなさい．また，分割が終了した後に整列を行うべき左部分配列と右部分配列を答えなさい．

(0)	3	5	7	1	2	9	6	8	4
(1)		(ア)				(イ)			
(2)	(ウ)				(エ)			(オ)	

左部分配列：_____ 右部分配列：_____

■設問 3 ソート 2

9 個のデータからなる以下の配列が与えられたものとする。

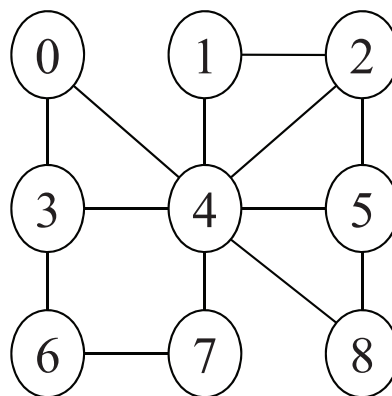
配列 : 175, 122, 912, 285, 563, 473, 404, 91, 667

この配列を，基数ソート法を用いて整列することを考える．次の表は，整列の途中経過を表しており， (i) 行 $(1 \leq i \leq 3)$ は配列を下位から第 i 桁目に関して整列した結果を表している．空欄をすべて埋め（ア）～（オ）に入る数を，それぞれ答えなさい．

(0)	175	122	912	285	563	473	404	91	667
(1)	(ア)				(イ)				(ウ)
(2)			(エ)				(オ)		
(3)	91	122	175	285	404	473	563	667	912

■設問 4 グラフ探索

以下のグラフについて，ノード 0 から到達可能なすべてのノードを探索する．このとき，次の問いに答えなさい．なお，各探索手法において，探索の途中で，選択可能なノードが複数存在した場合は，番号が小さいノードが優先されるものとする．



(1) 深さ優先探索により出力される辺を，順番にすべて答えなさい．

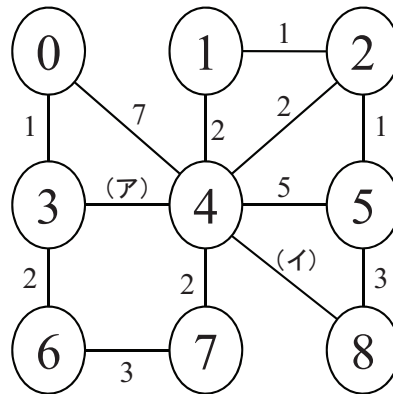
(,)(,)(,)(,)(,)(,)(,)(,)

(2) 幅優先探索により出力される辺を，順番にすべて答えなさい．

(,)(,)(,)(,)(,)(,)(,)(,)

■設問 5 最短経路問題

以下は，ノード間の各経路に重みが付与された重みつき無向グラフと，これに対応する隣接行列である．これについて，次の問いに答えなさい．



	0	1	2	3	4	5	6	7	8
0	∞	∞	∞	1	7	∞	∞	∞	∞
1	∞	∞	(ウ)	∞	2	∞	∞	∞	∞
2	∞	(ウ)	∞	∞	2	1	∞	∞	∞
3	1	∞	∞	∞	8	∞	2	∞	∞
4	7	2	2	8	∞	5	∞	(エ)	7
5	∞	∞	1	∞	5	∞	∞	∞	3
6	∞	∞	∞	2	∞	∞	∞	3	∞
7	∞	∞	∞	∞	(エ)	∞	3	∞	∞
8	∞	∞	∞	∞	7	3	∞	∞	∞

- 上に示す無向グラフ，隣接行列における，空欄（ア）～（エ）に当てはまる値をそれぞれ答えなさい．
- ノード 0 からノード 8 までの最短経路長と最短経路をダイクストラ法により求めることを考える．次ページの表のように，探索の過程において対象となるノード番号 u の値，および，更新されるノード情報 $dist$, $flag$, $path$ の値を順に求めていく．なお，探索の途中で，選択可能なノードが複数存在した場合は，ノード番号の小さいものが優先されるものとする．次ページの表の空欄をすべて埋め（ア）～（オ）に入る記号や値をそれぞれ答えなさい．
- 上記 (2) で得られた結果をもとに，ノード 0 からノード 8 までの最短経路長と最短経路を，それぞれ答えなさい．

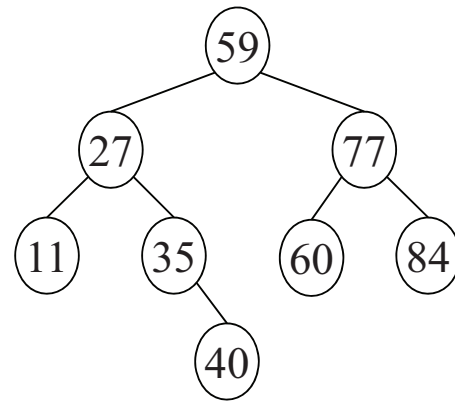
最短経路長：_____ 最短経路：0 → _____ → 8

	0	1	2	3	4	5	6	7	8
node $u = [\quad 0 \quad]$									
<i>dist</i>	0								
<i>flag</i>	1								
<i>path</i>	EOP								
node $u = [\quad \quad]$									
<i>dist</i>	0								
<i>flag</i>	1								
<i>path</i>	EOP								
node $u = [\quad \quad]$									
<i>dist</i>	0								
<i>flag</i>	1								
<i>path</i>	EOP								
node $u = [\quad \quad]$									
<i>dist</i>	0								
<i>flag</i>	1								
<i>path</i>	EOP							(ア)	
node $u = [(イ)]$									
<i>dist</i>	0								
<i>flag</i>	1								
<i>path</i>	EOP								
node $u = [\quad \quad]$									
<i>dist</i>	0								(ウ)
<i>flag</i>	1								
<i>path</i>	EOP								
node $u = [\quad \quad]$									
<i>dist</i>	0								
<i>flag</i>	1								
<i>path</i>	EOP					(エ)			
node $u = [\quad \quad]$									
<i>dist</i>	0								(オ)
<i>flag</i>	1								
<i>path</i>	EOP								
node $u = [\quad 8 \quad]$									
<i>dist</i>	0								
<i>flag</i>	1								1
<i>path</i>	EOP								

■設問 6 二分探索木

右の二分探索木に関して，次の問いに答えなさい。

- (1) データ 60 を探索する際に走査するノードを答えなさい。
例) 11 : 59 → 27 → 11
- (2) 行きがけ順，帰りがけ順にトランバーサルした結果をそれぞれ答えなさい。
例) 通りがけ順 : 11, 27, 35, 40, 59, 60, 77, 84
- (3) データ 38 を追加したとき，生成される二分探索木を答えなさい。



■設問 7 二分木のバランス

キーボードから n 個の数値を読み込みながら，完全にバランスした二分木を生成するソースコードを以下に示す。

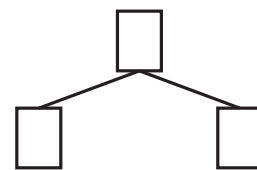
下記 8 個の数値に対して，下記ソースコードの `pbtree` 関数を実行した時の，完全にバランスした二分木を図示せよ。数値は左から順番にキーボードから入力されるものとする。

3, 8, 13, 25, 37, 41, 57, 67

```

typedef struct Node {
    int num;
    struct Node *left, *right;
} node;

node *pbtree(int n) {
    int nleft, nright;
    int nleftplusright = n - 1;
    node *p;
    if(n == 0)
        return NULL;
    nleft = nleftplusright / 2;
    nright = nleftplusright - nleft;
    p = (node *)malloc(sizeof(node));
    p -> left = pbtree(nleft);
    scanf("%d", &p -> num);
    p -> right = pbtree(nright);
    return p;
}
  
```



(下書き用)