■設問１．アルゴリズムとデータ構造概説

次の文章を読んで，［ア］～［ケ］にあてはまる適切な語句を，以下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［ア］と呼ぶ．この［ア］に具体的なデータを加えることによりプログラムが完成する．また，使用す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［イ］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［ア］に他ならない．よって，［ア］と［イ］には，密接な関係がある．

データ群の表現において代表的なものにテーブルがあり，テーブルを構成する 1 件ずつのデータのことをレコードと呼ぶ．レコードはふつう［ウ］とデータの組み合わせにより構成される．この［ウ］について，昇順，あるいは降順にレコードを並べ替える処理のことをソートという．ソートは，［エ］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，［エ］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［エ］ソートのことを指す．

使用するメモリの量を考慮しなくてもよい，すなわち，無制限にメモリを使用できる理想的な環境にあるのであれば，より高速かつ簡便なソート手法が考えられる．しかしながら，メモリは，外部記憶装置に比べて圧倒的に［オ］であり，そのような環境の実現は，使用されるデータの範囲があらかじめ限定されているような特殊な場合を除いて，一般的には不可能である．よって，［エ］ソートにおいては，そのメモリ使用量を極力抑える必要がある．すなわち，メモリ使用量は最大でもデータ数 n に依存しない程度に抑えなければならない．

対象となる問題を処理する計算時間は短ければ短いほどよい．よって，ある問題に対して複数の［ア］が考えられる場合，その優劣は，その計算時間 T に依存する．ふつう，［ア］の性能は［カ］により評価され，$O(n)$ という形で表される．たとえば，計算時間 $T(3n^2+\sqrt{5}n+1)$ である［ア］の［カ］は $O($［キ］$)$ となる．

なお，現在の計算機では，プログラム格納方式が採用されており，その処理は逐次実行型，すなわち，［ク］の手順となっている．よって，現在の計算機において事実上計算可能な処理は，多項式時間での計算が可能なものに限られる．これに対して，［ケ］時間以上の処理を必要とする［ア］は，その処理が現実的には不可能となるので注意が必要である．

選択肢：
アル=フワリズミ，アルゴリズム，シナリオ，データ構造，ラーメン構造，ドア，キー，内部，中部，安価，高価，ステップ数，計算量，重量，1，3，n，$\sqrt{5}n$，$3n$，$\log n$，$n\log n$，n^2，$3n^2$，2^n，安定性，非安定性，決定性，非決定性，定数，指数，階乗

■設問２．ソート

９個のデータからなる次の配列について，以下の問いに答えなさい．（並べ替えはすべて昇順とする．）

配列：9, 6, 2, 7, 4, 1, 3, 5, 8

（１）以下の表は単純選択法を用いてソート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）に入る数字をそれぞれ答えなさい．

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

（２）以下の表はバブルソート法を用いてソート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）に入る数字をそれぞれ答えなさい．

(0)	9	6	2	7	4	1	3	5	8
(1)	9	6	2	7	1	4	3	5	8
(2)	9	6	2	1	7	4	3	5	8
(3)	9	6	1	2	7	4	3	5	8
(4)	9	1	6	2	7	4	3	5	8
(5)	1	9	6	2	7	4	3	5	8
(6)	1	9	6	2	7	3	4	5	8
(7)	1								
(8)	1								
(9)	1	(ア)							
(10)	1								
(11)	1								
(12)	1			(イ)			(ウ)		
(13)	1								
(14)	1								
(15)	1								
(16)	1								
(17)	1					(エ)			
(18)	1							(オ)	
(19)	1								
(20)	1	2	3	4	5	6	7	8	9

（3）クイックソート法を用いてソートすることを考える．以下の表は pivot を 4 としたときのすべての置換結果を表している．空欄をすべて埋め（ア）～（オ）に入る数字をそれぞれ答えなさい．また，このとき，置換が終了した後に並べ替えを行うべき左部分配列と右部分配列を答えなさい．

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

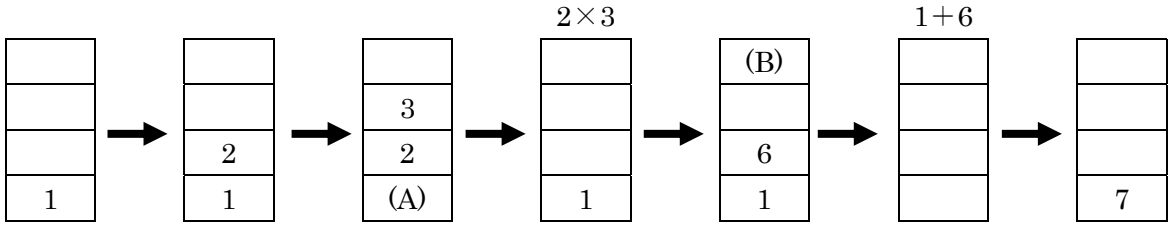
■設問３．スタック

以下の式を逆ポーランド記法で記せ．また，以下の図は，その計算をスタックにより実現した場合の計算過程における，プッシュ(push)，ポップ(pop)時のすべてのスタックの状態を表している．空欄を埋めスタックの各状態を完成させ，（ア）～（オ）に入る数字を，例にならって答えなさい．

数式：$6 - 5 \times (1 - 3) \div 4$

[例] $1 + 2 \times 3$

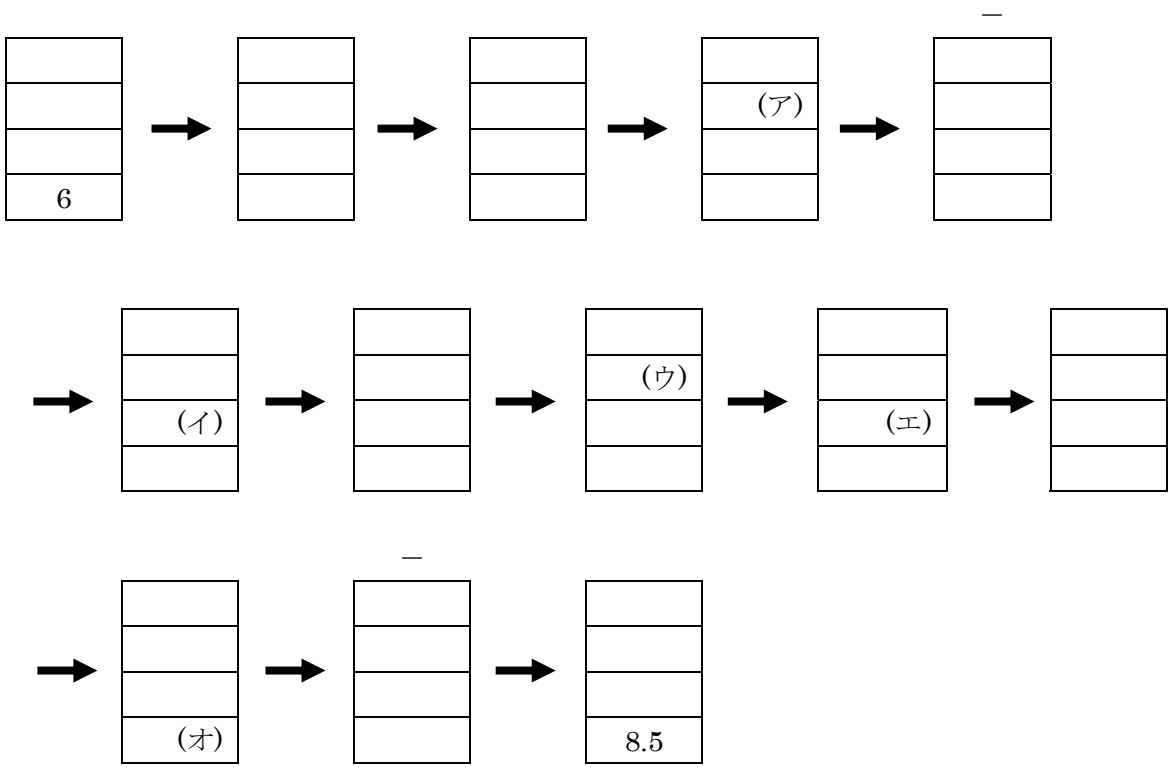
逆ポーランド記法：1 2 3 × +



※(A)には 1 が入るので「1」と答え，(B)は空欄なので「なし」と答える．

（下書き用）

逆ポーランド記法：



■設問４．線形連結リスト

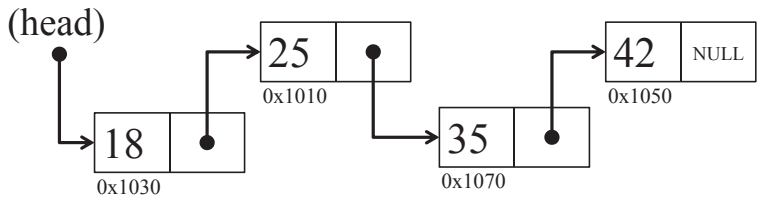
以下のデータを順番に読み込み，昇順に並ぶように線形連結リストに格納する．

25, 18, 42, 35, 20

データ 42 までを追加した後のメモリの状態を以下に示す．

address	value	next
—	(head)	0x1030
0x1010	25	0x1050
0x1030	18	0x1010
0x1050	42	NULL

これに対して，データ 35 を追加した後のメモリの状態に対応するセルの図を以下に示す．



上記にならい，以下の問いに答えなさい．

(1) データを追加する際には，以下の関数が実行される．

```
void insert_cell(struct cell **pointer, int new_value)
{
  struct cell *new_cell;
  new_cell = (struct cell *)malloc(sizeof(struct cell));
  new_cell->value = new_value;
  new_cell->next = *pointer; ①
  *pointer = new_cell; ②
}
```

データ 20 を追加する際における，上記①，②の各処理実行後に対応するメモリの状態の表を以下に記す．空欄を埋めて表を完成させ，（ア）～（エ）に当てはまる値をそれぞれ答えなさい．

①

address	value	next
——	(head)	0x1030
0x1010	25	0x1070
0x1030	18	0x10 （ア）
0x1050	42	NULL
0x1070	35	0x1050
0x1090	20	0x10 （イ）

②

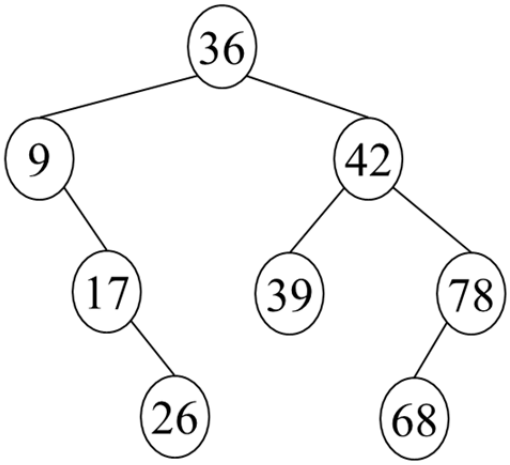
address	value	next
——	(head)	0x1030
0x1010	25	0x1070
0x1030	18	0x10 （ウ）
0x1050	42	NULL
0x1070	35	0x1050
0x1090	20	0x10 （エ）

(2) 上記に対して，データ 35 を削除した後におけるメモリの状態の表を以下に記す．空欄を埋めて表を完成させ，（ア）に当てはまる値を答えなさい．

address	value	next
——	(head)	0x1030
0x1010	25	0x10 （ア）
0x1030	18	0x10
0x1050	42	NULL
0x1070	–	——
0x1090	20	0x10

■設問５．二分探索木

以下の二分探索木に関して，次の問いに答えなさい.



（1） データ 39 を探索する際に走査するノードを答えなさい．例）9：36→9

（2） 行きがけ順，帰りがけ順にトラバーサルした結果をそれぞれ答えなさい.
例）通りがけ順：9, 17, 26, 36, 39, 42, 68, 78

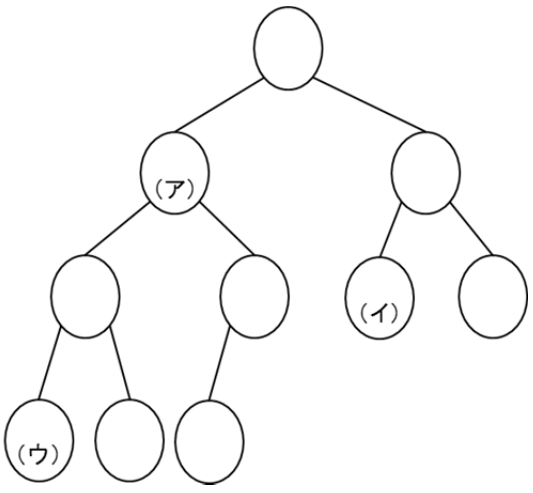
（3） データ 20 を追加したとき，生成される二分探索木を答えなさい.

■設問６．ヒープ

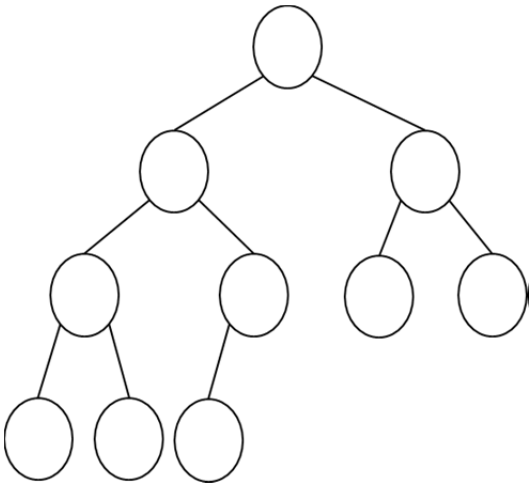
10 個のデータからなる次の配列について，以下の問いに答えなさい．ただし，当該配列において，0 番目の要素は使用せず，1 番目以降の要素として順に 10 個のデータが格納されているものとする.

6, 4, 10, 7, 5, 1, 2, 8, 9, 3

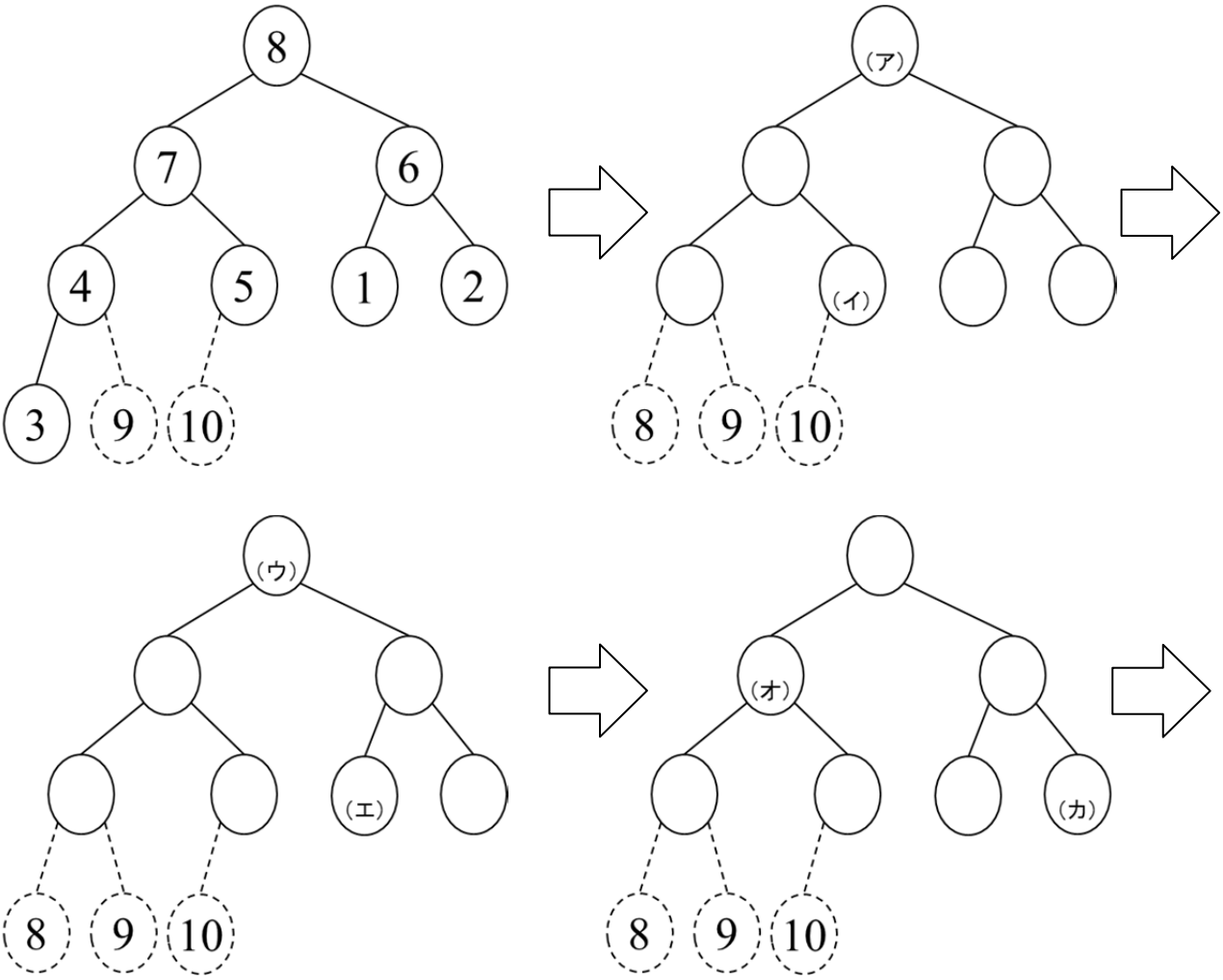
（1）以下の図は、上記配列に対応する完全二分木を表している．空欄になっている各ノードを適当な数値で埋め，（ア）～（ウ）に入る数値をそれぞれ答えなさい.

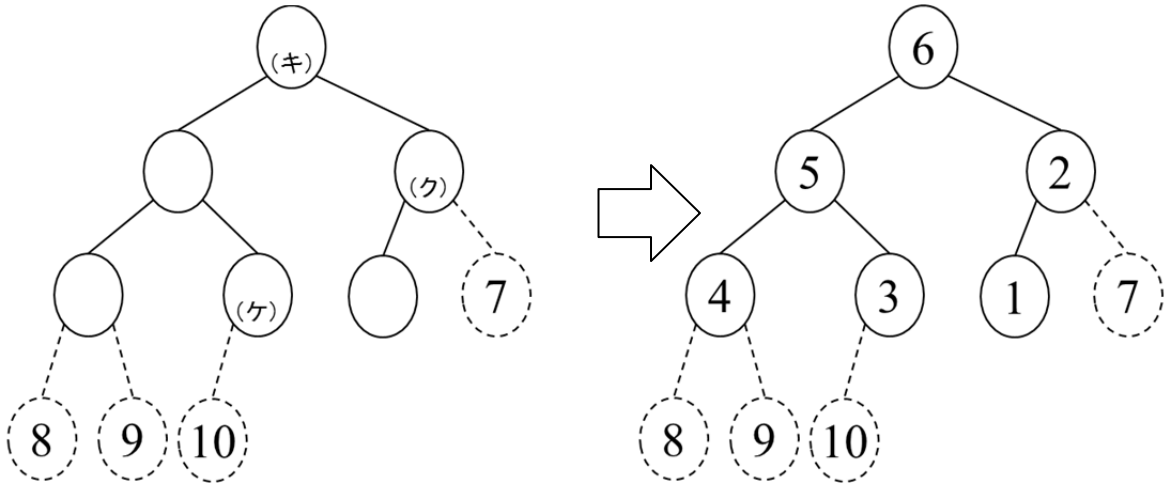


（2）上記（1）の完全二分木を下降修復の繰り返しによりヒープ化した結果を配列の形で答えなさい．　　（下書き用）



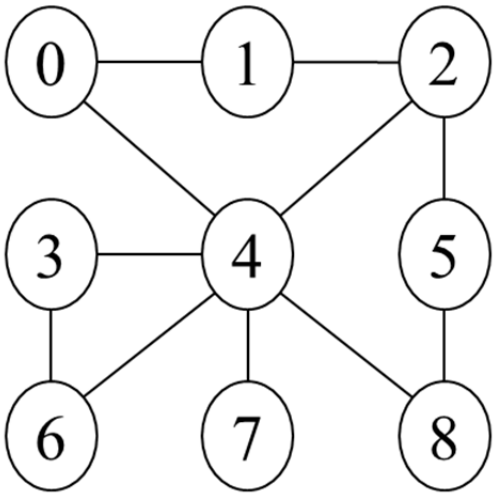
（3）上記（2）で得られたヒープに対して，ヒープソートによりデータを昇順に整列することを考える．このときの途中経過として，5 回分の置換結果を以下に示している．空欄を埋めて図を完成させ，（ア）～（ケ）に入る数値をそれぞれ答えなさい．なお，点線で記したノードは値が確定していることを表している．





■設問７．グラフ探索

以下は，ノード間の重みがすべて一定の無向グラフである．このグラフについて，ノード 0 から到達可能なすべてのノードを探索する．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には，番号が小さいもの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．

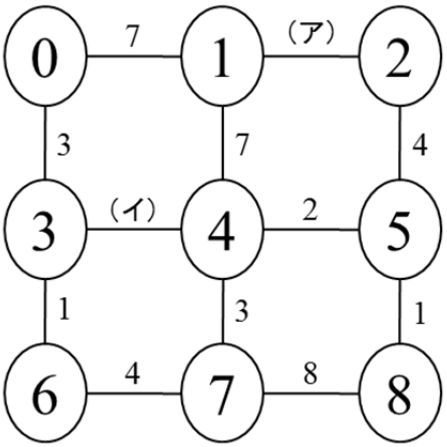
(0, 1) (　,　) (　,　) (4,　3) (　,　) (　,　) (　,　) (　,　)

（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．

(0, 1) (　,　) (　,　) (4,　3) (　,　) (　,　) (　,　) (　,　)

■設問８．最短経路問題

（１）以下は，ノード間の各経路に重みが付与された，重みつきの無向グラフと，これに対応する隣接行列である．空欄（ア）～（エ）を埋め，それぞれ完成させなさい．



	0	1	2	3	4	5	6	7	8
0	∞	(ウ)	∞	3	∞	∞	∞	∞	∞
1	(ウ)	∞	4	∞	7	∞	∞	∞	∞
2	∞	4	∞	∞	∞	4	∞	∞	∞
3	3	∞	∞	∞	9	∞	1	∞	∞
4	∞	7	∞	9	∞	(エ)	∞	3	∞
5	∞	∞	4	∞	(エ)	∞	∞	∞	1
6	∞	∞	∞	1	∞	∞	∞	4	∞
7	∞	∞	∞	∞	3	∞	4	∞	8
8	∞	∞	∞	∞	∞	1	∞	8	∞

（２）上記（１）のグラフについて，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．以下に従い，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の dist，flag，path の値をノード毎に全て順に答えなさい．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には，番号の小さいものが優先されるものとする．下の空欄を埋め（ア）～（オ）に入る記号・数字を答えなさい．

	0	1	2	3	4	5	6	7	8
node u = [　0　]									
dist:	0	7	∞	3	∞	∞	∞	∞	∞
flag:	1	0	0	0	0	0	0	0	0
path:	EOP	0	0	0	0	0	0	0	0
node u = [　3　]									
dist:	0	7	∞	3	12	∞	4	∞	∞
flag:	1	0	0	1	0	0	0	0	0
path:	EOP	0	0	0	3	0	3	0	0
node u = [　6　]									
dist:	0	7	∞	3	12	∞			∞
flag:	1	0	0	1	0	0	(ア)		0
path:	EOP	0	0	0	3	0			0
node u = [　　]									
dist:	0			3		∞			∞
flag:	1			1		0			0
path:	EOP	(イ)		0		0			0

	0	1	2	3	4	5	6	7	8
node u = [　　]									
dist:	0			3		∞			
flag:	1			1		0			
path:	EOP			0	(ウ)	0			
node u = [　　]									
dist:	0		(エ)	3					
flag:	1			1					
path:	EOP			0					
node u = [　　]									
dist:	0			3					
flag:	1			1					
path:	EOP			0					
node u = [　　]									
dist:	0			3					
flag:	1			1					0
path:	EOP			0					(オ)
node u = [　8　]									
dist:	0			3					
flag:	1			1					1
path:	EOP			0					

（3）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