

# ダイクストラ法の実装

静岡大学工学部  
安藤 和敏

浜松工業高校出張授業

2008.11.19

## 目次

1. 最短経路問題
2. ダイクストラ法(練習1)
3. 演習1
4. ダイクストラ法(練習2)
5. 演習2
6. C言語による実装
7. 演習3
8. 演習4
9. おわりに

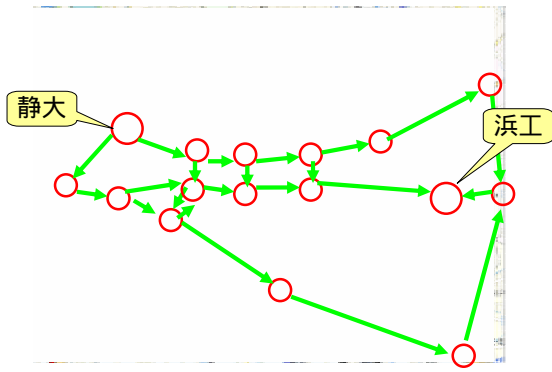
## 目次

1. 最短経路問題
2. ダイクストラ法(練習1)
3. 演習1
4. ダイクストラ法(練習2)
5. 演習2
6. C言語による実装
7. 演習3
8. 演習4
9. おわりに

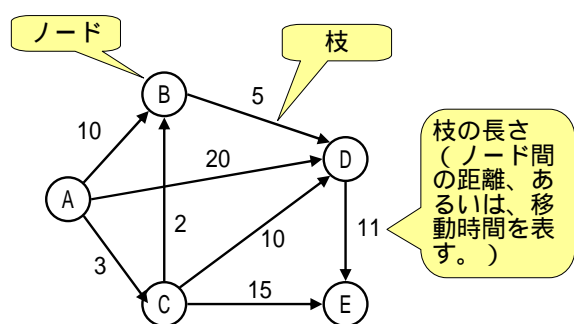
## 静大から浜工への最短経路は？



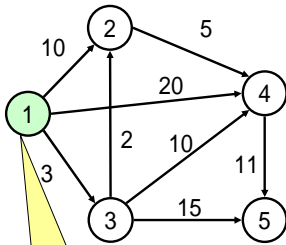
## 道路網のグラフとしての表現



## グラフ



## 最短経路問題



始点ノード  
(出発点)

グラフと始点ノードが与えられたとき、始点ノードから他の全てのノードへの最短経路を見つける問題

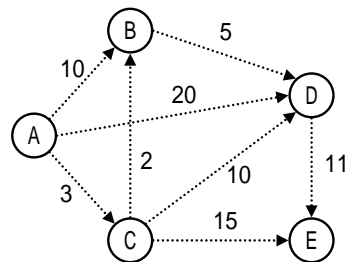
## 目次

1. 最短経路問題
2. ダイクストラ法 (練習1)
3. 演習1
4. ダイクストラ法 (練習2)
5. 演習2
6. C言語による実装
7. 演習3
8. 演習4
9. おわりに

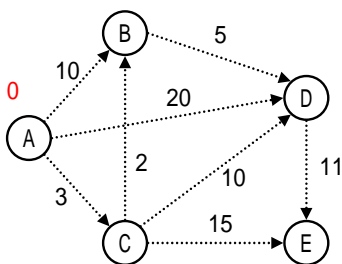
## ダイクストラ法

1. 全てのノードの距離を  $\infty$  とする。
2. 始点ノードの距離を0とする。
3. 以下の A~C をノードの数だけ繰り返す。
  - A) 距離が確定していないノードの中でノードの距離が最小のノードを  $n$  とする。
  - B)  $n$  の距離を確定する。
  - C)  $n$  から出る全ての枝  $nk$  に対して、以下を行う。  
 (\*) もし ( $k$  の距離  $>$   $n$  の距離 +  $nk$  の長さ)、かつ、 $k$  の距離が確定していないならば、 $k$  の距離 =  $n$  の距離 +  $nk$  の長さとし、さらに、 $k$  の前の点を  $n$  とする。

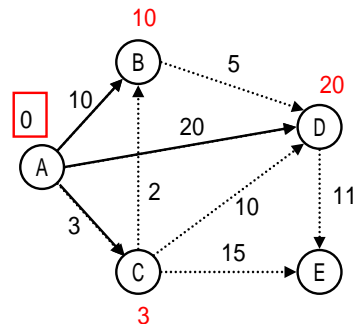
## ダイクストラ法 (1)



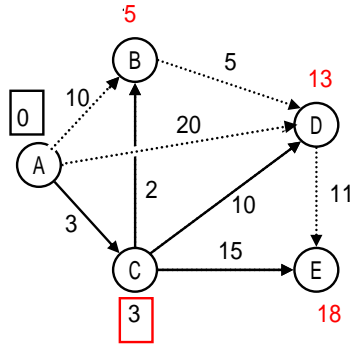
## ダイクストラ法 (2)



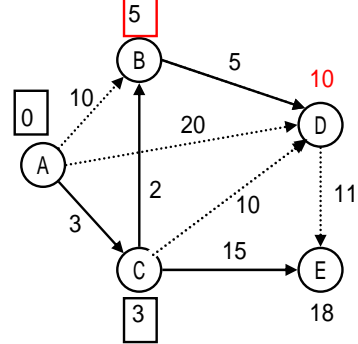
## ダイクストラ法 (3) (1回目)



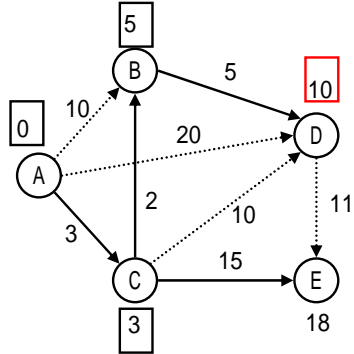
### ダイクストラ法 ( 3 ) ( 2回目 )



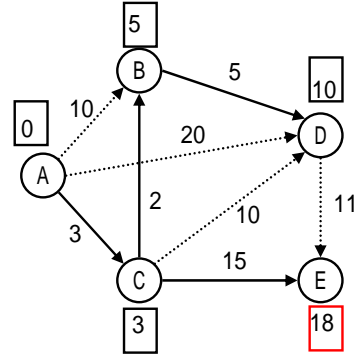
### ダイクストラ法 ( 3 ) ( 3回目 )



### ダイクストラ法 ( 3 ) ( 4回目 )



### ダイクストラ法 ( 3 ) ( 5回目 )



### 目次

1. 最短経路問題
2. ダイクストラ法 (練習1)
3. 演習1
4. ダイクストラ法 (練習2)
5. 演習2
6. C言語による実装
7. 演習3
8. 演習4
9. おわりに

### 目次

1. 最短経路問題
2. ダイクストラ法 (練習1)
3. 演習1
4. ダイクストラ法 (練習2)
5. 演習2
6. C言語による実装
7. 演習3
8. 演習4
9. おわりに

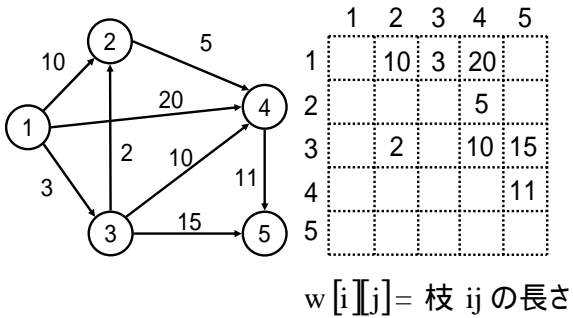
### 使用する変数

変数名	意味
start	始点ノード
U[ ]	各ノードの距離が未確定かどうかを表す配列。 U[ k ] == 0 のとき確定で、 U[ k ] == 1 のとき未確定。
dist [ ]	各ノードの距離を表す配列
prev[ ]	各ノードの前のノードを表す配列

### 使用する変数

変数名	意味
w[ ][ ]	入力グラフを表現する2次元配列
datasize	グラフのノード数

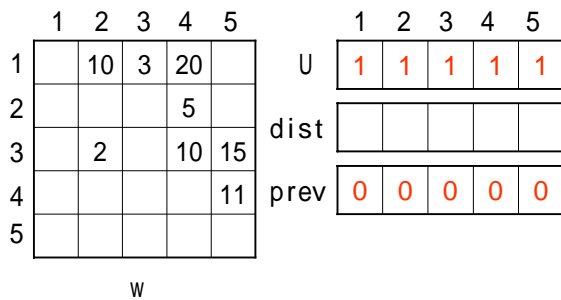
### グラフの2次元配列wによる表現



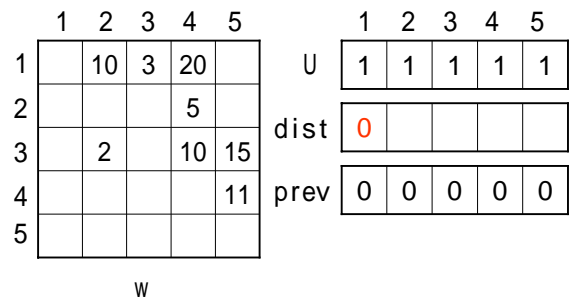
### ダイクストラ法

- 全てのノード k について、 $distance[k] = \infty$ 、 $U[k] = 1$ 、 $prev[k] = 0$  とする。
- $distance[start] = 0$  とする。
- 以下の A ~ C を  $datasize$  回繰り返す。
  - $U[k] = 1$  であるノードの中で  $dist[k]$  が最小のノードを n とする。
  - $U[n] = 0$  とする。(n の距離を確定する。)
  - 全てのノード k について、以下を行う。  
(\* もし  $(dist[k] > dist[n] + w[n][k])$ 、かつ、 $U[k] = 1$  ならば、  
 $dist[k] = dist[n] + w[n][k]$ 、  
 $prev[k] = n$  とする。

### ダイクストラ法 ( 1 )



### ダイクストラ法 ( 2 )



### ダイクストラ法(3-A,B) (1回目)

	1	2	3	4	5	
1		10	3	20		
2				5		
3		2		10	15	
4					11	
5						

w

	n				
	1	2	3	4	5
U	0	1	1	1	1
dist	0				
prev	0	0	0	0	0

### ダイクストラ法(3-C) (1回目)

		k	k	k		
	1	2	3	4	5	
n 1		10	3	20		
2				5		
3		2		10	15	
4					11	
5						

w

	1	2	3	4	5
U	0	1	1	1	1
dist	0	10	3	20	
prev	0	1	1	1	0

### ダイクストラ法(3-A,B) (2回目)

	1	2	3	4	5	
1		10	3	20		
2				5		
3		2		10	15	
4					11	
5						

w

	n				
	1	2	3	4	5
U	0	1	0	1	1
dist	0	10	3	20	
prev	0	1	1	1	0

### ダイクストラ法(3-C) (2回目)

		k	k	k		
	1	2	3	4	5	
n 1		10	3	20		
2				5		
3		2		10	15	
4					11	
5						

w

	1	2	3	4	5
U	0	1	0	1	1
dist	0	5	3	13	18
prev	0	3	1	3	3

### ダイクストラ法(3-A,B) (3回目)

	1	2	3	4	5	
1		10	3	20		
2				5		
3		2		10	15	
4					11	
5						

w

	n				
	1	2	3	4	5
U	0	0	0	1	1
dist	0	5	3	13	18
prev	0	3	1	3	3

### ダイクストラ法(3-C) (3回目)

		k				
	1	2	3	4	5	
n 1		10	3	20		
2				5		
3		2		10	15	
4					11	
5						

w

	1	2	3	4	5
U	0	0	0	1	1
dist	0	5	3	10	18
prev	0	3	1	2	3

### ダイクストラ法(3-A,B) (4回目)

	1	2	3	4	5
1		10	3	20	
2				5	
3		2		10	15
4					11
5					

w

	1	2	3	4	5
U	0	0	0	0	1
dist	0	5	3	10	18
prev	0	3	1	2	3

### ダイクストラ法(3-C) (4回目)

	1	2	3	4	5
1		10	3	20	
2				5	
3		2		10	15
4					11
5					

w

	1	2	3	4	5
U	0	0	0	0	1
dist	0	5	3	10	18
prev	0	3	1	2	3

### ダイクストラ法(3-A,B) (5回目)

	1	2	3	4	5
1		10	3	20	
2				5	
3		2		10	15
4					11
5					

w

	1	2	3	4	5
U	0	0	0	0	0
dist	0	5	3	10	18
prev	0	3	1	2	3

### ダイクストラ法(3-C) (5回目)

	1	2	3	4	5
1		10	3	20	
2				5	
3		2		10	15
4					11
5					

w

	1	2	3	4	5
U	0	0	0	0	0
dist	0	5	3	10	18
prev	0	3	1	2	3

## 目次

1. 最短経路問題
2. ダイクストラ法(練習1)
3. 演習1
4. ダイクストラ法(練習2)
5. 演習2
6. C言語による実装
7. 演習3
8. 演習4
9. おわりに

## 目次

1. 最短経路問題
2. ダイクストラ法(練習1)
3. 演習1
4. ダイクストラ法(練習2)
5. 演習2
6. C言語による実装
7. 演習3
8. 演習4
9. おわりに

## プログラムの断片

<http://coconut.sys.eng.shizuoka.ac.jp/hamako08/>  
からプログラムの断片  
dijkstra.c  
と入力ファイル  
data0.txt, data1.txt, data2.txt  
をダウンロードせよ。

## 使用する定数

定数名	値	意味
N_NODE	100	入力グラフのサイズの最大数
INF	100000	(十分に大きい数)

## 入力ファイル

data0.txt

```
datasize 5
100000 10 3 20 100000
100000 100000 100000 5 100000
100000 2 100000 10 15
100000 100000 100000 100000 11
100000 100000 100000 100000 100000
```

## ファイルからデータの読み込み

```
void read_graph(char filename[],
int *datasize_p,
int w[N_NODE][N_NODE])
```

ファイル filename からデータを読み、  
datasize\_pと2次元配列 w[][] に格納する。  
(この関数の中身を理解しなくても良い。)

## main関数

```
main(int argc, char *argv[]){
    int i,j;
    int w[N_NODE][N_NODE];
    int datasize;
    int dist[N_NODE];
    int prev[N_NODE];

    read_graph(argv[1], &datasize, w);

    /* w の表示 (省略)*/

    dijkstra(1, datasize, w, dist, prev);

    /*
    show_path(1, datasize, dist, prev);
    */
}
```

## 関数dijkstra

```
void dijkstra(int start,
             int datasi ze,
             int w[N_NODE][N_NODE],
             int dist[N_NODE],
             int prev[N_NODE]
            ) {
    int i, n, k;
    int U[N_NODE];
    int minVal ue; (次ページに続く)
```

## 関数dijkstra

```
(前ページからの続き)
/* 初期化 */(省略)

/* 始点ノードの距離を 0 とする */
dist[start] = 0;

for(i=1; i<=datasi ze; i++) {
    /* (A) 距離が最小のノード n を見つける */

    /* (B) */
    U[n]=0;

    /* (C) dist と prev を更新する */

    /* U, dist, prev の表示 */(省略)
}
}
```

## 目次

1. 最短経路問題
2. ダイクストラ法(練習1)
3. 演習1
4. ダイクストラ法(練習2)
5. 演習2
6. C言語による実装
7. 演習3
8. 演習4
9. おわりに

## 演習3

<http://coconut.sys.eng.shizuoka.ac.jp/Hamako08/>

関数 dijkstra を完成させて、入力ファイルを与えて実行させてみよう。

## 目次

1. 最短経路問題
2. ダイクストラ法(練習1)
3. 演習1
4. ダイクストラ法(練習2)
5. 演習2
6. C言語による実装
7. 演習3
8. 演習4
9. おわりに

## 演習4

関数 show\_path を作成して、終点ノード= datasi ze から 始点ノード 1 までの経路を表示させよ。

(もし、余力があるようであれば、始点ノード=1 から終点ノード=datasi ze までの経路を表示させるようにせよ。)



## 目次

1. 最短路問題
2. ダイクストラ法(練習1)
3. 演習1
4. ダイクストラ法(練習2)
5. 演習2
6. C言語による実装
7. 演習3
8. 演習4
9. おわりに

## おわりに

- 本日で示した実装ではグラフを2次元配列を使用して表現した。この表現方法には無駄がある(dist と prev の更新には無駄がある)が、もっと良い表現方法もある。
- 距離が最小のノードを発見する方法も、もっと良い方法もある。(ヒープ、フィボナッチヒープのようなデータ構造)
- どんなアルゴリズムでも、データの表現方法の工夫が高速な実装に結びつくことが多い。