

Project

วิชา Data Structure and Algorithms

523231

กลุ่ม G8

หัวข้อ 2C

ผู้จัดทำ

1. นางสาวเอกปวีร์	อุ้นภักดี	รหัส B6210236	เลขที่30	เบอร์โทรศัพท์ 0908208456
2. นายเฉลิมเกียรติ	คงกะพัน	รหัส B6214562	เลขที่ 43	เบอร์โทรศัพท์ 0930101275
3. นางสาวเพชรณสร	พงษ์ศิริยะกุล	รหัส B6215415	เลขที่ 47	เบอร์โทรศัพท์ 0823642199
4. นางสาวสุชาวดี	เที่ยงตรง	รหัส B6238124	เลขที่ 103	เบอร์โทรศัพท์ 0878807717

User manual

โปรแกรมหา Shortest path ของกราฟแบบมีน้ำหนักโดยใช้ Dijkstra's Algorithm

ขั้นตอนการทำงานของโปรแกรม

1. เข้าสู่โปรแกรมและโปรแกรมจะให้ทำการใส่จำนวน node ของกราฟ
2. โปรแกรมจะให้ค่าน้ำหนักจาก node หนึ่งไปยัง node หนึ่ง ซึ่งเราสามารถดูค่าน้ำหนักได้จากกราฟที่เราต้องการหา

หมายเหตุ การป้อนค่าน้ำหนักจะต้องทำการใส่เครื่องหมายลูกน้ำ (,) คั่นในแต่ละค่า

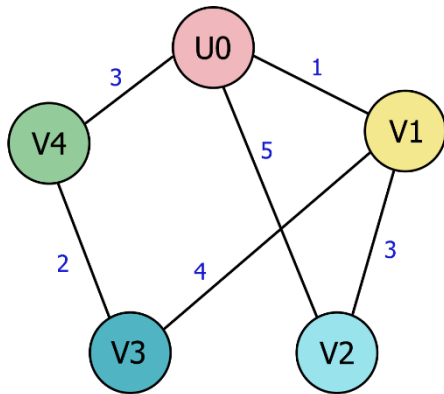
2.1) หากทั้งสอง node มี edge เชื่อมถึงกัน ให้เราใส่ค่าน้ำหนักระหว่าง node

2.2) หากทั้งสอง node ไม่มี edge เชื่อมถึงกัน ให้เราใส่ค่า -1

2.3) ป้อนค่าน้ำหนักทั้งหมดลงไปโดยมีเครื่องหมาย , คั่นค่าน้ำหนักไว้ เมื่อครบทุก node แล้วหน้าจอจะแสดงขั้นตอนวิธีและผลลัพธ์

3. จบการทำงานของโปรแกรม

Ex.



$$\begin{bmatrix} 0 & 1 & 5 & -1 & 3 \\ 1 & 0 & 3 & 4 & -1 \\ 5 & 3 & 0 & -1 & -1 \\ -1 & 4 & -1 & 0 & 2 \\ 3 & -1 & -1 & 2 & 0 \end{bmatrix}$$

(ภาพประกอบโจทย์)

ตัวอย่างการทำงานของโปรแกรม

Input number of node : 5

Node u0 : 0,1,5,-1,3

Node v1 : 1,0,3,4,-1

Node v2 : 5,3,0,-1,-1

Node v3 : -1,4,-1,0,2

Node v4 : 3,-1,-1,2,0

l(u0)	v1	v2	v3	v4	Add to S
0	(INF,-)	(INF,-)	(INF,-)	(INF,-)	u0
	(1,u0)	(5,u0)	(INF,-)	(3,u0)	v1
		(4,v1)	(5,v1)	(3,u0)	v4
		(4,v1)	(5,v1)		v2
			(5,v1)		v3

Path : u0->v1 , u0->v4 , v1->v2 , v1->v3

ส่วนของการอธิบาย Code

(รูปภาพประกอบสามารถดูได้ด้านล่างหรือในไฟล์ชื่อ File code)

Class Main

บรรทัด	คำอธิบาย
6-11	ประกาศตัวแปรชื่อ sc รับค่าจากแป้นพิมพ์ สร้าง object จากคลาส Class ประกาศตัวแปรประเภท integer ชื่อ n ตัวแปร array2 มิติชื่อ weight ของข้อมูลประเภทinteger ประกาศตัวแปรarrayชื่อtempของข้อมูลประเภทstring
12-13	ประกาศตัวแปร array ชื่อว่า name ของข้อมูลประเภท string มีขนาดเท่ากับจำนวน node ที่รับเข้ามาเพื่อเก็บค่าชื่อของ node โดย node เริ่มต้นนั้นจะมีชื่อว่า u0 เสมอ
14-16	วน loop for เพื่อกำหนดชื่อของ node ที่เหลือ ถ้าหากมี 5 node จะได้ชื่อ 4 node ดังนี้ v1 คือชื่อ node ที่ 2 , v2 คือชื่อ node ที่ 3 , v3 คือชื่อ node ที่ 4 , v4 คือชื่อ node ที่ 5
17-25	<p>รับค่าน้ำหนักของแต่ละ node ดังตัวอย่างด้านบนคือ</p> <p>node u0 ไปที่ u0 (ตัวเอง) จะให้ค่าเป็น 0</p> <p>u0 ไป v1 มีค่าน้ำหนัก = 1</p> <p>u0 ไป v2 มีค่าน้ำหนัก = 5</p> <p>u0 ไม่มี edge เชื่อมไปที่ v3 ดังนั้นจะให้ค่าน้ำหนัก = -1</p> <p>u0 ไป v4 มีค่าน้ำหนัก = 3</p> <p>จากข้อมูลด้านบนทำให้เราจะใส่ค่าของ node0 ว่า 0,1,5,-1,3 โดยใช้ .split(",") ในการตัดแบ่งข้อความแล้วจัดเก็บค่าที่ได้เป็น array จากนั้นนำค่าที่ได้วนลูปเพื่อแทนค่านั้นลงไป array2 มิติชื่อว่า weight โดยค่าที่ได้นั้นจะแทนไปในแต่ละ column เมื่อครบทุกตัวแล้วจะให้ temp=null เพื่อให้ array ของตัวแปร temp กลับมาว่างอีกครั้งเพื่อรับค่าใหม่ จากนั้นจึงกลับไปทำ for loop แรกเพื่อรับค่าน้ำหนักของ node ต่อไป</p>
26	print output ออกเพื่อค้นส่วนของวิธีทำและผลลัพธ์ออกจากขั้นตอนการรับค่า
27-28	ประกาศตัวแปรประกาศตัวแปร array ชื่อ visit ของข้อมูลประเภท integer มีขนาดเท่ากับจำนวนของ node ที่รับเข้ามา ซึ่งตัวแปรนี้ใช้ในการเช็คค่า node ใดที่ถูกจัดเก็บเข้า stack แล้วถ้ามีค่าเท่ากับ 1 คืออยู่ใน stack แล้ว ถ้ามีค่าเท่ากับ 0 คือยังไม่อยู่ใน stack และกำหนดให้ visit[0] = 1 เพราะ node u0 เป็น node เริ่มต้น จึงจัดเก็บเข้า stack แล้วเสมอ
30-31	ประกาศตัวแปรarrayชื่อว่า stack ของข้อมูลประเภท integer มีขนาดเท่ากับจำนวนnodeที่รับเข้ามา โดยให้ stack[0] = 0

บรรทัด	คำอธิบาย
32-33	ประกาศตัวแปร array ชื่อว่า stackw ของข้อมูลประเภท integer มีขนาดเท่ากับจำนวน node ที่รับเข้ามา ทำหน้าที่เก็บสะสมค่าน้ำหนักของ node ที่ต้องการหาไปจนถึง node เริ่มต้น และกำหนดให้ $stackw[0] = 0$ คือ node เริ่มต้นนั้นมีน้ำหนักเท่ากับ 0
34-36	ประกาศตัวแปรประเภท integer ชื่อว่า minw , minad และ minadwint โดยกำหนดให้ minw มีค่าเท่ากับ -1 , minad มีค่าเท่ากับ 0 และ minadwint มีค่าเท่ากับ 0
37	ประกาศตัวแปร array ชื่อว่า minadw ของข้อมูลประเภท integer มีขนาดเท่ากับจำนวน node ที่รับเข้ามา ไว้เก็บ node ที่ node(j) แต่ละตัวเชื่อมอยู่
38	ประกาศตัวแปร array ชื่อว่า path ของข้อมูลประเภท string มีขนาดเท่ากับจำนวน node ที่รับเข้ามา โดยทำหน้าที่เก็บค่าว่า node ที่เราต้องการนั้นมีเส้นทางผ่านมาจาก node ไດ
40-44	Print บรรทัดแรกของส่วนวิธีทำ ใช้ for loop เพื่อ print ชื่อ node ตัวที่สองขึ้นไป
45-49	Print บรรทัดที่สองของส่วนวิธีทำ ใช้ for loop สำหรับ node ที่สองขึ้นไปเพื่อ print ค่าน้ำหนักและเส้นทางที่ได้คือ (INFI,-) เพราะในขั้นตอนนี้จะพบแค่ node เริ่มต้นคือ u0 ทำให้ node อื่น ๆ จะมีค่าน้ำหนักเป็น infinity และยังไม่มีเส้นทาง
51-125	ใช้ for loop(int i) เพื่อเข้าโปรแกรมและแสดงผลวิธีทำจนทุก node ถูกเก็บเข้าไปในสแตก
52	Print ช่องว่างที่ตรงกับ u0 เพราะ u0 ถูกเก็บเข้าสแตกแล้ว จึงไม่ต้องทำต่อ
53	ประกาศตัวแปร array ชื่อว่า w ของข้อมูลประเภท integer มีขนาดเท่ากับจำนวน node ที่รับค่าเข้ามา ไว้เก็บค่าที่ node(j) ต่อกับทุกตัวในสแตก
54	ประกาศตัวแปร array ชื่อว่า wnode ของข้อมูลประเภท integer มีขนาดเท่ากับจำนวน Node ที่รับค่าเข้ามา ไว้เก็บค่าน้ำหนักที่น้อยที่สุดของแต่ละ node
56-59	ใช้ for loop (int j) เพื่อเข้าถึงตำแหน่งของ node หาก node นั้นถูกจัดเก็บเข้าสแตกแล้วจะเข้าเงื่อนไข if ซึ่งจะ print ช่องว่างออกมาเพราะ node นั้นอยู่ในสแตกแล้วไม่ต้องทำต่อ หากยังไม่อยู่ในสแตกจะเข้าเงื่อนไข else if
59-61	เมื่อพบว่า node นั้นยังไม่อยู่ในสแตกจะเข้า for loop (int k) เพื่อหาค่าในแต่ละเส้นทางของ Node นั้น ถ้า $i > 0$ จะเข้าไปทำในเงื่อนไข if
62-63	ถ้ามี edge เชื่อมระหว่าง node(j) กับ node(k) จะเข้าไปทำในเงื่อนไข if คือค่าของ w ณ ตำแหน่งที่ k จะมีค่าเท่ากับ ค่าน้ำหนักของ edge ที่เชื่อมระหว่าง node(j) กับ node(k) + ค่าน้ำหนักสะสมของ node(k) ที่เชื่อมไปถึง node เริ่มต้น
64-65	ถ้าไม่มี edge เชื่อมระหว่าง node(j) กับ node(k) จะเข้าไปทำในเงื่อนไข else if คือค่าของ w ณ ตำแหน่งที่ k จะมีค่าเท่ากับ -1 คือ infinity
69	ถ้า node นั้นยังไม่อยู่ในสแตกและ $i = 0$ จะเข้าไปทำในเงื่อนไข if
70-72	ใช้ for loop (int k) เพื่อเก็บค่าน้ำหนักของ node(k) ที่ต่อกับ node เริ่มต้น (u0) ใน w[k]

บรรทัด	คำอธิบาย
74	ให้ตัวแปร integer minw = -1 เพื่อรีเซ็ตค่า
75	ประกาศตัวแปร boolean ชื่อว่า chk ให้มีค่าเป็น false
76	ใช้ for loop (int l) เพื่อหาค่า minw ที่ไม่เท่ากับ -1 เพื่อใช้ในการเปรียบเทียบต่อไป
77-81	ถ้าค่า w ณ ตำแหน่งที่ l มีค่ามากกว่า -1 และ node(j) ยังไม่ถูกเก็บเข้าสแตก จะเข้าไปทำในเงื่อนไข if คือให้ chk = true และให้ minw มีค่าเท่ากับค่าจาก array w ตัวสุดท้ายที่ไม่ใช่ -1
84	ถ้า chk = true นั่นคือใน array w มีค่าที่ไม่เท่ากับ -1 อยู่
85	ใช้ for loop (int l) เพื่อนำค่าจาก array w ออกมา
86	ถ้า node(j) ยังไม่ถูกเก็บเข้าสแตก จะเข้าไปทำในเงื่อนไข if
87	ถ้าค่า l น้อยกว่า minw และ l ไม่เท่ากับ -1 จะเข้าไปทำในเงื่อนไข if คือให้ minw มีค่าเท่ากับ l นั้น
94	ให้ค่าใน array wnode ณ ตำแหน่งที่ j มีค่าเท่ากับ minw นั่นคือให้ wnode[j] เก็บค่าน้อยที่สุด
95	หาดำแหน่งของ node ที่ node(j) เชื่อมต่ออยู่ โดยการหาดำแหน่งของ minw ใน w[] นั้น จะต้องส่งค่า minw และ w เข้าไปใน c.findindex(minw , w);
96-97	ถ้า wnode ณ ที่ j มีค่าเท่ากับ -1 คือ ถ้า wnode ที่ node(j) ไม่เชื่อมต่อกับตัวใดในสแตกเลย จะ output ว่า (INFI, -)
98-99	ถ้า wnode ที่ node(j) มีการเชื่อมต่อกับตัวในสแตก จะ output ว่า (ค่าน้ำหนักที่น้อยที่สุดของ node(j) ไปจนถึง node เริ่มต้น , ชื่อของ node ที่เชื่อมอยู่กับ node(j) ที่ทำให้ได้ค่าน้ำหนักที่น้อยที่สุด)
104	ใช้ for loop (int q) เพื่อหาค่า minn ที่ไม่เท่ากับ -1 เพื่อนำมาใช้เปรียบเทียบ
105-107	ถ้าค่าของ wnode[q] ไม่เท่ากับ -1 และ node(q) ยังไม่อยู่ในสแตก ให้ส่งค่า wnode[q] เข้าไปเก็บไว้ที่ c.minn(wnode[q]) นั่นคือ c.minn(); จะเก็บค่าจาก array wnode ตัวสุดท้ายที่ไม่เป็น -1
109	กำหนดให้ integer minn รับค่าจาก c.getminn();
110	ใช้ for loop (int q) เพื่อหาค่าน้อยที่สุดในรอบนั้น ๆ
111-113	ถ้าค่าของ wnode[q] น้อยกว่าค่าของ c.getminn(); และ ค่าของ wnode[q] ไม่เท่ากับ -1 และ node(q) ยังไม่อยู่ในสแตก ให้ส่งค่า wnode[q] ไปเก็บไว้ใน c.minn();
116	หาดำแหน่งของค่า c.getminn(); ใน wnode[] โดยการส่งค่าเข้าไปใน c.findindex(c.getminn() , wnode); และ minad จะเท่ากับค่าที่หาได้

บรรทัด	คำอธิบาย
117	ให้ minadwint เก็บค่าตำแหน่งของ node ที่ node(minad) เชื่อมอยู่
118	จัดเก็บ node(minad) เข้าสแตก
119	ให้ stack[i+1] มีค่าเท่ากับ minad
120	output ชื่อของ node ที่ถูกจัดเก็บเข้าสแตก
122	ค่าน้ำหนักของ node ที่ถูกจัดเก็บเข้าสแตกจะเท่ากับ (ค่าน้ำหนักสะสมของ node ที่เชื่อมอยู่กับ node(minad) + ค่าน้ำหนักของnode(minad) ไป node ที่เชื่อมกันอยู่)
123	ให้ path[i] เก็บชื่อเส้นทางของnodeที่ถูกนำเข้าสู่สแตกแล้ว
124	output ให้ชั้นบรรทัดต่อไปเพื่อแสดงผลวิธีทำขั้นต่อไป
127-132	แสดงผลลัพธ์เส้นทางของแต่ละ node และจบการทำงานของโปรแกรม

Class Class

บรรทัด	คำอธิบาย
4-6	ประกาศตัวแปรแบบ private ชื่อว่า minn , minad , min เป็นตัวแปรประเภท integer
8-11	method min รับค่า integer เข้ามาแล้วเซตค่า
13-16	method getmin ใช้ส่งค่า min ที่เซตแล้วกลับไป
18-21	method minn รับค่า integer เข้ามาแล้วเซตค่า
23-26	method getminn ใช้ส่งค่า minn ที่เซตแล้วกลับไป
28-31	method minad รับค่า integer เข้ามาแล้วเซตค่า
33-36	method getminn ใช้ส่งค่า minad ที่เซตแล้วกลับไป
38-50	method findindex(int m, int[] n); ใช้หาตำแหน่ง
39	ประกาศตัวแปรประเภท integer ชื่อ index โดยกำหนดให้มีค่าเท่ากับ 0
40	ประกาศตัวแปรประเภท integer ชื่อ i โดยกำหนดให้มีค่าเท่ากับ 0
41	ประกาศตัวแปรประเภท Boolean ชื่อ chk โดยกำหนดให้ chk มีค่าเป็น false

Class Main

```
1 package Group8.MainClass;
2 import java.util.*;
3 public class Main {
4
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         Class c = new Class();
8         System.out.print("Input number of node : ");
9         int n = sc.nextInt();
10        int[][] weight = new int[n][n];
11        String[] temp = new String[n];
12        String[] name = new String[n];
13        name[0] = "u0";
14        for(int y=1;y<n;y++){
15            name[y] = "v"+y;
16        }
17        for(int y=0;y<n;y++){
18            System.out.print("Node "+name[y]+" : ");
19            String node = sc.next();
20            temp = node.split(",");
21            for(int z=0;z<n;z++){
22                weight[y][z] = Integer.valueOf(temp[z]);
23            }
24            temp = null;
25        }
26        System.out.println("\n
-----");
27        int visit[] = new int[n];
28        visit[0] = 1;
29
30        int stack[] = new int[n];
31        stack[0] = 0;
32        int stackw[] = new int[n];
33        stackw[0] = 0;
34        int minw = -1;
35        int minad = 0;
36        int minadwint = 0;
37        int minadw[] = new int[n];
38        String path[] = new String[n];
39
40        System.out.printf("%8s", "l(u0)");
41        for(int y=1;y<name.length;y++){
42            System.out.printf("%15s", name[y]);
43        }
44        System.out.printf("%15s\n","Add to S");
45        System.out.printf("%8s", "0");
46        for(int y=1;y<name.length;y++){
47            System.out.printf("%15s", "(INFI,-)");
48        }
49        System.out.printf("%15s\n","u0");
50
51        for (int i = 0; i < n - 1; i++) {
52            System.out.printf("%8s", "");
53            int w[] = new int[n];
54            int wnode[] = new int[n];
55
56            for (int j = 1; j < n; j++) {
57                if (visit[j] == 1 && j != 0) {
58                    System.out.printf("%15s", "");
59                } else if (visit[j] == 0) {
60                    for (int k = 0; k < stack.length; k++) {
61                        if(i>0){
62                            if (weight[j][stack[k]] != -1) {
63                                w[k] = weight[j][stack[k]]+stackw[stack[k]];
64                            } else if (weight[j][stack[k]] == -1) {
65                                w[k] = -1;
66                            }
67                        }
68                    }
69                }
70            }
71        }
72    }
73 }
```



```

69         if(i==0){
70             for (int k = 0; k < stack.length; k++) {
71                 w[k] = weight[j][stack[k]];
72             }
73         }
74         minw = -1;
75         boolean chk = false;
76         for (int l=0;l<w.length;l++) {
77             if (w[l] > -1 && visit[j] == 0) {
78                 chk = true;
79                 minw = w[l];
80             }
81         }
82
83

```

```

84         if (chk) {
85             for (int l : w) {
86                 if (visit[j] == 0) {
87                     if(l<minw && l!=-1){
88                         minw = l;
89                     }
90                 }
91             }
92         }
93
94         wnode[j] = minw;
95         minadw[j] = c.findindex(minw, w);
96         if (wnode[j] == -1) {
97             System.out.printf("%15s", "(INFI,-)");
98         } else {
99             System.out.printf("%15s", "("+wnode[j]+","+name[stack[minadw[j]]]+")");
100         }
101     }
102 }
103
104 for (int q = 0; q < wnode.length; q++) {
105     if (wnode[q] != -1 && visit[q] == 0) {
106         c.minn(wnode[q]);
107     }
108 }
109 int minn = c.getminn();
110 for (int q = 0; q < wnode.length; q++) {
111     if (wnode[q] < c.getminn() && wnode[q] != -1 && visit[q] == 0) {
112         c.minn(wnode[q]);
113     }
114 }
115
116 minad = c.findindex(c.getminn(), wnode);
117 minadwint = minadw[minad];
118 visit[minad] = 1;
119 stack[i + 1] = minad;
120 System.out.printf("%15s ", name[minad]);
121
122 stackw[minad] = stackw[stack[minadwint]]+weight[minad][stack[minadwint]];
123 path[i] = name[stack[minadwint]]+"->" + name[minad];
124 System.out.println("");
125 }
126
127 System.out.print("\nPath : ");
128 for(int t=0;t<path.length-1;t++){
129     System.out.print(path[t]+" , ");
130 }
131 System.out.println("\b\b\b");
132 System.out.println(
133     "-----");
134 }
135

```

Class Class

```
1 package Group8.MainClass;
2
3 public class Class {
4     private int minn;
5     private int minad;
6     private int min;
7
8     public void min(int min){
9
10        this.min = min;
11    }
12
13    public int getmin(){
14
15        return min;
16    }
17
18    public void minn(int minn){
19
20        this.minn = minn;
21    }
22
23    public int getminn(){
24
25        return minn;
26    }
27
28    public void minad(int minad){
29
30        this.minad = minad;
31    }
32
33    public int getminad(){
34
35        return minad;
36    }
37
38    public int findindex(int m,int[] n){
39        int index = 0;
40        int i = 0;
41        boolean chk = false;
42        for(i=0;i<n.length;i++){
43            if(n[i] == m) {
44                index = i;
45                chk = true;
46            }
47            if(chk) break;
48        }
49        return index;
50    }
51 }
52
```