

Kruskal's Algorithm

with disjoint set

소프트웨어학부

20170294 박해영

< 목 차 >

1. 목표	-----	3p
2. Problem & Input/Output	-----	3p
3. 구현 Language & 사용 Tool	-----	3p
4. 교재의 입력 데이터 테스트	-----	4p
5. 자작 입력 데이터 생성 & 알고리즘 과정 손계산	-----	5p
6. 자작 입력 데이터 테스트	-----	7p

1. 목표

비방향그래프에서 비순환적 그래프를 일컫는 신장트리(Spanning tree)가 무엇인지 정의를 이해하고, 신장트리를 이용하여 최소비용, 즉 최소 가중치를 가지는 최소비용 신장트리를 구하고자 한다. 최소비용을 구하기 위해서 Kruskal's Algorithm과 disjoint set abstract data type 자료구조를 이용하여 구현해보고자 한다.

2. Problem & Input / Output

* **Problem** : 최소비용 신장트리를 구하시오

* **Input** : n개의 정점과 m개의 이음선을 가진 연결된 가중치 포함 비방향 그래프.

이 그래프는 가중치와 함께 이음선이 포함된 집합 E로 표현한다.

* **Output** : 최소비용 신장트리에서 이음선의 집합 F

3. 구현 Language & 사용 Tool

* **구현 언어** : C++ language

* **사용 Tool** : Visual Studio 2015

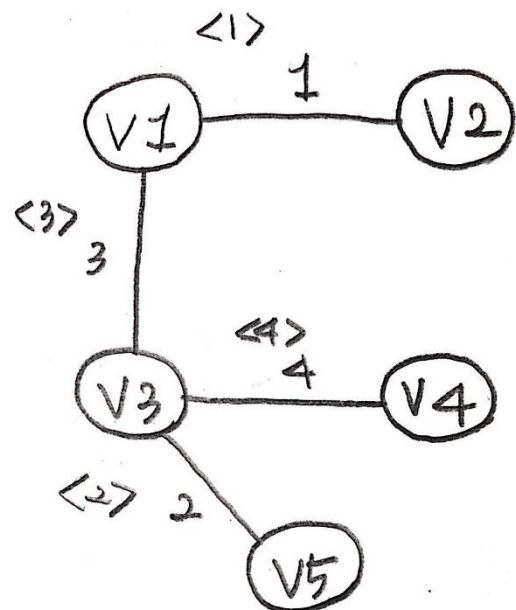
4. 교재의 입력 데이터 테스트

1) Ex4.7

```
Input Number of Vertices : 5
Input Number of Edges : 7
Input Format : Two Vertices and Weight
Input 1 : 1 2 1
Input 2 : 2 3 3
Input 3 : 1 3 3
Input 4 : 2 4 6
Input 5 : 3 4 4
Input 6 : 3 5 2
Input 7 : 4 5 5
```

< Minimum Spanning Tree >

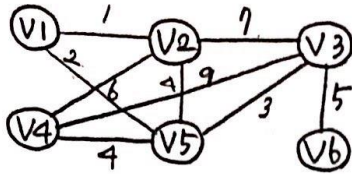
```
<1> ( v1, v2 ), weight = 1
<2> ( v3, v5 ), weight = 2
<3> ( v1, v3 ), weight = 3
<4> ( v3, v4 ), weight = 4
계속하려면 아무 키나 누르십시오 . . .
```



5. 자작 입력 데이터 생성 & 알고리즘의 과정 손계산

자작 데이터 생성.

정점 갯수 : 6 , 이음선 갯수 : 9.



$(V_1, V_2) - 1$, $(V_1, V_5) - 2$,
 $(V_2, V_3) - 7$, $(V_2, V_4) - 6$, $(V_2, V_5) - 4$
 $(V_3, V_4) - 9$, $(V_3, V_5) - 3$, $(V_3, V_6) - 5$
 $(V_4, V_5) - 4$,

Sort \Rightarrow quick Sort $O(m \lg m)$.

$(V_1, V_2) - 1$
 $(V_1, V_5) - 2$
 $(V_3, V_5) - 3$
 $(V_2, V_5) - 4$
 $(V_4, V_5) - 4$
 $(V_3, V_6) - 5$
 $(V_2, V_4) - 6$
 $(V_2, V_3) - 7$
 $(V_3, V_4) - 9$

\Rightarrow weight 기준으로
오름차순 정렬

Initial (n).

	V1	V2	V3	V4	V5	V6	
U	1	2	3	4	5	6	- parent
	0	0	0	0	0	0	- depth

반복 실행

① $(V_1, V_2) - 1$

$\rightarrow V_1$ 과 V_2 는 parent가 다르다 (다른집합)

\therefore merge $\Rightarrow \{V_1, V_2\}$

$U[1].depth = 1$, $U[2].parent = 1$

$Fset += (V_1, V_2) - 1$

	V1	V2	V3	V4	V5	V6	
U	1	1	3	4	5	6	- parent
	1	0	0	0	0	0	- depth

이와 같은 작업을 반복한다.

② $(V_1, V_5) - 2$

$\rightarrow V_1$ 과 V_5 는 parent가 다르다 (다른집합)

\therefore merge $\Rightarrow \{V_1, V_2, V_5\}$

$U[5].parent = 1$

$Fset += (V_1, V_5) - 2$

③ $(V_3, V_5) - 3$

$\rightarrow V_3$ 과 V_5 는 parent가 다르다 (다른집합)

\therefore merge $\Rightarrow \{V_1, V_2, V_3, V_5\}$

$U[3].parent = 1$

$Fset += (V_3, V_5) - 3$

	V1	V2	V3	V4	V5	V6	
U	1	1	1	4	1	6	- parent
	1	0	0	0	0	0	- depth

④ $(V_2, V_5) - 4$

$\rightarrow V_2$ 와 V_5 는 parent가 같다 (같은집합)

⑤ $(V_4, V_5) - 4$

$\rightarrow V_4$ 와 V_5 는 parent가 다르다 (다른집합)

\therefore merge $\Rightarrow \{V_1, V_2, V_3, V_4, V_5\}$

$U[4].parent = 1$

$Fset += (V_4, V_5) - 4$

⑥ $(V_3, V_6) - 5$

$\rightarrow V_3$ 과 V_6 는 parent가 다르다 (다른집합)

\therefore merge $\Rightarrow \{V_1, V_2, V_3, V_4, V_5, V_6\}$

$U[6].parent = 1$

	V1	V2	V3	V4	V5	V6	
U	1	1	1	1	1	1	- parent
	1	0	0	0	0	0	- depth

$$F_{set} += (v_3, v_6) - 5$$

$$F_{set} \text{ size} = 5 = 6 - 1$$

∴ minimum Spanning tree 완성

$$\begin{aligned} F_{set} = & (v_1, v_2) - 1 \\ & (v_1, v_5) - 2 \\ & (v_3, v_5) - 3 \\ & (v_4, v_5) - 4 \\ & (v_3, v_6) - 5 \end{aligned}$$

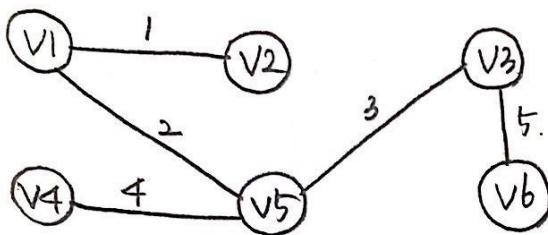
그러므로 depth가 0이 될 것이다

그래서 만약 minimum Spanning tree를

그리고 싶다면 집합 F를 이용하여

그려야 한다.

*MST 그리기



U 배열의 depth가

V1을 제외하고, 모두 0인 것은.

disjoint set의 merge 방법에 있다.

find에서 찾아주는 것은 어느 집합에 속하느냐,

이를 부모의 부모를 타고 올라가. parent 값과

인덱스 값이 같아질 때의 인덱스 값을 리턴한다

그렇다면 merge는 p와 q의 depth를

비교하여 합치게 된다.

따라서 이어제의 경우 모든 노드의

최상위 노드가 vertex 1이기 때문에

모든 vertex 1의 자식들로 불게 되고,

6. 자작 입력 데이터 테스트

```
Input Number of Vertices : 6
Input Number of Edges : 9
Input Format : Two Vertices and Weight
Input 1 : 1 2 1
Input 2 : 1 5 2
Input 3 : 2 3 7
Input 4 : 2 4 6
Input 5 : 2 5 4
Input 6 : 3 4 9
Input 7 : 3 5 3
Input 8 : 3 6 5
Input 9 : 4 5 4
```

< Minimum Spanning Tree >

<1> (v1, v2), weight = 1

<2> (v1, v5), weight = 2

<3> (v3, v5), weight = 3

<4> (v4, v5), weight = 4

<5> (v3, v6), weight = 5

계속하려면 아무 키나 누르십시오 . . .

