



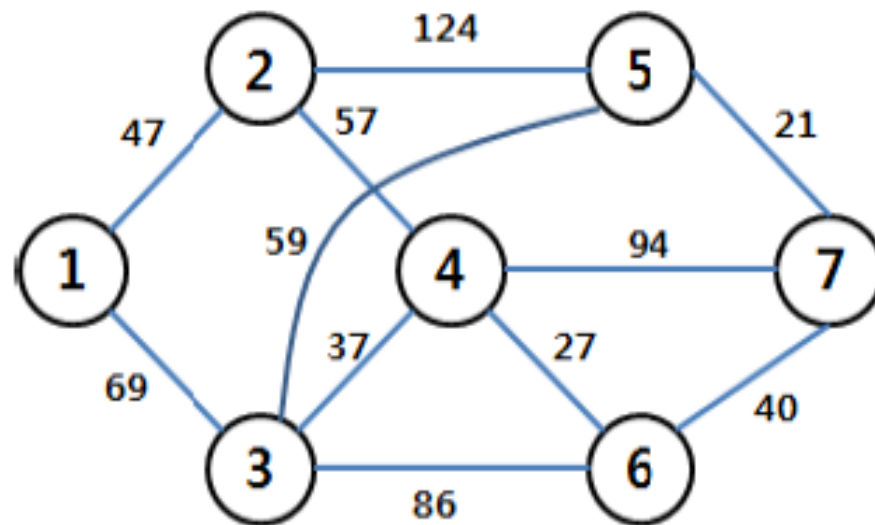
SW 역량 테스트 대비반 - 7회차

2020년 08월 26일

오늘 할 내용

구분	내용		
1주차 (07/08)		그래프 알고리즘	
2주차 (07/15)			
3주차 (07/22)			
4주차 (07/29)			
5주차 (08/05)			
6주차 (08/12)			
7주차 (08/19)	동적 계획법 강의 및 실습		
8주차 (08/26)	그래프 알고리즘 강의 및 실습		
(옵션) 9주차 ~	실전 감각 기르기 (실전 기출 기반 문제풀이 집중반)		

그래프의 표현 - 인접 행렬



7개의 정점과 11개의 간선을 가지는 가중치 그래프의 예



	1	2	3	4	5	6	7
1		47	69				
2	47			57	124		
3	69			37	59	86	
4		57	37			27	94
5		124	59				21
6			86	27			40
7				94	21	40	

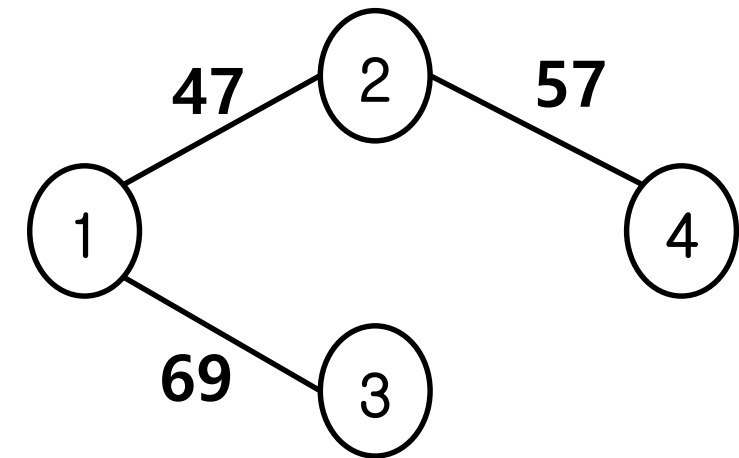
인접 행렬로
표현

그래프의 표현 - 인접 행렬 구현

입력 형식	
<p>첫 번째 줄에 정점의 수 n과 간선의 수 m이 공백으로 구분되어 입력된다.</p> <p>두 번째 줄부터 m개의 줄에 걸쳐서 간선으로 연결된 두 정점의 번호와 가중치가 공백으로 구분되어 입력된다.</p>	7 11
	1 2 47
	1 3 69
	2 4 57
	2 5 124
	3 4 37
	3 5 59
	3 6 86
	4 6 27
	4 7 94
	5 7 21
	6 7 40

입력

(가) 4 3
1 2 47
1 3 69
2 4 57



(나) $G[1][2] = 47$
 $G[2][1] = 47$
 $G[1][3] = 69$
 $G[3][1] = 69$
 $G[2][4] = 57$
 $G[4][2] = 57$
 로 저장

출력

(다) 1 2 47
1 3 69
2 1 47
2 4 57
3 1 69
4 2 57

가중치가 있는
 $G[a][b]$ 를 출력

그래프의 표현 - 인접 행렬 구현

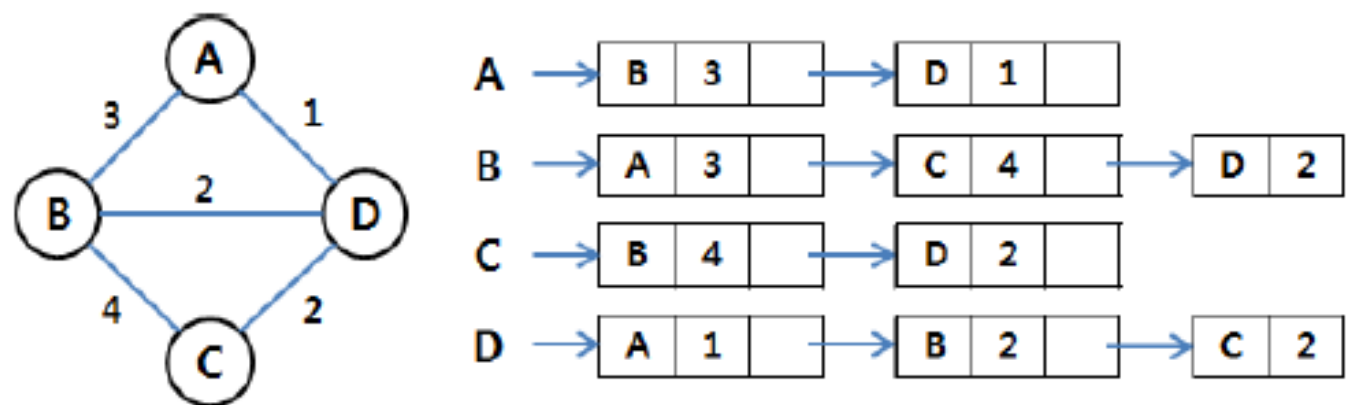
답안

```
#include <iostream>
using namespace std;

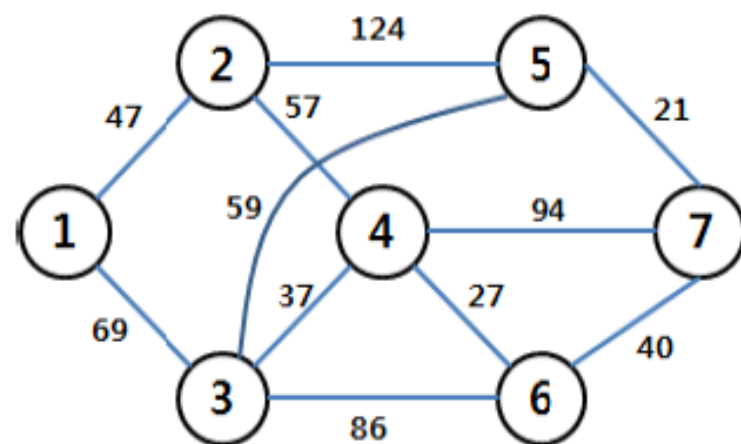
int n, m, G[101][101];
int main()
{
    cin >> n >> m;
    for (int i = 0; i < m; i++)
    {
        int u, v, w;
        cin >> u >> v >> w;
        G[u][v] = G[v][u] = w;
    }
    cout << endl;

    for (int i = 0; i <= n; i++)
    {
        for (int j = 0; j <= n; j++)
        {
            if (G[i][j] > 0)
            {
                cout << i << " " << j << " " << G[i][j] << endl;;
            }
        }
    }
}
```

그래프의 표현 - 인접 리스트



[그림-11] 가중치 그래프의 인접리스트 표현



1	2	47	3	69				
2	1	47	4	57	5	124		
3	1	69	4	37	5	59	6	86
4	2	57	3	37	6	27	7	94
5	2	124	3	59	7	21		
6	3	86	4	27	7	40		
7	4	94	5	21	6	40		

인접리스트에서는 정점과 가중치의 쌍으로 간선이 있는 것만 연결한다.
예를 들어 1행의 경우 1-2로의 간선의 가중치는 47이고 1-3으로의 간선의
가중치는 69라는 의미이다.

7개의 정점과 11개의 간선을 가지는 가중치 그래프의 예

구조체&포인터 이용 또는 std::vector() 이용

그래프의 표현 - 인접 리스트

답안

```
<iostream>
#include <vector>
using namespace std;

int main() {
    int n, m;
    vector<pair<int, int>> graph[101];

    cin >> n >> m; // 정점과 간선의 개수를 입력받음

    for (int i = 0; i < m; i++) {
        int u, v, w;

        cin >> u >> v >> w;
        graph[u].push_back(make_pair(v, w));
        graph[v].push_back(make_pair(u, w));
    }

    cout << endl;
    for (int i = 1; i <= n; i++) {
        for (int j = 0; j < graph[i].size(); j++) {
            if (graph[i][j].first <= 0) continue;
            cout << i << " ";
            cout << graph[i][j].first << " ";
            cout << graph[i][j].second << " ";
            cout << endl;
        }
    }
    return 0;
}
```

오일러 회로, 오일러 길, 해밀턴 순환로

모서리를 한 번만 지나면서 모두 지나는 경로가 존재하는지에 대한 한붓그리기 문제는 두 가지가 있습니다.

1 출발점과 도착점이 같은 **오일러 회로**(Euler circuit)

2 출발점, 도착점이 같다는 조건이 없는 **오일러 길**(Euler walk) 또는 **오일러 트레일**(Euler trail)

(출발점과 도착점이 같아도, 달라도 됨, 오일러 회로보다 넓은 의미)

쾨니히스버그 다리 문제는 1 오일러 회로 문제이고 <문제적 남자>에 나온 문제는 2 오일러 길 문제입니다.

꼭지점이 모두 짝수점이거나, 단 두개만 홀수점일 때 한붓그리기 문제가 가능하다 했습니다.

모두 짝수점일 때, 오일러 회로가 존재하며

두개만 홀수점일 때, 출발점과 도착점이 다른 오일러 길이 존재합니다.

그럼 각 도시에 있는 우체통의 편지를 수거하는 집배원은 어떤 경로를 이용하면 편리할까요?

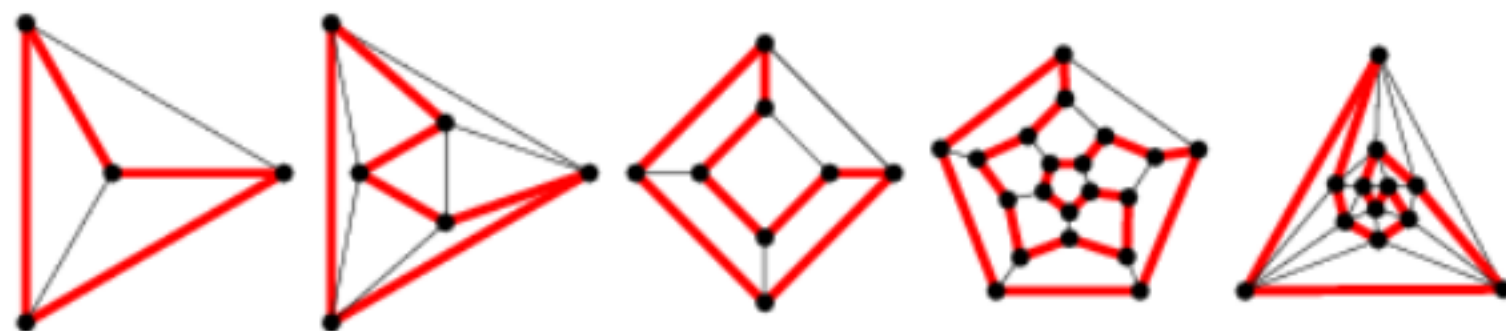
집배원은 길이 아닌 (우체통이 있는) 구역을 한 번만 지나면서 모두 돌아야 효율적입니다.

즉 꼭지점을 한 번만 그리고 모두 지나야 한다는 것입니다.

이것을 바로 **해밀턴 순환로**(Hamilton cycle)라고 합니다.

출발점과 도착점이 같으면서 꼭지점을 한 번, 그리고 모두 지나는 길, 경로를 해밀턴 순환로라고 합니다.

해밀턴 순환로의 예



오일러 회로

<https://www.acmicpc.net/problem/1199>

문제

문제

어느 점에서 출발하여 그래프 상에 있는 모든 간선을 지나되 한번 지난 간선은 다시 지나지 않고 출발점으로 돌아오는 회로를 오일러 회로라 한다. 단, 그래프는 양방향 그래프가 주어진다.

문제는 그래프가 주어졌을 때 오일러 회로 경로를 출력하는 것이다.

입력

첫 줄에는 정점의 수 N ($1 \leq N \leq 1,000$)이 주어진다. 그리고 다음 N 개의 줄에 대해 인접행렬의 정보가 주어진다. $i+1$ 번째 줄에는 i 번 정점에 대한 인접행렬이 주어진다. 두 정점 사이에 간선이 여러 개 있을 수 있다. 인접행렬의 값은 두 정점 사이의 간선 개수를 의미하며, 0보다 크거나 같고, 10보다 작거나 같은 정수이다.

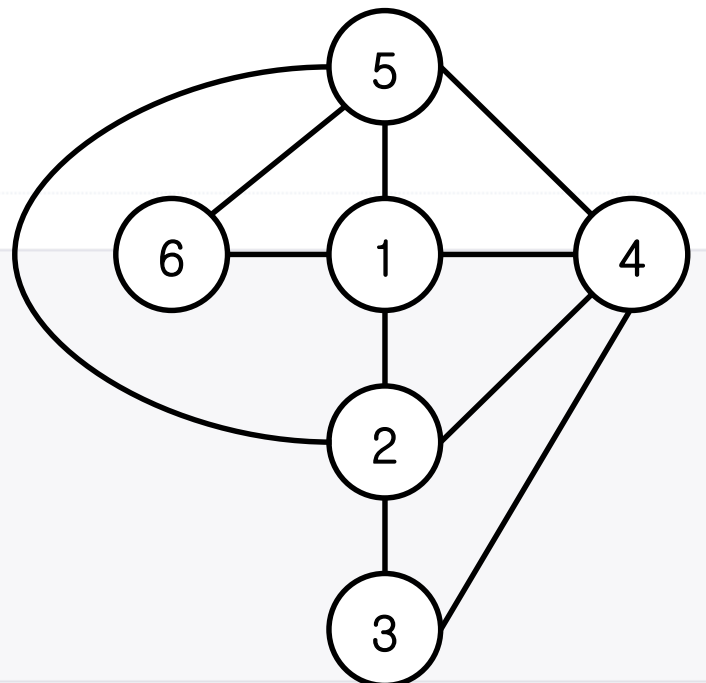
입력으로 주어지는 그래프에는 루프 (양 끝점이 같은 간선)는 없다. 또, 입력으로 주어지는 그래프는 모두 연결되어 있다.

출력

첫 줄에 방문하는 점 순서를 공백으로 구분하여 출력한다. 단, 시작점은 어느 위치에서든 상관없고 아무 경로만 하나 찍으면 된다. 불가능한 경우에는 -1을 출력한다.

예제 입력 1 복사

```
6
0 1 0 1 1 1
1 0 1 1 1 0
0 1 0 1 0 0
1 1 1 0 1 0
1 1 0 1 0 1
1 0 0 0 1 0
```

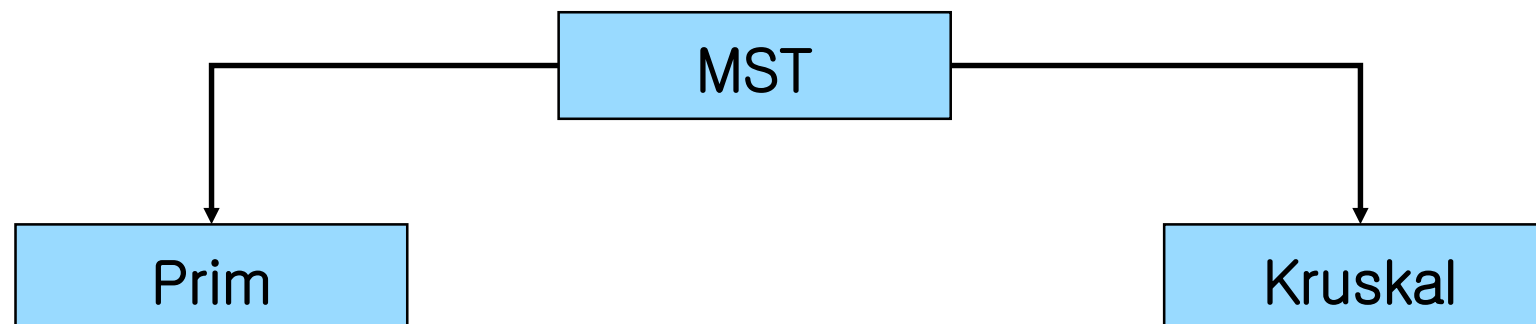
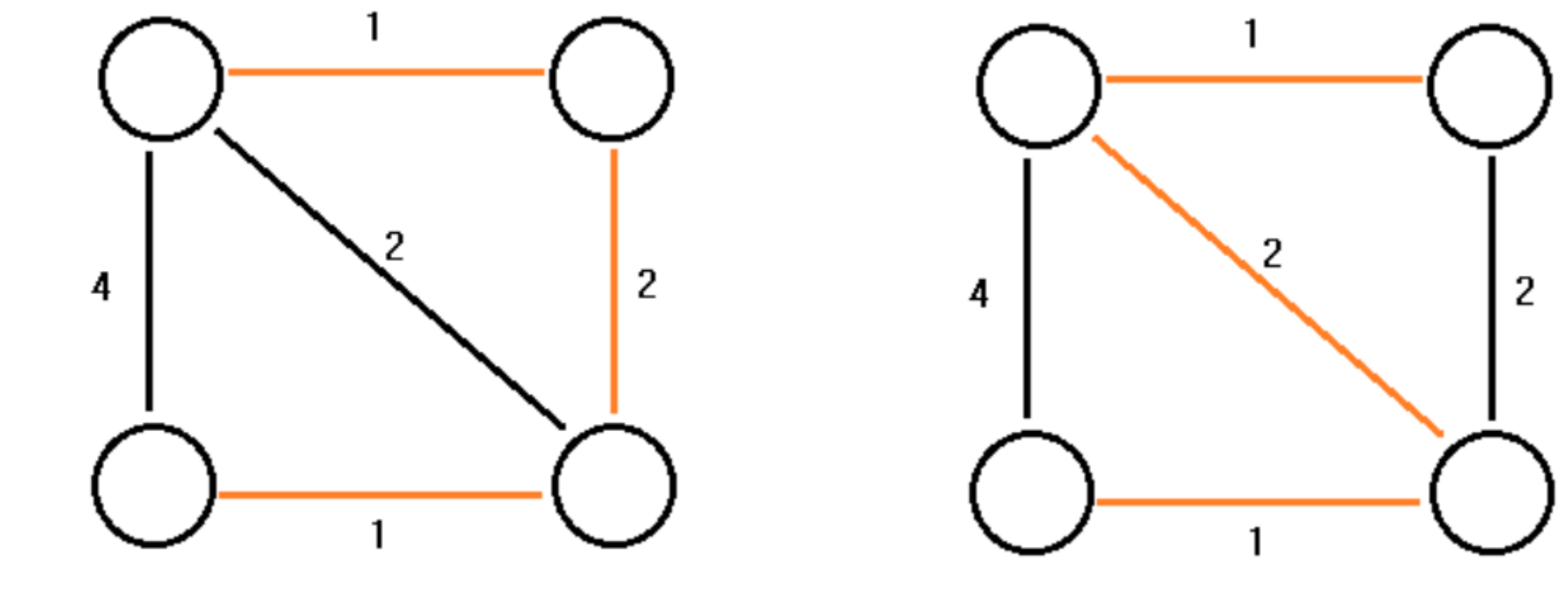


예제 출력 1 복사

```
1 2 3 4 1 5 2 4 5 6 1
```

최소 스패닝 트리 (MST)

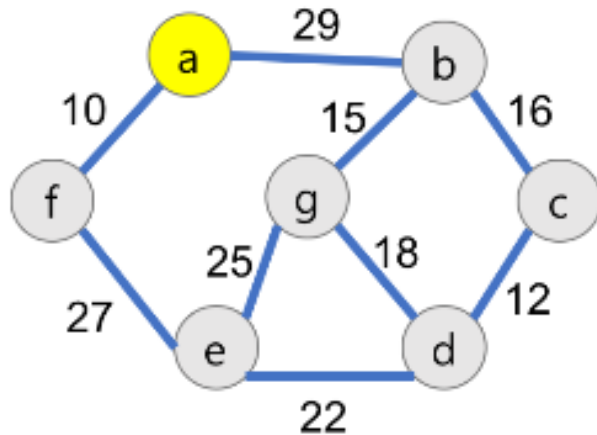
- Spanning Tree : 그래프 내의 모든 정점을 포함하는 트리
- Minimum Spanning Tree : Spanning Tree 중에서 사용된 간선들의 가중치 합이 최소인 트리



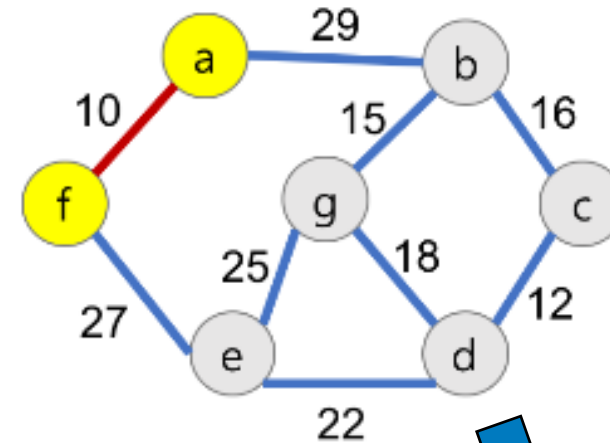
MST – Prim

Prim algorithm은 각 정점을 방문하며 그 정점에 연결된 가장 가중치가 작은 간선들을 선택하여 MST를 구합니다.

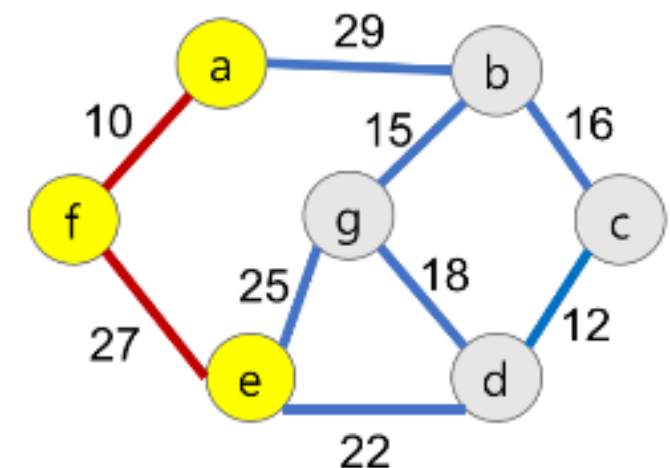
1) 시작 단계: 시작 정점만 포함



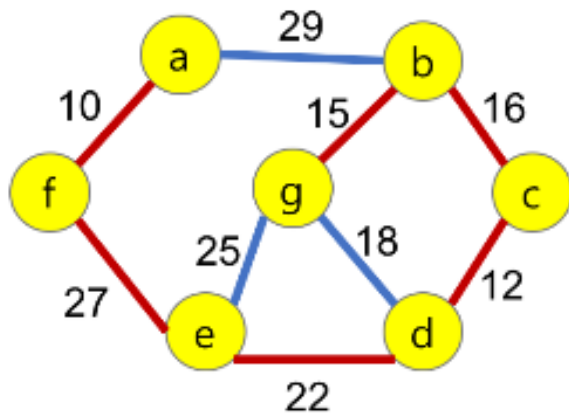
2) 인접 정점 중 최소 간선으로 연결된 정점 선택



3)



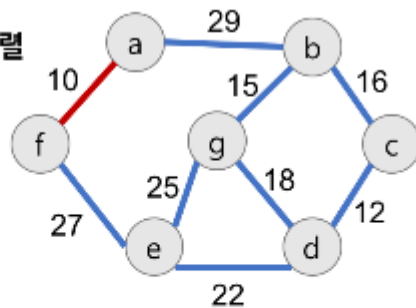
7) N-1개의 간선 생성. 종료



MST – Kruskal

Kruskal algorithm은 각 간선들을 탐색하여 가장 가중치가 작은 간선들을 택하면서 MST를 구합니다.

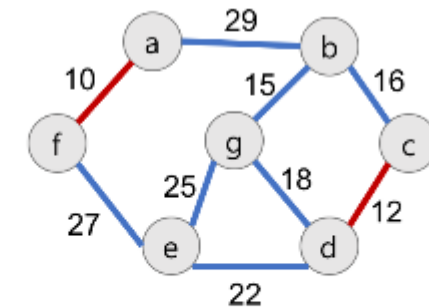
1) 간선들의 가중치 오름차순 정렬



간선의 가중치

af	cd	bg	bc	dg	de	eg	ef	ab
10	12	15	16	18	22	25	27	29

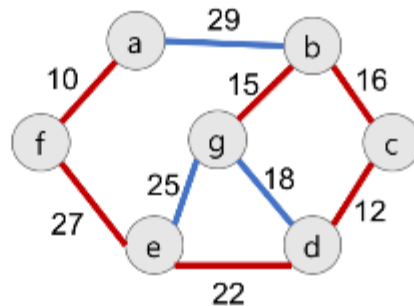
2)



간선의 가중치

af	cd	bg	bc	dg	de	eg	ef	ab
10	12	15	16	18	22	25	27	29

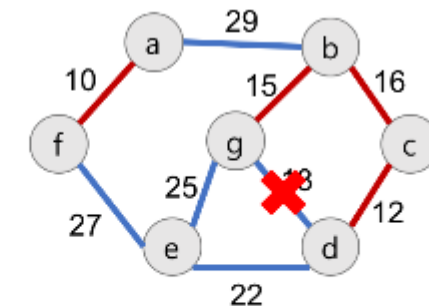
8) N-1개의 간선 생성. 종료



간선의 가중치

af	cd	bg	bc	dg	de	eg	ef	ab
10	12	15	16	18	22	25	27	29

5) 사이클 형성. dg는 제외



간선의 가중치

af	cd	bg	bc	dg	de	eg	ef	ab
10	12	15	16	18	22	25	27	29

문제풀이1

문제

<https://www.acmicpc.net/problem/1197>

문제

그래프가 주어졌을 때, 그 그래프의 최소 스패닝 트리를 구하는 프로그램을 작성하시오.

최소 스패닝 트리는, 주어진 그래프의 모든 정점들을 연결하는 부분 그래프 중에서 그 가중치의 합이 최소인 트리를 말한다.

입력

첫째 줄에 정점의 개수 V ($1 \leq V \leq 10,000$)와 간선의 개수 E ($1 \leq E \leq 100,000$)가 주어진다. 다음 E 개의 줄에는 각 간선에 대한 정보를 나타내는 세 정수 A, B, C 가 주어진다. 이는 A 번 정점과 B 번 정점이 가중치 C 인 간선으로 연결되어 있다는 의미이다. C 는 음수일 수도 있으며, 절댓값이 1,000,000을 넘지 않는다.

최소 스패닝 트리의 가중치가 -2147483648보다 크거나 같고, 2147483647보다 작거나 같은 데이터만 입력으로 주어진다.

출력

첫째 줄에 최소 스패닝 트리의 가중치를 출력한다.

예제 입력 1 복사

```
3 3
1 2 1
2 3 2
1 3 3
```

예제 출력 1 복사

```
3
```

문제풀이2

문제

문제

도현이는 컴퓨터와 컴퓨터를 모두 연결하는 네트워크를 구축하려 한다. 하지만 아쉽게도 허브가 있지 않아 컴퓨터와 컴퓨터를 직접 연결하여야 한다. 그런데 모두가 자료를 공유하기 위해서는 모든 컴퓨터가 연결이 되어 있어야 한다. (a와 b가 연결이 되어 있다는 말은 a에서 b로의 경로가 존재한다는 것을 의미한다. a에서 b를 연결하는 선이 있고, b와 c를 연결하는 선이 있으면 a와 c는 연결이 되어 있다.)

그런데 이왕이면 컴퓨터를 연결하는 비용을 최소로 하여야 컴퓨터를 연결하는 비용 외에 다른 곳에 돈을 더 쓸 수 있을 것이다. 이제 각 컴퓨터를 연결하는데 필요한 비용이 주어졌을 때 모든 컴퓨터를 연결하는데 필요한 최소비용을 출력하라. 모든 컴퓨터를 연결할 수 없는 경우는 없다.

입력

첫째 줄에 컴퓨터의 수 N ($1 \leq N \leq 1000$)가 주어진다.

둘째 줄에는 연결할 수 있는 선의 수 M ($1 \leq M \leq 100,000$)가 주어진다.

셋째 줄부터 $M+2$ 번째 줄까지 총 M 개의 줄에 각 컴퓨터를 연결하는데 드는 비용이 주어진다. 이 비용의 정보는 세 개의 정수로 주어지는데, 만약에 $a\ b\ c$ 가 주어져 있다고 하면 a컴퓨터와 b컴퓨터를 연결하는데 비용이 c ($1 \leq c \leq 10,000$) 만큼 든다는 것을 의미한다.

출력

모든 컴퓨터를 연결하는데 필요한 최소비용을 첫째 줄에 출력한다.

예제 입력 1 복사

```
6
9
1 2 5
1 3 4
2 3 2
2 4 7
3 4 6
3 5 11
4 5 3
4 6 8
5 6 8
```

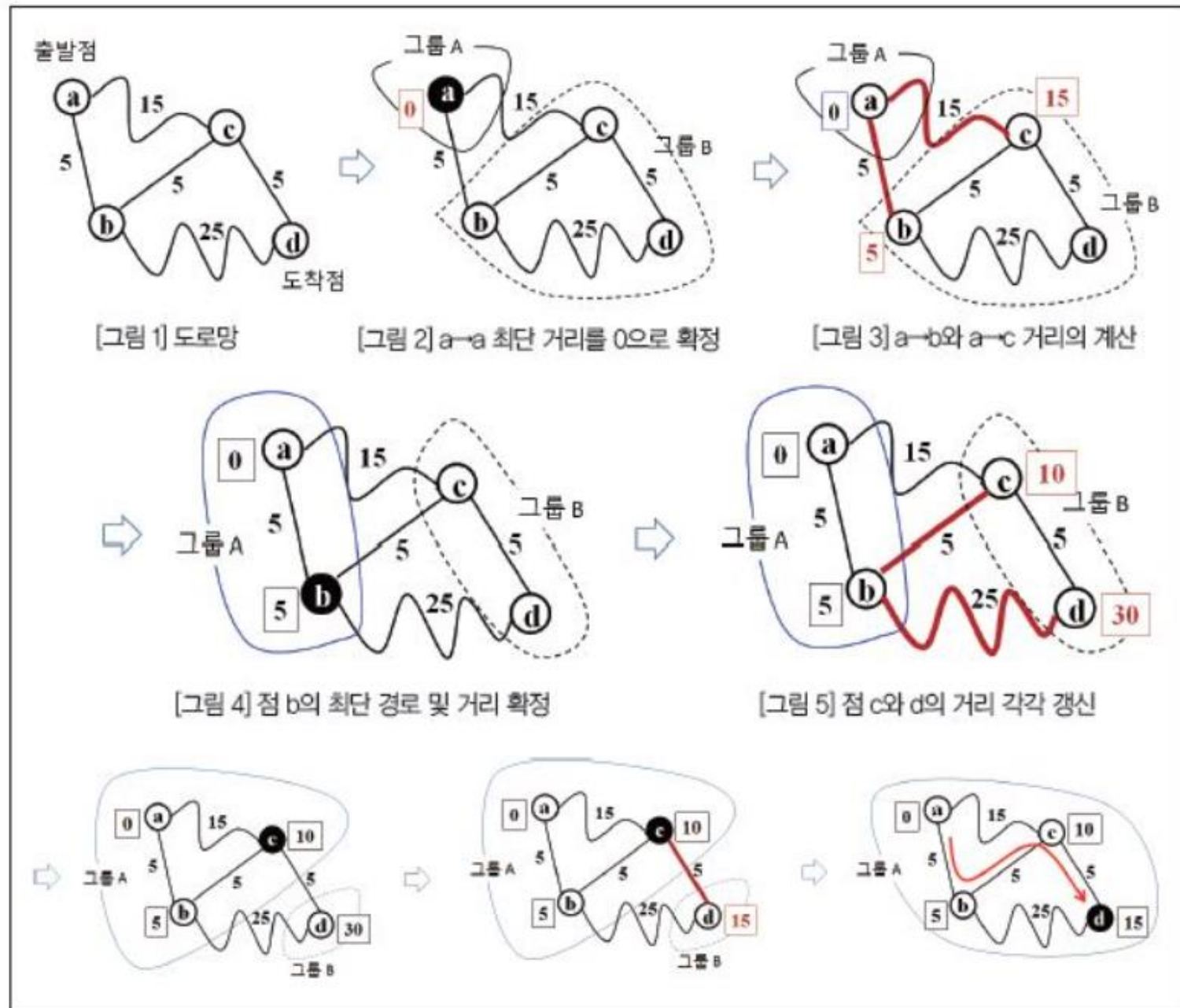
예제 출력 1 복사

```
23
```

다익스트라 알고리즘

다익스트라 알고리즘은 도로 교통망 같은 곳에서 나타날 수 있는 그래프에서 꼭짓점 간의 최단 경로를 찾는 알고리즘

하나의 시작 정점으로부터 모든 다른 정점까지의 최단 경로를 찾는 알고리즘



문제풀이3 (다익스트라)

문제

<https://www.acmicpc.net/problem/1753>

문제

방향그래프가 주어지면 주어진 시작점에서 다른 모든 정점으로의 최단 경로를 구하는 프로그램을 작성하시오. 단, 모든 간선의 가중치는 10 이하의 자연수이다.

입력

첫째 줄에 정점의 개수 V 와 간선의 개수 E 가 주어진다. ($1 \leq V \leq 20,000$, $1 \leq E \leq 300,000$) 모든 정점에는 1부터 V 까지 번호가 매겨져 있다고 가정한다. 둘째 줄에는 시작 정점의 번호 K ($1 \leq K \leq V$)가 주어진다. 셋째 줄부터 E 개의 줄에 걸쳐 각 간선을 나타내는 세 개의 정수 (u, v, w)가 순서대로 주어진다. 이는 u 에서 v 로 가는 가중치 w 인 간선이 존재한다는 뜻이다. u 와 v 는 서로 다르며 w 는 10 이하의 자연수이다. 서로 다른 두 정점 사이에 여러 개의 간선이 존재할 수도 있음에 유의한다.

출력

첫째 줄부터 V 개의 줄에 걸쳐, i 번째 줄에 i 번 정점에서의 최단 경로의 경로값을 출력한다. 시작점 자신은 0으로 출력하고, 경로가 존재하지 않는 경우에는 INF를 출력하면 된다.

예제 입력 1 복사

```
5 6
1
5 1 1
1 2 2
1 3 3
2 3 4
2 4 5
3 4 6
```

예제 출력 1 복사

```
0
2
3
7
INF
```


문제풀이4

문제

사이클 제거

문제 설명

n개의 노드로 구성된 그래프에서 하나의 노드만을 제거해서 사이클이 없도록 만들고 싶습니다.

노드의 개수 n, 노드간의 연결 edges가 매개변수로 주어질 때, 노드를 딱 **하나** 제거해서 그래프를 사이클이 없도록 만들 수 있다면 해당 노드의 번호를 return 하도록 solution 함수를 완성하세요.

단, 그런 노드가 없다면 0을 return하고, 여러 개라면 노드의 번호의 합을 return하세요.

제한사항

- 노드 번호는 1 부터 시작합니다.
- 주어진 그래프에 반드시 하나 이상의 사이클은 있습니다.
- 노드간의 연결에는 방향이 없습니다.
- 주어지는 노드의 수는 2개 이상 5,000개 이하입니다.
- 연결의 수는 1개 이상 $n(n + 1)/2$ 개 이하입니다.

입력

- 첫 줄에는 노드의 개수(n)와 간선의 수(edge)가 주어진다.
- 두 번째 줄부터 간선의 정보가 주어진다.

입력	출력
4 5	5
1 2	
1 3	
2 3	
2 4	
3 4	

입력	출력
8 10	0
1 2	
2 3	
3 4	
4 5	
5 6	
6 7	
7 8	
8 1	
2 7	
3 6	

기타

1. 온라인 시험인 경우에는 Visual Studio를 꼭 미리 설치해 놓자
(디버깅을 위해)
2. 데이터 타입에 따른 크기를 대략적으로라도 숙지하자
(<https://docs.microsoft.com/ko-kr/cpp/cpp/data-type-ranges?view=vs-2019>)
3. 파일 입출력을 잘 활용하자. (fscanf, fprintf 등)
(샘플을 일일이 입력하는 것은 시간 낭비임)
4. 최대한 스스로 풀어보자. (노트에 설계 -> 코딩)

참고 : <https://blog.encrypted.gg/724>