

〈알고리즘 실습〉 - 최소신장트리

※ 입출력에 대한 안내

- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정하라.
- 특별한 언급이 없으면, 각 줄의 맨 앞과 맨 뒤에는 공백을 출력하지 않는다.
- 출력 예시에서 □는 각 줄의 맨 앞과 맨 뒤에 출력되는 공백을 의미한다.
- 입출력 예시에서 ↦ 이 후는 각 입력과 출력에 대한 설명이다.

주의:

- 1) 프로그램 작성 시 사용 데이터구조의 **간편성**과 **효율성**은 모두 중요하다. 이 점에서 문제해결을 위해 사용한 데이터구조가 최선의 선택인지 여부는 채점 시 평가에 고려될 수 있다.
- 2) 예를 들어 그래프 알고리즘 구현 시, 그래프의 **인접 정보**(즉, 부착간선리스트 또는 인접행렬) 없이도 수행 가능한 문제라고 판단되면 **교재 13.4절의 간선리스트 구조**로 그래프를 **간편하게** 구현할 것을 우선적으로 고려하라. 그렇지 않고, **인접 정보**가 있어야 수행한다고 판단되면 **인접리스트 구조** 또는 **인접행렬 구조** 중에 해당 문제 해결에 **효율성** 면에서 유리하다고 판단되는 것을 선택하여 구현하라.

[문제 1] (**Prim-Jarnik 알고리즘**) 입력으로 주어지는 그래프를 Prim-Jarnik 알고리즘을 이용하여 최소신장트리(Minimum Spanning Tree, MST)를 생성하는 프로그램을 작성하고, 최소신장트리의 생성 과정과 총무게를 결과로 출력하시오.

입력 그래프의 성질:

- n ($1 \leq n \leq 100$) 개의 정점과 m ($1 \leq m \leq 1,000$) 개의 간선으로 구성된다.
- 정점은 $1 \sim n$ 사이의 정수로 번호가 매겨져 있고, 정점의 번호는 모두 다르다.
- 모든 간선은 **무방향간선**이고, 한 정점에서 임의의 다른 정점으로 가는 경로는 반드시 존재한다.
- 간선의 **무게는 중복이 없는 양의 정수**다.

주의:

- 알고리즘 수행의 출발정점은 번호가 가장 빠른 정점인 1부터 시작한다.
- Prim-Jarnik 알고리즘의 첫 출발정점은 그래프 내 아무 정점이라도 무방하지만, 이번 실습에서는 번호가 가장 빠른 정점인 1에서 출발해야 OJ 시스템의 정답과 일치시킬 수 있다.

입출력:

- 입력
 - 첫 줄에 정점의 개수 n , 간선의 개수 m 이 주어진다.

- 이후 m 개의 줄에 한 줄에 하나씩 간선의 정보(간선의 양끝 정점 번호와 무게)가 주어진다.
간선은 임의의 순서로 입력되고, 중복 입력되는 간선은 없다.
(무방향간선이므로 간선 (u, v) 와 (v, u) 는 동일한 간선으로 취급)
- 무게로는 양의 정수가 입력되고, 중복되는 무게는 없다.

○ 출력

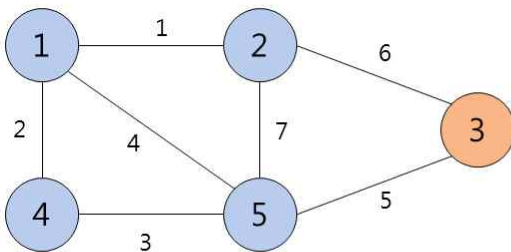
- 모든 정점의 번호를 출력한 후, 마지막 줄에 MST 간선 무게의 합 즉, 총무게를 출력한다.

입력 예시

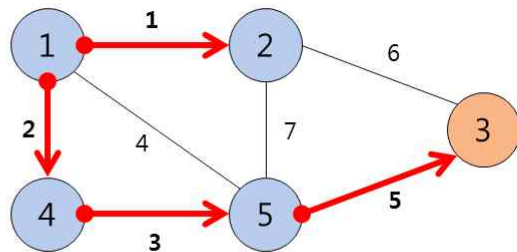
5 7	↪ $n = 5, m = 7$
1 2 1	↪ 정점, 정점, 무게
1 4 2	
1 5 4	
2 5 7	
4 5 3	
3 5 5	
2 3 6	

출력 예시

□ 1 2 4 5 3	↪ MST 생성시 추가되는 정점
11	↪ MST 총무게



<입력 그래프>



<Prim-Jarnik 결과>