

2. 최대 신장 트리

N개의 정점과 E개의 간선으로 이루어진 weighted, undirected graph G가 아래의 조건을 만족하도록 주어진다. 각 간선의 가중치는 1 이상 1,000 이하의 정수값을 갖는다.

- 1) G의 임의의 두 정점에 대해, 이를 잇는 경로(path)가 항상 존재한다. 즉, G는 connected graph다. 따라서 G에는 모든 정점을 잇는 신장 트리(spanning tree)가 존재한다.
- 2) G는 기본 그래프(normal graph)이다. 즉, 한 정점에서 동일한 정점으로 향하는 간선(self-loop)는 존재하지 않으며, 한 쌍의 정점을 두 개 이상의 간선이 연결하는 경우도 존재하지 않는다.

G의 신장 트리가 가질 수 있는 **최대** 가중치의 합을 구하는 알고리즘을 작성하라.

[제약사항]

정점의 개수 $5 \leq N \leq 20,000$, 간선의 개수 $5 \leq E \leq 40,000$

* 수행 시간이 다른 학생들에 비해 지나치게 길면 감점 요인이 됨.

[파일]

파일 이름은 "Solution2.java", 클래스 이름은 "Solution2"으로 작성한다.

```
> javac Solution2.java -encoding UTF8 && java Solution2
```

명령어를 입력했을 때 output2.txt 파일이 생성되어야 한다.

[입력]

입력 파일에는 10개의 테스트 케이스가 주어진다. 각 케이스는 두 줄로 이루어진다. 첫 줄에는 그래프 G의 정점 수 N과 간선 수 E가 주어진다. 다음 줄에는 E개 간선의 정보를 표현하는 정수들이 나열된다. 각 간선은 양 끝 정점의 번호들과 가중치, 총 세 개의 정수로 표현된다. 정점의 번호는 1번부터 N번까지로 할당된다. 예를 들어 "3 9 17 4 3 5 ..."은 3번 정점과 9번 정점을 잇는 가중치 17의 간선이 있고, 4번 정점과 3번 정점을 잇는 가중치 5의 간선이 있음을 의미한다. 즉, 총 3E 개의 정수가 공백을 사이에 두고 나열된다.

[출력]

각 테스트 케이스에 대해서, 케이스의 번호를 "#x" 의 형식으로 출력한 후(여기서 x는 테스트 케이스 번호), 공백을 하나 둔 다음 G의 신장 트리가 가질 수 있는 최대 가중치 합을 출력한다. 출력 결과물을 "output2.txt"로 저장한다.

[예제]

입력 (input2.txt)

5 10	← 1번 케이스
4 1 3 1 5 1 2 1 1 5 3 5 2 3 6 3 4 6 4 2 6 4 5 2 3 1 5 2 5 1	
7 12	← 2번 케이스
1 2 7 6 4 1 7 2 1 1 3 1 4 2 10 4 3 3 4 1 5 2 5 4 7 1 9 7 4 7 6 3 7 5 4 6	
...	

출력 (output2.txt)

#1 22
#2 42
...