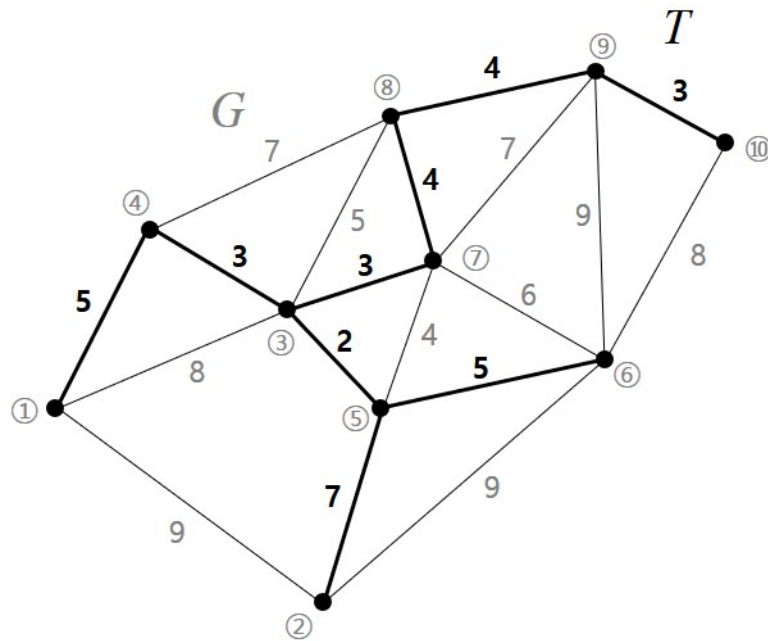


Minimum Spanning Tree

Connected, Weighted, Undirected 그래프 G 가 주어졌을 때, 이 그래프의 minimum spanning tree(MST)를 구하는 Prim 혹은 Kruskal의 알고리즘을 구현하시오. 두 알고리즘 중에서 어느 알고리즘으로 구현하여도 된다.

예를 들어, 10개의 노드와 가중치를 가지는 19개의 에지로 구성된 아래 그래프 G 의 MST T 는 다음과 같다.



위의 그래프에서 9개의 에지로 구성된 MST T 의 에지의 가중치의 총합은 37이다.

<주의>

C++을 사용하는 경우에는 STL 중에서 정렬 알고리즘만 사용할 수 있으며, 기타 `<vector>`, `<utility>`, `pair` 클래스 등 STL은 사용하지 못합니다. C에서는 `qsort()` 함수를 사용할 수 있으며, Java에서는 `Arrays.sort()` 메소드 혹은 다른 정렬 메소드를 사용할 수 있음. 그러나 기타의 라이브러리 함수 혹은 모듈을 사용하는 경우에는 0점 처리함.

입력

입력은 표준입력(standard input)을 사용한다. 입력은 t 개의 테스트 케이스로 주어진다. 입력 파일의 첫 번째 줄에 테스트 케이스의 개수를 나타내는 정수 t 가 주어진다. 두 번째 줄부터 테스트 케이스가

입력된다. 각 테스트 케이스에 해당되는 각 줄에는 그래프의 노드의 개수를 나타내는 정수 n ($3 \leq n \leq 100$) 이 주어진다. 그 다음 n 개의 줄에는 한 줄에 하나의 노드 k ($1 \leq k \leq n$) 에 인접한 노드와 번호와 그 노드에 연결된 에지의 가중치가 주어진다. 각 줄에는 먼저 노드 번호 k ($1 \leq k \leq n$) 가 주어지고, 그 다음에는 노드 k 에 인접한 노드들의 개수 m ($1 \leq m \leq n-1$) 이 주어진다. 그 다음에는 $2m$ 개의 정수 $v_1 w_1 v_2 w_2 \cdots v_m w_m$ 가 주어지는데, v_1, v_2, \dots, v_m 은 노드 k 에 인접한 노드들의 번호이고 w_1, w_2, \dots, w_m 은 해당 노드와 연결된 에지의 가중치를 나타낸다. 가중치의 최대는 100이며 최소는 1이다. 각 정수들 사이에는 한 개의 공백이 있으며, 잘못된 데이터가 입력되는 경우는 없다.

출력

출력은 표준출력(standard output)을 사용한다. 입력되는 테스트 케이스의 순서대로 다음 줄에 이어서 각 테스트 케이스의 결과를 출력한다. 각 테스트 케이스에 해당하는 출력의 첫 줄에 입력되는 그래프의 MST의 에지의 총합을 나타내는 정수를 출력한다.

입력과 출력의 예

입력
3 10 1 3 2 9 3 8 4 5 2 3 1 9 5 7 6 9 3 5 1 8 4 3 5 2 7 3 8 5 4 3 1 5 3 3 8 7 5 4 2 7 3 2 6 5 7 4 6 5 2 9 5 5 7 6 9 9 10 8 7 5 3 3 5 4 6 6 8 4 9 7 8 4 3 5 4 7 7 4 9 4 9 4 6 9 7 7 8 4 10 3 10 2 6 8 9 3 5 1 3 2 1 4 8 5 5 2 2 1 1 3 6 3 1 2 6 4 1 1 8 5 1 1 5 5 1 4 2 20 3 12 4 12 5 8 2 3 1 20 3 17 4 5 3 3 1 12 2 17 5 6 4 3 1 12 2 5 5 5 5 3 1 8 3 6 4 5

출력
36 20 24