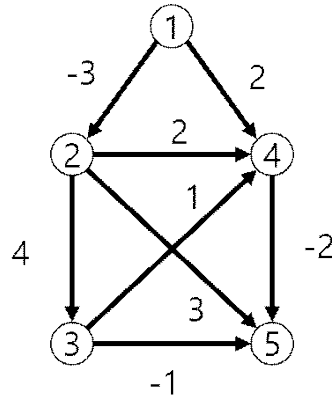


1. DAG에서의 최단 경로

Weighted directed acyclic graph, 즉 간선에 방향과 가중치가 있고 사이클이 없는 그래프가 주어진다. 간선의 가중치는 양 또는 음의 정수이다. 이 그래프의 정점을 각각 1번~N번이라 하자. 이때 하나의 정점으로부터 나머지 모든 정점으로의 최단거리를 구하고자 한다.



위의 그래프에서 1번 정점으로부터 나머지 정점으로의 최단거리는 각각 2: -3, 3: 1, 4: -1, 5: -3이다. 이와 같이 그래프가 주어질 때 1번 정점으로부터 나머지 모든 정점으로의 최단거리를 구하는 알고리즘을 작성하여라.

알고리즘은 최대한 효율적으로 작성하라. 10개의 케이스 모두를 수행한 알고리즘의 총 수행 시간이 3초를 넘지 않아야 한다. 채점을 위한 컴파일 시에 최적화 옵션은 쓰지 않는다.

[제약사항]

정점의 개수 $5 \leq N \leq 10,000$, 간선의 개수 $5 \leq E \leq 500,000$
총 수행 시간이 3초를 넘지 않아야 한다.

[입력]

입력 파일에는 10 개의 테스트 케이스가 주어진다. 각 케이스마다 첫 번째 줄에는 정점의 개수 N 과 간선의 개수 E 가 주어진다. 이후 E 개 줄에 각 간선의 시작점, 끝점, 가중치가 한 줄에 하나씩 주어진다. 정점 번호는 1번~ N 번으로 구성되어 있다. 입력파일의 이름은 "input1.txt"이다.

[출력]

각 테스트 케이스에 대해서 케이스 번호를 "#x"의 형식으로 출력한 후 (여기서 x 는 테스트 케이스 번호), 공백을 하나 둔 다음에 주어진 케이스에 대하여 2~ N 번 정점 각각에 이르는 최단 경로 거리 $N-1$ 개 수를 나열한다. 1번 정점에서 도달할 수 없는 정점에 대해서는 "X"를 출

력한다. 최단거리 계산 중 overflow는 일어나지 않도록 입력이 주어질 것이니 고려하지 않아도 된다. 출력 결과물을 “output1.txt”로 저장한다.

[예제]

입력 (input1.txt)

5 8	← 1번 케이스
1 2 -3	
1 4 2	
2 3 4	
2 4 2	
2 5 3	
3 4 1	
3 5 -1	
4 5 -2	
10 21	← 2번 케이스
6 10 -3	
9 5 5	
7 4 8	
5 10 -5	
7 6 4	
9 2 -2	
9 8 2	
1 3 -1	
5 2 10	
9 3 -7	
1 5 1	
6 2 5	
7 10 8	
1 10 -1	
3 4 -8	
1 6 -10	
3 6 10	
7 8 -4	
9 6 9	
8 5 -8	
1 7 4	
...	

출력 (output1.txt)

```
#1 -3 1 -1 -3  
#2 -5 -1 -9 -8 -10 4 0 X -13  
...
```