

3. All Pairs Shortest Paths

N개의 정점으로 이루어진 weighted directed graph G가 다음 조건을 만족하도록 주어진다.

- 1) 가중치 합이 음이 되는 사이클이 존재하지 않는다.
- 2) 강연결요소(strongly connected component)의 개수가 1개다. 즉, 임의의 한 점에서 임의의 다른 점으로 가는 경로가 항상 존재한다.

이 때, 모든 정점 쌍 사이의 최단 거리를 구하는 알고리즘을 구현하라. 각 간선의 가중치는 -1,000 이상 1,000 이하의 정수로 주어진다. 제출하는 프로그램에서는 모든 정점 쌍 (전체 $N*(N-1)$ 개) 사이의 최단 거리 총합을 출력하도록 한다. 알고리즘은 최대한 효율적으로 작성하라. 채점을 위한 컴파일 시에 최적화 옵션은 쓰지 않는다.

[제약사항]

정점의 개수 $1 \leq N \leq 200$, 간선의 개수 $1 \leq E \leq 10,000$

총 수행 시간이 3초를 넘지 않아야 한다.

[입력]

입력 파일에는 10 개의 테스트 케이스가 주어진다. 각 케이스는 두 줄로 이루어진다. 첫 줄에는 정점의 개수 N이 주어지고 공백을 두고 간선의 개수 E가 주어진다. 다음 줄에는 E개의 간선 정보가 공백을 두고 주어지는데, 각각의 간선 정보는 출발 정점, 도착 정점, 가중치로 이루어진 3개의 값이 공백을 두고 주어진다. 이 때, 정점의 번호는 1부터 N까지로 매긴다. 입력파일의 이름은 "input3.txt"이다.

[출력]

각 테스트 케이스에 대해서, 케이스의 번호를 "#x" 의 형식으로 출력한 후(여기서 x 는 테스트 케이스 번호), 공백을 하나 둔 다음 주어진 케이스에서 모든 정점 쌍 사이의 최단 거리의 총합을 출력한다. 총합 계산 중 overflow는 일어나지 않도록 입력이 주어질 것이니 고려하지 않아도 된다. 출력 결과물을 "output3.txt"로 저장한다.

[예제]

입력 (input3.txt)

2 2	← 1번 케이스
1 2 100 2 1 -50	
3 3	← 2번 케이스
1 2 100 2 3 -50 3 1 30	
...	

출력 (output3.txt)

#1 50
#2 240
...