

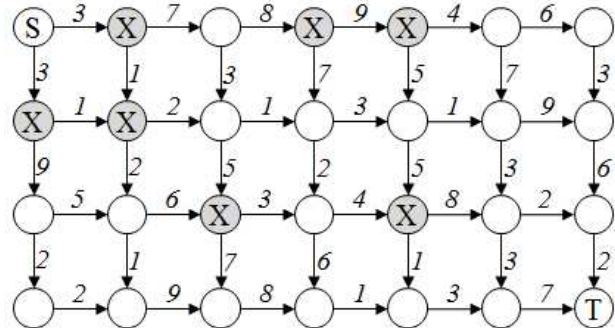
## 문제 : 격자에서의 최단경로

## 문제 설명 :

가로로 난 도로가  $M$ 개, 세로로 난 도로가  $N$ 개인 격자모양의 도로망이 있다. 이 도로에서는 우측 또는 아래로만 이동이 가능하며 좌측상단 모서리( $S$ )에서 출발하여 우측하단 모서리( $T$ )까지 가고자 한다. 격자의 교차점 몇 군데는  $\otimes$ 로 표시되어 있다. 그리고 각 구간에는 이동 비용을 나타내는 정수 값  $w$ 가 주어져 있다. 이동 경로에 따라  $\otimes$ 로 표시된 지점을 한 군데도 지나가지 않을 수 있고, 또는  $\otimes$ 로 표시된 곳을  $k(k \geq 1)$ 번 지나갈 수 있다. 우리는 가능한  $k$ 값 각각에 대해,  $S$ 에서  $T$ 로 가는 경로 중  $\otimes$ 로 표시된 곳을 정확히  $k$ 번 지나가는 경로가 총 몇 가지인지, 그리고 그런 가능한 경로 중 이동비용이 가장 적게 드는 경로의 비용은 얼마인지, 최소 비용의 경로는 어떻게 되는지를 찾고자 한다. 물론 정확히  $k$ 번 지나가는 경로가 존재하지 않는 경우는 고려 대상에서 제외한다.

도로의 각 교차점의 위치는 행 번호와 열 번호로 표시되는 좌표로 표시한다. 출발점  $S$ 의 좌표는  $(0,0)$ 으로, 도착점  $T$ 의 좌표는  $(M-1, N-1)$ 으로 나타낸다.

예를 들어 그림에서 보인 것처럼, 가로로 난 도로가 4개, 세로로 난 도로가 7개인 즉,  $M=4$ ,  $N=7$ 인 격자 모양의 도로를 생각하자.  $S$ 에서  $T$ 로 가는 경로 중  $\otimes$ 로 표시된 곳을 한 번도 지나가지 않는 경로는 존재하지 않는다. 하지만,  $\otimes$ 로 표시된 곳을 정확히 한 번 지나가는 경로는 6개가 있다 (각자 해아려 보자.) 또한  $\otimes$ 로 표시된 곳을 정확히 두 번, 세 번, 네 번 지나가는 경로는 각각 24, 39, 15개가 있다.  $\otimes$ 로 표시된 곳을 정확히 한 번 지나가는 6개의 경로 중 비용이 가장 적게 드는 경로는



$(0,0) \rightarrow (0,1) \rightarrow (0,2) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (1,4) \rightarrow (1,5) \rightarrow (2,5) \rightarrow (2,6) \rightarrow (3,6)$  이고, 그 때의 비용은 25이다. 유사하게,  $\otimes$ 로 표시된 곳을 정확히 네 번 지나가는 15개의 경로 중 비용이 가장 적게 드는 경로는

$(0,0) \rightarrow (1,0) \rightarrow (1,1) \rightarrow (1,2) \rightarrow (2,2) \rightarrow (2,3) \rightarrow (2,4) \rightarrow (3,4) \rightarrow (3,5) \rightarrow (3,6)$ 이고, 그 때의 비용은 29이다. 참고로  $\otimes$ 로 표시된 곳을 정확히 네 번 지나가는 최소비용의 경로는 위에서 보인 경로와  $(0,0) \rightarrow (0,1) \rightarrow (1,1) \rightarrow (1,2) \rightarrow \dots$  으로 따라가는 두 가지이다. 최소 비용의 경로가 두 개 이상일 경우 우측 보다는 아래쪽으로 이동하는 경로를 따라 가는 것을 택하기로 한다.

## 입력 :

입력파일의 이름은 path.inp이다. 입력 파일의 첫 번째 줄에는 총 테스트할 경우의 수를

나타내는 정수  $C$ 가 주어진다.

각 테스트 케이스 첫째 줄엔 세 정수  $M, N, K$ 가 주어진다.  $M, N(1 \leq M, N \leq 100)$ 은 문제에서 설명한 격자모양 도로의 개수를 나타낸다.  $K(1 \leq K \leq 200)$ 는 ㊸로 표시된 교차점의 개수를 나타낸다. 이어서,  $K$ 줄에 걸쳐 ㊸로 표시된 교차점의 좌표값이 주어진다. 이어서, 각 도로 구간에 할당된 이동 비용이 주어진다. 처음  $M$ 줄에는 가로 도로의 각 구간에 할당된 이동 비용이 주어지며, 각 줄엔  $N-1$ 개 값이 공백으로 구분되어 주어진다. 이어서  $M-1$ 줄 걸쳐 세로 도로의 각 구간에 할당된 이동 비용이 주어지며, 각 줄엔  $N$ 개의 값이 공백으로 구분되어 주어진다. 이동 비용은 모두 100이하의 양수이다.

각 테스트 케이스는 빈 줄로 구분된다.

**출력 :**

출력파일의 이름은 path.out이다. 테스트 케이스 각각에 대해 다음을 출력해야 한다. 우선 테스트 케이스 번호를 예에서 보인 것처럼 보인다. 그리고,  $k(0 \leq k \leq K)$  각각에 대해, ㊸로 표시된 교차점을 정확히  $k$ 번 지나가는 경로가 존재할 경우, ① 그 때의  $k$ 값 ② 그런 경로가 몇 개인지, ③ 그 중 가장 비용이 최소인 경로의 비용이 얼마인지, ④ 그 최소 비용의 경로 정보를 예에서 보인 것처럼 출력한다.

정보 ①,②,③은 예에서 보인 것처럼 같은 줄에 출력하며 각 값은 공백으로 구분하고, 정보 ④는 다음 줄에 따로 **한 줄**에 출력한다.

정확히  $k$ 번 지나가는 경로가 없다면 그런  $k$ 에 대해서는 정보를 보이지 않아도 된다.

**경로의 개수를 보일 때 주의사항:** 경로의 개수가 너무 많아 integer overflow가 생기는 것을 방지하기 위해, 경로의 수를 보일 때는 하위 5자리까지만 보인다. 예를 들어, 경로의 수가 123,456,789인 경우 56789로 출력한다. 만약, 마지막 5자리의 값이 0인 경우, 실제로는 경로가 존재하더라도 경로가 존재하지 않는 것처럼 처리한다.

**만약 최소비용의 경로가 하나 이상 존재할 경우:** 아래 방향으로 먼저 진행하는 경로를 선택하여 보인다. 예에서 보인 두 번째 test case 및 첫 번째 case에서  $k=4$ 인 경우를 참조하라.

**예제 :**

(참고로 첫 번째 테스트 케이스는 문제 설명 그림에서 보인 것에 대응된다.)

입력 예
<pre> 3  4 7 7 1 0 0 3 0 4 0 1 1 1 2 2 2 4 3 7 8 9 4 6 1 2 1 3 1 9 5 6 3 4 8 2 2 9 8 1 3 7 3 1 3 7 5 7 3 9 2 5 2 5 3 6 2 1 7 6 1 3 2  3 4 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  4 7 5 0 1 1 0 2 2 3 5 2 6 6 9 8 9 4 6 1 7 8 3 7 9 5 6 3 4 8 2 2 9 8 1 3 7 6 1 3 7 5 7 3 9 2 5 2 5 8 6 2 9 6 6 1 3 2 </pre>
입력 예에 대한 출력
<pre> Test Case No:1 k:1 count:6 cost:25 (0,0)-&gt;(0,1)-&gt;(0,2)-&gt;(1,2)-&gt;(1,3)-&gt;(1,4)-&gt;(1,5)-&gt;(2,5)-&gt;(2,6)-&gt;(3,6) k:2 count:24 cost:18 (0,0)-&gt;(1,0)-&gt;(1,1)-&gt;(1,2)-&gt;(1,3)-&gt;(1,4)-&gt;(1,5)-&gt;(2,5)-&gt;(2,6)-&gt;(3,6) k:3 count:39 cost:24 (0,0)-&gt;(1,0)-&gt;(1,1)-&gt;(1,2)-&gt;(1,3)-&gt;(2,3)-&gt;(2,4)-&gt;(3,4)-&gt;(3,5)-&gt;(3,6) k:4 count:15 cost:29 (0,0)-&gt;(1,0)-&gt;(1,1)-&gt;(1,2)-&gt;(2,2)-&gt;(2,3)-&gt;(2,4)-&gt;(3,4)-&gt;(3,5)-&gt;(3,6)  Test Case No:2 k:0 count:10 cost:5 (0,0)-&gt;(1,0)-&gt;(2,0)-&gt;(2,1)-&gt;(2,2)-&gt;(2,3)  Test Case No:3 k:2 count:54 cost:39 (0,0)-&gt;(1,0)-&gt;(1,1)-&gt;(1,2)-&gt;(1,3)-&gt;(2,3)-&gt;(2,4)-&gt;(3,4)-&gt;(3,5)-&gt;(3,6) k:3 count:30 cost:33 (0,0)-&gt;(1,0)-&gt;(1,1)-&gt;(2,1)-&gt;(2,2)-&gt;(2,3)-&gt;(2,4)-&gt;(3,4)-&gt;(3,5)-&gt;(3,6) </pre>

제한조건: 프로그램은 path.{c,cpp,java}로 한다.