안녕하세요

알다마 1회차 문제를 올릴께요

회사 알고리즘 시험에서 출제빈도가 높은것은

최단거리 경로를 구하는 문제라 생각됩니다.

하지만 첫번째 회차이기 때문에 그거보다는 조금 간단한

가능한 경로의 개수를 푸는 문제를 골라보았어요.

(아직 실제로 풀어보지 않아서 아닐수도 있습니다 ㅋㅋㅋ)

해당 문제는

<https://algospot.com/> -> 문제풀기 -> 3번째 페이지 -> 어떤 과학의 인공지능 청소기

에서 답안을 제출하고 확인해 볼 수 있습니다.

|  |
| --- |
| 문제  어떤 최첨단 과학 도시에 살고 있는 일방통행 씨는 직사각형 모양의 방에서 살고 있었다. 최첨단 과학 도시임에도 불구하고 손수 청소를 하던 그는 큰 결심을 했다. 최첨단 과학 도시의 이름에 걸맞게 인공지능 청소기를 구입하기로 한 것이다.  인공지능 청소기는 과학 도시에서 만들어진 물건답게 매우 훌륭한 인공지능을 갖고 있다. 청소기는 지나간 곳을 절대로 다시 지나가지 않으면서도 모든 바닥을 청소해내는 능력을 가지고 있다. 그러한 전략적인 움직임을 보여주기 위해서는 높은 수준의 계산 능력이 필요하므로 몇 가지 움직임에 제약이 있다. 청소기는 직사각형 방 안에서만 훌륭하게 동작한다. 청소기가 직사각형 방 안에서 청소를 하게 되면 항상 이동은 벽에 수직이거나 평행한 방향으로만 하게 된다. 또한 한 번 이동하게 되면 정확히 1미터는 직선으로 이동하며 이동시간은 1분 정도 걸린다.  일방통행 씨의 방은 다행히 이러한 청소기가 잘 작동하는 직사각형 방이었다. 그의 방은 가로 M 미터, 세로 N 미터이다.  며칠 뒤 일방통행 씨는 인공지능 청소기를 받아들고는 곧장 방 구석에 놓고 작동시켰다. 그리고 다른 방의 소파에 누워서 TV를 틀어보았다. 한참을 재밌게 보던 그는 가끔씩 방에 돌아다니는 청소기가 제대로 돌고 있는지 쳐다보곤 했다. 마침내 청소 로봇은 청소가 끝났다는 신호음을 시끄럽게 울려댔고, 일방통행 씨는 청소기가 반대쪽 방 구석에 있는 것을 확인하고 전원을 뽑았다.  이 때 일방통행 씨는 궁금한 것이 생겼다. ‘대체 이 청소기는 어떻게 움직인거지?’ 일방통행 씨는 자신의 기억을 더듬어보면서 청소기가 언제 어느 위치에 있었는지를 가능한 한 복원해보았다. 그런데 이것만으로는 청소기의 경로를 알기가 쉽지 않았다.  당신이 해야할 것은 일방통행이 얻은 청소기 위치 정보를 이용해 청소기가 이동하는 경로의 경우의 수를 구하는 것이다.  입력  입력의 첫번째 줄에는 테스트 케이스의 수 T(≤100)가 주어진다. 각 테스트 케이스는 여러 줄로 구성되어있다.  각 테스트 케이스의 첫번째 줄에는 방의 크기 N, M (1 ≤ N ≤ 6, 1 ≤ M ≤ 6)이 주어져 있다.  그 다음 N개의 줄에는 M개의 수가 주어진다. i번째 줄의 j번째 수는 방의 (i,j) 위치에 청소기가 위치했을 때의 분 단위의 시간을 나타낸다. 만약 -1이라면 그 위치에 청소기가 있었던 정보를 얻지 못했다는 의미이다. 시작 시간은 0분이고, 끝 시간은 N x M - 1이다. 첫 번째 줄의 첫 번째 수는 항상 0이다. 마지막 줄의 마지막 수는 항상 N x M - 1이다.  출력  테스트 케이스마다 한 줄씩 가능한 경로의 수를 1,000,000,007로 나눈 나머지를 출력한다. 가능한 경로가 없다면 0을 출력한다.  예제 입력  3  3 2  0 -1  -1 2  -1 5  3 3  0 -1 -1  -1 2 -1  -1 -1 8  4 3  0 1 2  -1 -1 3  -1 -1 -1  -1 -1 11  예제 출력  1  0  2 |

보너스 문제로

프로그램 실행 시간을 단축시킬 수 있는 기법을 시험해 볼 수 있는 문제를 추가합니다.

다이나믹프로그래밍?메모제이션?동적계획법? 이름은 정확히 정의하기 힘들지만 실행 시간을

단축시킬 수 있는 간단한 기법이에요.

다음 기본 코드는 피보나치 수열을 구하는 문제입니다.(지금은 집에서 작성중이라 문법이 틀릴수도 있어요)

이 코드를 돌려보면 실행시간이 너무 길어서 결과가 안나올 수도있고 엄청 기다려서 나올수도 있는데요 이 코드에 적절한 기법을 추가해서 실행하자마자 결과가 나오게 바꿔보면 됩니다.

|  |
| --- |
| public class test{  public static void main(String [] args) {  System.out.println(fibonacci(100));  }  static long fibonacci(long n) {  if(n == 0) {  return 0;  }  else if(n ==1) {  return 1;  }  else {  return fibonacci(n-1) + fibonacci(n-2);  }  }  } |