question

# 이동하기 성공 풀이

문제집

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **시간 제한** | **메모리 제한** | **제출** | **정답** | **맞은 사람** | **정답 비율** |
| 1 초 | 256 MB | 7065 | 3952 | 2748 | 57.901% |

## 문제

준규는 N×M 크기의 미로에 갇혀있다. 미로는 1×1크기의 방으로 나누어져 있고, 각 방에는 사탕이 놓여져 있다. 미로의 가장 왼쪽 윗 방은 (1, 1)이고, 가장 오른쪽 아랫 방은 (N, M)이다.

준규는 현재 (1, 1)에 있고, (N, M)으로 이동하려고 한다. 준규가 (r, c)에 있으면, (r+1, c), (r, c+1), (r+1, c+1)로 이동할 수 있고, 각 방을 방문할 때마다 방에 놓여져있는 사탕을 모두 가져갈 수 있다. 또, 미로 밖으로 나갈 수는 없다.

준규가 (N, M)으로 이동할 때, 가져올 수 있는 사탕 개수의 최대값을 구하시오.

## 입력

첫째 줄에 미로의 크기 N, M이 주어진다. (1 ≤ N, M ≤ 1,000)

둘째 줄부터 N개 줄에는 총 M개의 숫자가 주어지며, r번째 줄의 c번째 수는 (r, c)에 놓여져 있는 사탕의 개수이다. 사탕의 개수는 0보다 크거나 같고, 100보다 작거나 같다.

## 출력

첫째 줄에 준규가 (N, M)으로 이동할 때, 가져올 수 있는 사탕 개수를 출력한다.

## 예제 입력 복사

3 4  
1 2 3 4  
0 0 0 5  
9 8 7 6

## 예제 출력 복사

31

## 예제 입력 2 복사

3 3  
1 0 0  
0 1 0  
0 0 1

## 예제 출력 2 복사

3

## 예제 입력 3 복사

4 3  
1 2 3  
6 5 4  
7 8 9  
12 11 10

## 예제 출력 3 복사

47

answer -1 good 런타임이 좀 ㅜㅠ

package alogprithm;

import java.util.ArrayList;

import java.util.Scanner;

public class T\_11048 {

private static ArrayList<ArrayList<Integer>> candyl = new ArrayList<ArrayList<Integer>>();

private static ArrayList<ArrayList<Integer>> candyg = new ArrayList<ArrayList<Integer>>();

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int n = scan.nextInt();

int m = scan.nextInt();

ArrayList<Integer> candys = new ArrayList<>();

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

candys.add(scan.nextInt());

}

candyl.add(candys);

candyg.add(new ArrayList<Integer>(m));

candys = new ArrayList<Integer>();

}

candyg.get(0).add(candyl.get(0).get(0));

System.out.println(cal(n, m));

}

public static int cal(int n, int m) {

if (n <= 0 || m <= 0) {

return 0;

}

if (candyg.get(n-1).size() >= m) {

return candyg.get(n - 1).get(m - 1);

}

candyg.get(n-1).add(Math.max(Math.max(cal(n - 1, m ), cal(n, m - 1)), cal(n - 1, m - 1)) + candyl.get(n-1).get(m-1));

return candyg.get(n-1).get(m-1);

}

}