11048번: 이동하기

이동하기 풀이

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	256 MB	6796	3809	2638	58.195%

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.

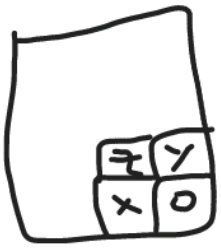
문제유형 → Dynamic programming

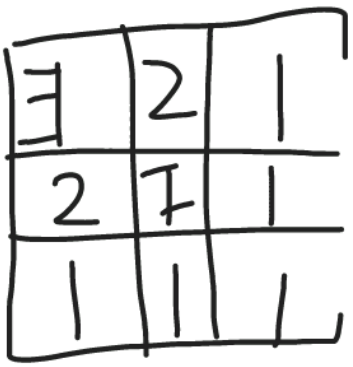
f(m,n) = m,n 방에서 얻을수 있는 점수

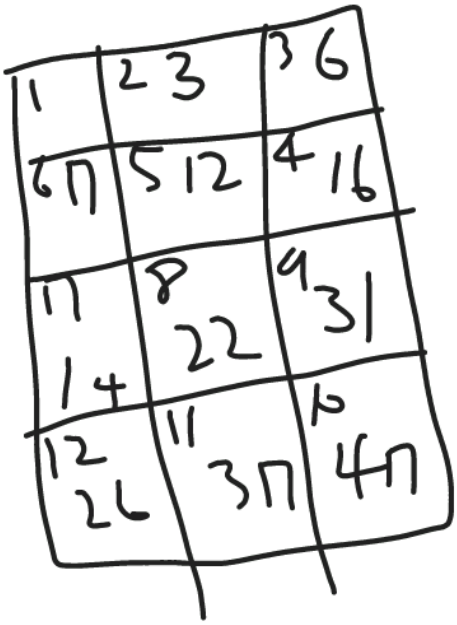
$$f(m,n) = \begin{cases} map[m,n] & \text{if } m=\text{행}, n=\text{열} \\ \max(f(m+1,n+1), f(m+1,n), f(m,n+1)) + map(m,n) \end{cases}$$

⇓

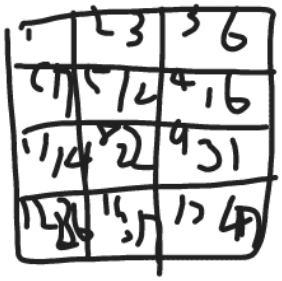
top-down







bottomUp



! Safe

( )

문제 유형 : 다이나믹 프로그래밍 분기점이 있음

3개의 연산중 어떤 것을 하는 것이 더 큰 숫자를 얻는 지 전체 경우의 수를 탐색하는 문제이다. 연산을 하고나면 같은 구조의 문제가 되므로 재귀적으로 해결 할 수 있으며, 단계들의 각 결과가 다음 단계에 사용 될 수있다.

```
m ,n = map(int, input().split())
maps = [[int(i) for i in input().split()] for i in range(m)]
dx = [-1,0,-1]
dy = [0,-1,-1]
def isSafe(x,y):
    return x>= 0 and y>=0 and x<m and y<n
for i in range(0,m):
    for j in range (0,n):
        tmp =[]
        for k in range(3):
            nx = i+dx[k]
            ny = j+dy[k]
            if isSafe(nx,ny) :
                tmp.append(maps[nx][ny])
        if tmp != [] : maps[i][j] += max(tmp)

print(maps[m-1][n-1])
```

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

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

예제 출력

```
31
```

예제 입력 2

```
3 3
1 0 0
0 1 0
0 0 1
```

예제 출력 2

```
3
```

예제 입력 3

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

예제 출력 3

```
47
```

힌트

출처

- 문제를 만든 사람: baekjoon