

미로 탐색

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	192 MB	139626	60191	38686	41.825%

문제

N×M크기의 배열로 표현되는 미로가 있다.

1	0	1	1	1	1
1	0	1	0	1	0
1	0	1	0	1	1
1	1	1	0	1	1

미로에서 1은 이동할 수 있는 칸을 나타내고, 0은 이동할 수 없는 칸을 나타낸다. 이러한 미로가 주어졌을 때, (1, 1)에서 출발하여 (N, M)의 위치로 이동할 때 지나야 하는 최소의 칸 수를 구하는 프로그램을 작성하시오. 한 칸에서 다른 칸으로 이동할 때, 서로 인접한 칸으로만 이동할 수 있다.

위의 예에서는 15칸을 지나야 (N, M)의 위치로 이동할 수 있다. 칸을 셀 때에는 시작 위치와 도착 위치도 포함한다.

입력

첫째 줄에 두 정수 N, M($2 \leq N, M \leq 100$)이 주어진다. 다음 N개의 줄에는 M개의 정수로 미로가 주어진다. 각각의 수들은 **붙어서** 입력으로 주어진다.

출력

첫째 줄에 지나야 하는 최소의 칸 수를 출력한다. 항상 도착위치로 이동할 수 있는 경우만 입력으로 주어진다.

Code

```
from collections import deque

def bfs(graph, x, y, visited):
    queue = deque([(x, y, 1)])
    visited[x][y] = True
    # print(f'graph within bfs {graph}')
    dx = [-1, 1, 0, 0]
    dy = [0, 0, -1, 1]

    while queue:
        x, y, dist = queue.popleft()
        if y == N and x == M:
            print(dist)
            break
        else:
            for i in range(4):
                nx = x + dx[i]
                ny = y + dy[i]
                #print(f'nx: {nx}, ny: {ny}')

                if ((0 <= nx <= M) and (0 <= ny <= N)) and (
                    not visited[ny][nx]) and (graph[ny][nx] == "1"):
                    visited[ny][nx] = True
                    queue.append([nx, ny, dist + 1])
                    #print(queue)

    #print(queue)

N, M = map(int, input().split())
# N, M = 4, 6
# visited = [[False] * (M + 1)] * (N + 1)
visited = [[False] * (M + 1) for _ in range(N + 1)]
graph = [[''] * (M + 1)]
for i in range(N):
    graph.append(" " + input())
# print(f'visited: {visited}')
# print(f'graph: {graph}')
bfs(graph, 1, 1, visited)
```

The problem is to find the least amount of places in the maze needed in order to go to the lower-right corner from the upper-left corner. The maze is made of 1 and 0 which represents a place possible to go and a place impossible to go respectively. The challenging part of the problem was to accurately implement the BFS(Breadth First Search) algorithm in order to efficiently search for the least value.