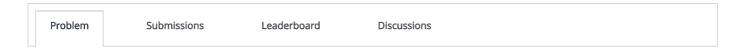


All Contests > Praktikum ASA Undip pertemuan 4 > Labirin

Labirin



Author: Albar N

Kamu diberi peta labirin, dan tugas kamu adalah menemukan jalur dari awal hingga akhir. Kamu bisa berjalan ke kiri(L), kanan(R), atas(U) dan bawah(D). Jalur yang dapat dilewati adalah '.'. Sedangkan '#' adalah dinding yang tidak bisa dilalui.

Input Format

- Baris input pertama memiliki dua bilangan bulat n dan m: tinggi dan lebar peta.
- Lalu ada n baris m karakter yang menggambarkan labirin. Setiap karakter adalah . (lantai), # (dinding), A (mulai), dan B (akhir). Tepatnya ada satu A dan satu B di input.

Constraints

1 < n, m < 1000

Output Format

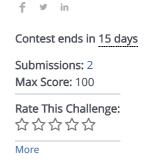
- Pertama cetak "Ya", jika ada jalur, dan "Tidak" jika tidak.
- Jika ada jalur, cetak panjang jalur terpendek tersebut dan deskripsinya sebagai string yang terdiri dari karakter L (kiri), R (kanan), U (atas), dan D (bawah).

Sample Input 0

```
5 8
#######
#.A#...#
#.##.#B#
#.....#
#######
```

Sample Output 0

Ya 9 LDDRRRRRU



```
2
3 n, m = map(int, input().split())
   maze = [list(input()) for _ in range(n)]
5
6 start = None
  end = None
7
8 vfor i in range(n):
9 ▼
        for j in range(m):
            if maze[i][j] == 'A':
10 🔻
                start = (i, j)
11
12 🔻
            elif maze[i][j] == 'B':
                end = (i, j)
13
14
15 ▼def bfs(maze, start, end):
        queue = deque([start])
16
        visited = set()
17
18
        directions = {'D': (1, 0), 'U': (-1, 0), 'R': (0, 1), 'L': (0, -1)}
19
        came_from = {}
20
        while queue:
21 1
22
            current = queue.popleft()
23
            if current == end:
24
                return came_from
25
            for direction, (dx, dy) in directions.items():
26
                next_pos = (current[0] + dx, current[1] + dy)
27
28
29
                    0 <= next_pos[0] < n and
                    0 \le \text{next_pos[1]} \le \text{m} and
30
31
                    maze[next\_pos[0]][next\_pos[1]] != '#' and
32
                    next_pos not in visited
33
                ):
                    queue.append(next_pos)
34
35
                    visited.add(next_pos)
36
                    came_from[next_pos] = (current, direction)
37
38
        return None
39
40 ▼def reconstruct_path(came_from, start, end):
41
        current = end
42
        path = []
43
        while current != start:
            current, direction = came_from[current]
44
            path.append(direction)
45
        path.reverse()
46
47
        return path
48
49 came_from = bfs(maze, start, end)
50 vif came_from is None:
        print('Tidak')
51
52 ▼else:
        path = reconstruct_path(came_from, start, end)
53
54
        print('Ya')
55
        print(len(path))
        print(''.join(path))
56
                                                                                                Line: 1 Col: 1
```

<u>♣ Upload Code as File</u> Test against custom input

Run Code

Submit Code

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |