

PREPARE^{NEW}

CERTIFY

COMPETE

Search



C_24060121140112 ▾

[All Contests](#) > [Praktikum ASA Lab C](#) > [Pertemuan 6 - Newbie Explorer \(BFS\)](#)

Pertemuan 6 - Newbie Explorer (BFS)

Problem

Submissions

Leaderboard

Discussions

Kamu adalah seorang penjelajah labirin pemula. Sebagai latihanmu, guild menugaskanmu untuk menjelajah sebuah labirin pemula yang setiap jalannya terbentuk dari beberapa blok ruangan. Disini kamu diharuskan untuk mencapai ruang teleportasi keluar yang ada di dalam labirin tersebut. Sebagai tugasnya, kamu disuruh untuk menuliskan berapa blok yang telah kamu lewati, jumlah blok baru yang kamu lihat ketika kamu berjalan, serta ke arah mana saja kamu berjalan.

Hint: Gunakan BFS

Input Format

1. Ukuran labirin yaitu jumlah blok arah sumbu x dan y (w, h)
2. Posisi blok awalmu (x1, y1)
3. Posisi ruang teleportasi (x2, y2)
4. Jumlah blok yang tertutup / penghalang (N)
5. Posisi blok-blok yang tertutup

Constraints

Dipastikan w dan h adalah integer positif maksimal 5, dan tidak ada titik-titik yang sama atau diluar batas labirin.

Output Format

1. Jumlah blok yang kamu lewati
2. Jumlah blok yang kamu lihat
3. Arah kamu berjalan (dinyatakan dalam arah mata angin U, T, S dan B)

Sample Input 0

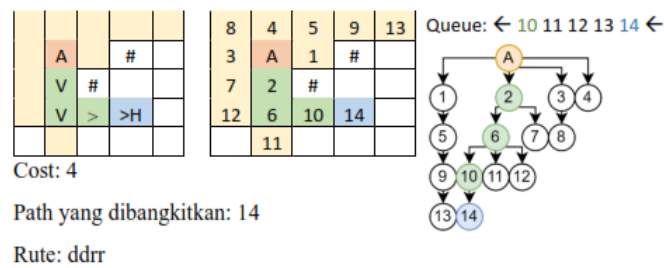
```
5 5
2 2
4 4
2
3 3
4 2
```

Sample Output 0

```
4
14
SSTT
```

Explanation 0

A adalah kamu, # adalah penghalang, dan H adalah ruang teleportasi.



Sample Input 1

3 3
1 1
3 3
0

Sample Output 1

4
8
TTSS

[f](#) [t](#) [in](#)

Contest ends in 5 hours

Submissions: 11
Max Score: 100

Rate This Challenge:
☆☆☆☆☆

[More](#)

Python 3

```
1 from collections import deque
2 from queue import Queue
3
4 def bfs(maze, start, end):
5     # inisialisasi queue dan visited
6     queue = deque([start])
7     visited = set([start])
8     dist = {start: 0}
9     directions = {start: ''}
10    length = 0
11    # iterasi selama queue tidak kosong
12    while queue:
13        current = queue.popleft()
14        x, y = current
15        # cek apakah sudah mencapai titik tujuan
16        if current == end:
17            return dist[current], length, directions[current]
18        length += 1
19        # cek seluruh arah yang mungkin
20        for dx, dy, direction in [(1, 0, 'T'), (0, 1, 'S'), (-1, 0, 'B'), (0, -1, 'U')]:
21            next_x, next_y = x+dx, y+dy
22            next_pos = (next_x, next_y)
23
24            # cek apakah selanjutnya berada di dalam labirin dan belum pernah dikunjungi
25            if (0 <= next_x < len(maze) and 0 <= next_y < len(maze[0]) and maze[next_x][next_y] !=
26            '#' and next_pos not in visited):
27                # tambahkan ke queue, visited, dan update jarak dan arah
28                queue.append(next_pos)
29                visited.add(next_pos)
30                dist[next_pos] = dist[current] + 1
31                directions[next_pos] = directions[current] + direction
32
33    # jika tidak ada jalan yang ditemukan, kembalikan nilai -1
```

```
33     return -1, -1, ''
34
35 # membaca input
36 h, w = map(int, input().split())
37 x1, y1 = map(int, input().split())
38 x2, y2 = map(int, input().split())
39 n = int(input())
40 maze = [['.' for _ in range(w)] for _ in range(h)]
41 for _ in range(n):
42     x, y = map(int, input().split())
43     maze[x-1][y-1] = '#'
44
45 # menjalankan BFS dari posisi awal ke posisi tujuan
46 result = bfs(maze, (x1-1, y1-1), (x2-1, y2-1))
47
48 # output hasil
49 if result[0] == -1:
50     print("Tidak ada jalan yang ditemukan")
51 else:
52     print(result[0])
53     print(result[1])
54     print(result[2])
```

Line: 1 Col: 1

 [Upload Code as File](#) ☐ [Test against custom input](#)[Run Code](#)[Submit Code](#)[Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |