Description | Hints | Submissions

Snakes and Ladders

Difficulty	Time Limit	Memory	Score
[illegible]	1 sec	256000KB	0/100

Description

Abhishek loves Snakes and Ladders game, he can always roll the die to whatever number he want between 1 to 6. Help him to find the least number of dice rolls to reach the destination i.e the 100^{th} cell from the 1^{st} cell.

Rules :-

1. The game is played with a cubic die of 6 faces numbered 1 to 6.
2. Starting from square 1, land on square 100 with the exact roll of the die. If moving the number rolled would place the player beyond square 100, no move is made.
3. If a player lands at the base of a ladder, the player must climb the ladder. Ladders go up only.
4. If a player lands at the mouth of a snake, the player must go down the snake and come out through the tail. Snakes go down only.

Input Format

The first line contains the number of tests, t.

For each test case:

- The first line contains n, the number of ladders.
- Each of the next n lines contains two space-separated integers, the start, and end of a ladder.
- The next line contains the integer m, the number of snakes.
- Each of the next m lines contains two space-separated integers, the start, and end of a snake.

Output Format

For each of the t test cases, print the least number of rolls to move from start to finish on a separate line. If there is no solution, print −1.

Constraints

$1 \le t \le 10$

$1 \le n, m \le 15$

C++14 | 12 px

```cpp
65        // Add edges for possible dice rolls with cost of 1
66        for (int i = 1; i <= 100; i++) {
67            for (int j = 1; j <= 6; j++) {
68                int target = i + j;
69                if (target <= 100) {
70                    g[i].emplace_back(go_to[target], 1);  // G
                      (considering ladders/snakes)
71                }
72            }
73        }
74
75        Dijkstra(1);  // Start Dijkstra from node 1
76
77        // If `dis[100]` is still 1e9, it means node 100 is un
78        if (dis[100] == 1e9) {
79            cout << -1 << endl;  // Impossible to reach cell 1
80        } else {
81            cout << dis[100] << endl;  // Minimum rolls to rea
82        }
83    }
84
85    int main() {
86        ios_base::sync_with_stdio(0);
87        cin.tie(0);
88        cout.tie(0);
89
90        int t;
91        cin >> t;
92        while (t--) {
93            solve();
94        }
95
96        return 0;
97    }
98
```

Sample Tests | Manual Tests

Test Case 1

Console | Run on Sample | [illegible] | Next