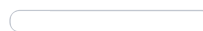


[All Domains](#) > [Algorithms](#) > [Graph Theory](#) > [Snakes and Ladders: The Quickest Way Up](#)

Badge Progress



Rank: 39473 Points: 262.00

# Snakes and Ladders: The Quickest Way Up

 by [idlecool](#)

Problem

Submissions

Leaderboard

Discussions

## Problem Statement

Markov takes out his [Snakes and Ladders](#) game and stares at the board, and wonders: If he had absolute control on the die, and could get it to generate any number (in the range 1-6) he desired, what would be the least number of rolls of the die in which he'd be able to reach the destination square (Square Number 100) after having started at the base square (Square Number 1)?

## Rules

1. Markov has total control over the die and the face which shows up every time he tosses it. You need to help him figure out the minimum number of moves in which he can reach the target square (100) after starting at the base (Square 1).
2. A die roll which would cause the player to land up at a square greater than 100, goes wasted, and the player remains at his original square. Such as a case when the player is at Square Number 99, rolls the die, and ends up with a 5.
3. If a person reaches a square which is the base of a ladder, (s)he has to climb up that ladder, and he cannot come down on it. If a person reaches a square which has the mouth of the snake, (s)he has to go down the snake and come out through the tail - there is no way to climb down a ladder or to go up through a snake.

## Constraints

The board is always of the size 10 x 10 and Squares are always numbered 1 to 100.

$1 \leq T \leq 10$

$1 \leq \text{Number of Snakes} \leq 15$

$1 \leq \text{Number of Ladders} \leq 15$

Square number 1 and 100 will not be the starting point of a ladder or a snake.

No square will have more than one of the starting or ending point of a snake or ladder (e.g. snake 56 to 44 and ladder 44 to 97 is not possible because 44 has both ending of a snake and a starting of a ladder)

## Input Format

The first line contains the number of tests, T. T testcases follow.

For each testcase, the first line contain N(Number of ladders) and after that N line follow. Each of the N line contain 2 integer representing the starting point and the ending point of a ladder respectively.

The next line contain interger M(Number of snakes) and after that M line follow. Each of the M line contain 2 integer representing the starting point and the ending point of a snake respectively.

## Output Format

For each of the T test cases, output one integer, each in a new line, which is the least number of moves (or rolls of the die) in which the player can reach the target square (Square Number 100) after starting at the base (Square Number 1). This corresponds to the ideal sequence of faces which show up when the die is rolled.

If there is no solution, print **-1**.

### Sample Input

```
2
3
32 62
42 68
12 98
7
95 13
97 25
93 37
79 27
75 19
49 47
67 17
4
8 52
6 80
26 42
2 72
9
51 19
39 11
37 29
81 3
59 5
79 23
53 7
43 33
77 21
```

### Sample Output

```
3
5
```

### Explanation

*For the first test:* To traverse the board via the shortest route, the player first rolls the die to get a 5, and ends up at square 6. He then rolls the die to get 6. He ends up at square 12, from where he climbs the ladder to square 98. He then rolls the die to get '2', and ends up at square 100, which is the target square. So, the player required 3 rolls of the die for this shortest and best case scenario. So the answer for the first test is 3.

Copyright © 2015 HackerRank.  
All Rights Reserved

Submissions: 3581

Max Score: 50

Difficulty: Easy

[More](#)

Current Buffer (saved locally, editable)  

C++



```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 using namespace std;
7
```

```
8
9 int main() {
10     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
11     return 0;
12 }
13
```

Line: 1 Col: 1

[Upload Code as File](#)

Test against custom input

Run Code

Submit Code

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Privacy Policy](#) | [Request a Feature](#)