July 2021 CSE204: Data Structures and Algorithms I Sessional

Offline on Graphs and Graph Traversals

Xavi has a special board for the Snakes and Ladders game with a unique die. His board has ***X*** squares and the die has ***n*** faces. One day he takes out his game, stares at the board, and wonders: "If I can always roll the die to whatever number I want, what would be the least number of rolls to reach the destination?"

Rules: The game is played with a die of ***n*** faces numbered ***1*** to ***n***.

Starting from square 1, land on square ***X*** with the exact roll of the die. If moving the number rolled would place the player beyond square ***X***, no move is made.

If a player lands at the base of a ladder, the player must climb the ladder. Ladders go up only.

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/Output:

You will take input from an input file and give output to an output file. Keep provision of using the standard I/O. [Hint: Use of I/O redirection conveniently handles this task]

Input Format:

The first line contains the number of test cases, ***t***.

For each test case:
- The first line contains two space-separated integers, the number of faces in the die(***n)***, and the size of the board(***X***)
- The next line contains the number of ladders, ***l***.
- Each of the next ***l*** lines contains two space-separated integers, the start and end of a ladder.
- The next line contains the integer ***s***, the number of snakes.
- Each of the next ***s*** lines contains two space-separated integers, the start and end of a snake.

Output Format:

For each of the ***t*** test cases, print three (3) lines.
In the first line, print the least number of rolls needed to move from ***1*** to ***X***. If there is no solution, print -1.
In the second line, print the shortest path. If there is no solution, print "No solution".
In the third line, print the unreachable squares from ***1***. If no square is unreachable, print "All reachable".

See the sample I/O for further clarification.

Constraints:

$1 < t \leq 10$
$1 < n \leq 16 \; and \; n \; is \; even$
$1 < X \leq 10000$
$1 < l, s \leq 50$

Neither square ***1*** nor square ***X*** will be the starting point of a ladder or snake.
A square will have at most one endpoint from either a snake or a ladder.

Special Instructions:

Write ***readable, re-usable, well-structured, quality*** code. This includes but is not limited to writing appropriate functions for implementation of the required algorithms, meaningful naming of the variables, suitable comments where required, proper indentation, etc.

Please **DO NOT COPY** solutions from anywhere (your friends, seniors, internet, etc.). Implement the algorithms with your style of coding. Any form of plagiarism (irrespective of source or destination), will result in getting -100% marks. You have to protect your code.

Also, be informed that for the repeated offense of plagiarism, the departmental policies suggest stricter measures.

Submission Guideline:

1. Create a directory with your 7 digit student id as its name
2. Put the source files only into the directory created in 1
3. Zip the directory
4. Upload the zip into moodle

   For example, if your student id is 1905123, create a directory named 1905123. Put your source files(.c, .cpp, .java, .py, .h, .hpp etc) only into 1905123. Zip 1905123 into 1905123.zip and upload the 1905123.zip into moodle.

   Failure to follow the above-mentioned submission guideline will result in up to 10% penalty.

Submission Deadline:

January 21, 2022, 11:45 PM

This is a hard deadline and there shall be no extensions for any reason whatsoever**.**

The sample I/O:

Input	Output
2 3 20 1 2 8 2 10 7 15 3 6 30 3 2 10 15 19 11 16 7 20 3 21 4 22 5 23 6 24 7 25 8 26 9	6 1 -> 2 -> 8 -> 11 -> 14 -> 17 -> 20 All reachable -1 No solution 26 27 28 29 30