

Project 3: Graph modeling and graph algorithms

COT 4400, Fall 2021

Due December 3, 2021

1 Overview

This project requires you to model the problem below as graph and then use a known graph algorithm to solve the problem. **You are not allowed to use the internet or consult any references. This is an individual project. This policy will be strictly enforced.**


This problem is based on the “Tarzan and Jojo” maze problem (from “MAD MAZES: Intriguing Mind Twisters for Puzzle Buffs, Game Nuts and Other Smart People” by Robert Abbott). The text of the problem is quoted below. A diagram of the maze is provided on the next page.

Tarzan was spending a peaceful day in the jungle when his friend Jojo the chimpanzee began taunting him. “You can’t catch me, ape-man,” shouted Jojo. Tarzan, always one to enjoy a good chase, began swinging after him, only to find that Jojo had tangled up all of the hanging tree vines. Therefore, as Tarzan swings through the jungle, he can only move in the direction of the arrow in the square at the beginning of each swing. And because of the length of the vines, each swing must carry him exactly three or four squares.

Tarzan begins on the square at the top. From there he can travel three squares to A or go four squares to B. Suppose he goes to square B. On the next turn he can only go three squares (from B it is impossible to travel four squares). From square C he can go three squares to D or four squares to E.



Jojo has hidden in the square at the bottom of the maze of vines. How can Tarzan get to that square? (Note that only one square, the one marked F, will enable Tarzan to swing onto Jojo’s square.





↓
↓

→	↓	↓	↖	←	→	↓	↓	←									
→	↓	A →	→	→	←	↓	↓	↓									
↓	↓	B ↓				↓	↓	↓									
↓	↑	↑				↑	↑	↘									
↑	↑	→	←	←	→	↑	←	↓									
→	→	C →	↖	→	D ↗	E ←	←	↓									
↓	→	↓	↖	→	↘	←	↓	←									
→	↓	↓	←	←	→	←	↓	↓									
↓	↓	↓				↓	↓	↑									
↑	↑	↑				↑	↑	↑									
↑	→	↑	→	←	→	F ↓	←	←									
→	↑	→	→	→	↗	↑	←	↑									

2 Modeling the problem

Before you write a program to solve this problem, you will first write a report describing (in English and pseudocode) how you will solve this problem. This report should answer two basic questions:

1. What type of graph would you use to model the problem input (detailed in the Section 3.1), and how would you construct this graph? (I.e., what do the vertices, edges, etc., correspond to?) Be specific here; we discussed a number of different types of graphs in class.
2. What algorithm will you use to solve the problem? Be sure to describe not just the general algorithm you will use, but how you will identify the sequence of moves Tarzan must take in order to reach the goal.

3 Coding your solution

In addition to the report, you should implement your algorithm in C++ or Java so that it can solve “Tarzan and Jojo” mazes. Your code may be in C++ or Java, but it must compile and run on the C4 Linux Lab machines.

Your code may be split into any number of files. In addition, you are allowed to make use of any built-in library, and C++ users may use the Boost library in their implementations. Boost is a free, open-source library with a rich collection of mathematical functions, including several that deal with graphs. You may read more about Boost at www.boost.org.

3.1 Input format

Your program should read its input from the file `input.txt`, in the following format. The input begins with a two positive integers on a line indicating the number of rows r and columns c of the maze, respectively. The next line contains two positive integers, s_r and s_c , representing the row and column at which Tarzan starts. (s_r will be in the range 1 to r , while s_c will be in the range 1 to c .)

The following r lines contain the directional information for each vine in the maze. Each line has c entries, where each integer represents one of three things: (1) the direction of the vine at that location (‘N’, ‘E’, ‘S’, ‘W’, ‘NE’, ‘NW’, ‘SE’, or ‘SW’), (2) a location with no vine (‘X’), or (3) the location of Jojo (‘J’).

We will not test your code on ill-formatted input, but validating the input may be useful if you develop any test cases on your own.

For the original “Tarzan and Jojo” maze, the input is:

```

23  9
1   3
X  X  S  X  X  X  X  X  X
X  X  S  X  X  X  X  X  X
E  S  S  SW W  E  S  S  W
E  S  E  E  E  W  S  S  S
S  S  S  X  X  X  S  S  S
S  N  N  X  X  X  N  N  NW
N  N  E  W  W  E  N  W  S
E  E  E  NW E  NE  W  W  S
S  X  X  X  X  X  X  X  N
S  X  X  X  X  X  X  X  S
N  X  X  X  X  X  X  X  S
N  X  X  X  X  X  X  X  N
S  X  X  X  X  X  X  X  S
S  X  X  X  X  X  X  X  S
N  X  X  X  X  X  X  X  N
N  E  S  SW E  SE  W  S  W
E  S  S  W  W  E  W  S  S
S  S  S  X  X  X  S  S  N
N  N  N  X  X  X  N  N  N
N  E  N  E  W  E  S  W  W
E  N  E  E  E  NE  N  W  N
X  X  X  X  X  X  N  X  X
X  X  X  X  X  X  J  X  X

```

The maximum dimensions for a maze are 1000 by 1000.

3.2 Output format

Your program should write its output to the file `output.txt`, in the following format. The output should consist of a path from Tarzan’s start location to Jojo. It should be a single line consisting of a sequence of moves, separated by spaces. Each move should be a string composed of two parts, separated by a hyphen (-). The first part should be the direction of the move (N, E, S, W, NE, NW, SE, or SW), and the second part should be the distance of the move (3 or 4). Beginning from Tarzan’s start location, this sequence of moves must match the direction of every vine Tarzan swings from and end at Jojo, and it should not leave the maze or encounter a vine marked with an ‘X’. If there are multiple valid paths through the vine maze, you may output any such path.

For example, if your first three moves take you 4 spaces south, 3 spaces south, and 3 spaces east, your output should begin with `S-4 S-3 E-3`.

You are welcome to try figuring out the solution to the “Tarzan and Jojo” puzzle on your own, but that won’t get you any points. Your assignment is to model the maze as a graph and to solve the problem using an appropriate graph algorithm.

4 Submission

You must submit a zip archive containing 1) your report (described in Section 2) as a PDF document, 2) your code (described in Section 3), and 3) a README file describing how to compile and run your code to Canvas. If your code requires more than a simple command to compile and run

then you must also provide a Makefile and/or shell script. A simple command might be something like:

```
g++ *.cpp -o maze
```

If you are using Boost in your solution, you must provide a Makefile and/or shell script that uses the environment variable `$BOOST_HOME` (pointing to the Boost installation directory) to compile your code.

As this is an *individual* project, your project report and code will be checked for plagiarism.

5 Grading

Report	50 points
Graph model	30
Algorithm description	20
Code	50 points
README file	5
Follows input and output specs	10
Compiles and is correct	30
Good coding style	5

Note: if your code is unreasonably slow, you will lose points for both your algorithm design and your correct output grade.