CS2040: Data Structures and Algorithms

Tutorial Problems for Week 12: Shortest Paths (I)

For: 7 November 2024, Tutorial 10

Problem 1. Unlock the lock

(UVa 12160 – `https://uva.onlinejudge.org/external/121/p12160.pdf`)

Mr. Ferdaus has created a special type of 4-digit lock named "FeruLock". It always shows a 4-digit value and has a specific unlock code U (An integer value between 0000 and 9999). The lock is unlocked only when the unlock code is displayed. This unlock code can be made to appear quickly with the help of some of the special buttons available with that lock. There are R ($1 \leq R \leq 10$) such special buttons and each button has a number (between 0000 and 9999) associated with it. When any of these buttons is pressed, the number associated with that button is added with the displayed value and so a new number is displayed. The display starts with some value L ($0000 \leq L \leq 9999$). The lock always uses least significant 4 digits after addition. After creating such a lock, he has found that, it is also very difficult for him to unlock the Ferulock. As a very good friend of Ferdaus, your task is to create a program that will help him to unlock the Ferulock by pressing these buttons minimum number of times. If there is no way to unlock the FeruLock simply output "Permanently locked".

L, U and the values of each special button will always be denoted by a 4 digit number (even if it is by padding with leading zeroes). Some examples are as follows:

- If $U = 9999, L = 0000, R = 1$ and the value of the button is 1000, then there is no way to unlock the FeruLock, so output "Permanently locked".
- If $U = 9999, L = 0000, R = 1$ and the value of the button is 0001, then the minimum number of button presses to unlock the FeruLock will be 9999.
- If $U = 1212, L = 5234, R = 3$ and the values of the buttons are 1023, 0101 and 0001 respectively, then the minimum number of button presses will be 48.

Problem 2. Escape Plan (2012/2013 CS2010 WQ2)

You work in a maze. Unfortunately, portions of the maze have caught on fire, and the owner of the maze neglected to create a fire escape plan. You need to escape QUICKLY. Given your location in the maze and which squares (can be zero, one, or more than one) of the maze that are initially on fire (when you first realized that there is a fire), you must determine **whether** you can exit the maze before the fire reaches you; and if you can, **how fast** you can do it.

Both you and any of the fire each move one square per minute, vertically or horizontally (not diagonally). The fire spreads in all four directions from each square that is on fire. You may exit

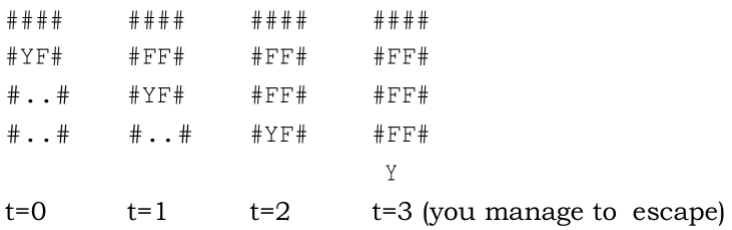
the maze from *any square that borders the edge of the maze*. Neither you nor the fire may enter a square that is occupied by a wall.

You are given two integers R and C ($1 \leq R, C \leq 1000$) and a 2D character array M with R rows and C columns that describes the starting condition of the maze. Each cell can contain either:

- '`#`', a wall. Neither you nor the fire may enter this cell.
- '`.`', a passable square (for you and the fire).
- '`Y`', your initial position in the maze, which is a passable square. There will be **exactly one** '`Y`' cell only in the test case.
- '`F`', a square that is currently on fire at this point of time. There can be **zero, one, or more than one** '`F`' cell(s) in the test case.

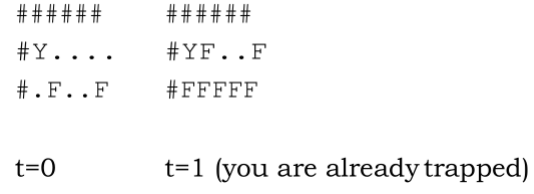
Your task is to implement the following function `EscapePlan` that takes in R, C, M and simply returns -1 if you cannot exit the maze before the fire reaches you, or a positive integer that gives the earliest time that you can safely exit the maze, in minutes.

Example 1: R1 = 4, C1 = 4, maze M1 is as shown below



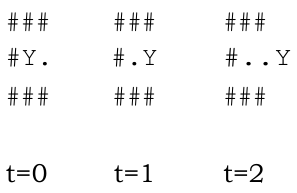
The function `EscapePlan(R1, C1, M1)` will return 3 because by running downwards three steps (in three minutes), you can escape the maze while the fire cannot catch you.

Example 2: R2 = 3, C2 = 6, maze M2 is as shown below



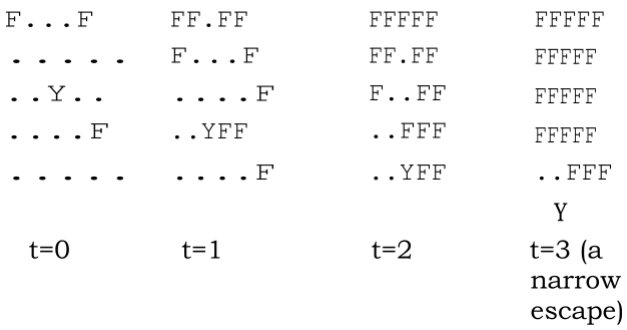
The function `EscapePlan(R2, C2, M2)` will return -1 because after 1 minute, the nearest fire already surrounds you. Note that if you and the fire reach the same cell at the same time, you die.

Example 3: R3 = 3, C3 = 3, maze M3 is as shown below



The function `EscapePlan(R3, C3, M3)` will return 2.

Example 4: R4 = 5, C4 = 5, maze M4 is as shown below



The function `EscapePlan(R4, C4, M4)` will return 3, a narrow escape as shown above.

Problem 2.a. What do the vertices and the edges of the graph represent?

Problem 2.b. What is the graph problem that we want to solve?

Problem 2.c. Give the most appropriate graph algorithm (or modified version of it) to solve this problem.

Problem 3. Money Changer (2011/2012 CS2010 WQ2)

Given n currencies and m exchange rates, determine if we can start with a certain amount of money in one currency, exchange this amount for other currencies, and end up at the same currency but with more money than what we had at first. Two examples are as follows:

- Suppose the money changer has n = 3 currencies and m = 3 exchange rates: 1 USD gives us 0.8 Euro; 1 Euro gives us 0.8 GBP (British pound sterling); and 1 GBP gives us 1.7 USD. So if we start with 1 USD, we can exchange it for 0.8 Euro, which can then be exchanged for 0.64 GBP, and if converted back to USD we have 1.088 dollars. We have just make a profit of 8.8 cents.

- Example 2: If the money changer has the following n = 2 currencies and m = 2 exchange rates: 1 USD gives us 0.8 Euro; and 1 Euro gives us 1.25 USD, then there is no way we can make a profit.

Can you model the n currencies and their m exchange rates as a graph problem and give an algorithm to report whether it is possible to start with any currency, exchange it with one (or more) other currencies, end with the same starting currency, and make a profit? State the time complexity of your algorithm.

Problem 4. Heights (2018/2019 Take-Home Lab 3)

Cats like to sit in high places. It is not uncommon to see cats climbing trees or furniture in order to lie on the top-most area within their feline reach. Rar the Cat is no exception. However, he does not know how high is one area relative to another.

Height can be measured in centimeters (cm) above sea level but Rar the Cat does not know the absolute height of any place. However, he knows that area B_i will be higher than area A_i by H_i centimetres because he needs to jump H_i to get from area A_i to B_i. There will be N areas in total with $N-1$ such descriptions. Areas are labelled from 1 to N and $0 < A, B \leq N$ where $A \neq B$.

Rar the Cat also has Q queries, each consisting 2 integers X and Y. He wants to know the height of area Y with respect to area X. Do note that $0 < X, Y \leq N$ but X can be equal to Y. In the event that area Y is lower than area X, please output a negative number. Otherwise, output a positive number.

It is guaranteed that the relative heights of all pairs of areas can be computed from the data provided in the input. **Design an algorithm that takes in the N areas and $N-1$ descriptions, and outputs the correct relative height for all Q queries efficiently**. Your time complexity should be dependent on both N and Q.