CST 370 – Homework 5

Due: **2:00pm on April 25, 2019**

Final Submission Deadline: **2:00pm on May 2, 2019**

Changelog:

- 4/25/19: Updated input description for Elevations
- 4/24/19: Specified order of outputs on Optimizing routes problem.

About

This homework assignment asks you to apply single source shortest path (path finding) algorithms to a variety of problems.

There are 5 problems in this homework **but you don't need to do all of them!** You must complete 40 points worth of problems. Any additional points you earn will count towards extra credit on the assignment (So completing the 3 15-point problems will give you 45/40).

Submissions

https://www.hackerrank.com/contests/cst370-s19-hw5/challenges

The problems for this homework will be hosted on HackerRank where automated tests will run. You may submit code on HackerRank as many times as you'd like. Once you are comfortable with your code, you can copy the submission link for that problem and include it in the ilearn assignment for me to grade. **I will only grade hackerrank links submitted on iLearn.** iLearn has videos describing how to get a submission link.

After the due date, you may continue working on problems on HackerRank. If you submitted something at the initial deadline, you may submit HackerRank links again on the iLearn assignment for resubmissions. Resubmissions will be assessed a 10% deduction.

Scoring

Your score on HackerRank or Kattis will give you *an approximation* of what your grade will be; however, I will read your code and determine your final grade myself, as detailed on the syllabus.

Problems

1	Grid Walk	10 points
2	Heuristics	10 points
3	Elevations	15 points
4	Optimizing Routes	15 points
5	Essential Services	15 points

1. Grid Walk (10 points)

https://www.hackerrank.com/contests/cst370-s19-hw5/challenges/grid-walk

Given a grid with obstacles, we want to find the shortest path from a starting position to an ending position using Dijkstra's algorithm. In this grid, you can move in all 8 directions: Up, Down, Left, Right, and the 4 diagonals.

Input

- The first line contains an integer, n, the dimension of the nxn grid.
- The second line contains two space-separated integers, the starting position in the order row column.
- The third line contains two space-separated integers, the ending position in the order row column.
- The next n lines contain the space separated row. Each element is either a O indicating no obstacle or a X indicating an obstacle.

Constraints

You can assume a path exists (and therefore that the starting and ending positions are Os) and that the starting and ending positions are in-bounds.

Output

The output contains a single line with an integer, the length of the shortest path (where each move, whether up, down, left, right, or diagonal) counts as 1 step.

Sample Input 1

```
4
0 2
3 1
X X O X
X X O O
X X X O
X O O O
```

Sample Output 1

```
4
```

Sample Input 2

```
6
0 1
5 2
O O O O O X
X X O X O O
O O X O O X
X O O O O O
O O X X X O
X O O X O O
```

Sample Output 2

```
5
```

2. Heuristics (10 points)

https://www.hackerrank.com/contests/cst370-s19-hw5/challenges/heuristics

Given a grid with obstacles, we want to find the shortest path from a starting position to an ending position. This time we'd like to use the A-Star algorithm. In this grid, you can only move in four directions: up, down, left, right therefore the heuristic that we will use will be the manhatten distance algorithm.

Input

- The first line contains an integer, n, the dimension of the nxn grid.
- The second line contains two space-separated integers, the starting position in the order row column.
- The third line contains two space-separated integers, the ending position in the order row column.
- The next n lines contain the space separated row. Each element is either a O indicating no obstacle or a X indicating an obstacle.

Constraints

You can assume a path exists (and therefore that the starting and ending positions are Os) and that the starting and ending positions are in-bounds.

Output

The output contains a single line with an integer, the length of the shortest path (where each move, whether up, down, left, or right) counts as 1 step.

Sample Input 1

```
4
0 2
3 1
X X O X
X X O O
X X X O
X O O O
```

Sample Output 1

```
6
```

Sample Input 2

```
6
0 1
5 2
O O O O O X
X X O X O O
O O X O O X
X O O O O O
O O X X X O
X O O X O O
```

Sample Output 2

```
12
```

3. Elevations (15 points)

https://www.hackerrank.com/contests/cst370-s19-hw5/challenges/elevations

I like running to different places but hate running uphill. I have a number of typical routes I run and I'd like to know the least elevation change it would take me to reach them. Help me find the elevation change to get from my house to each of the places I run to, taking only the routes I know. You will receive the list of pleces, the routes I know how to take, and the elevation change per route.

Starter code

The code below will help you read input for this problem.

- C++: https://repl.it/@dsyang/Elevations-cpp
- Java: https://repl.it/@dsyang/Elevations-java
- Python: https://repl.it/@dsyang/Elevations-py

Input

- The first line will contain a location, my home.
- The next line will contain an integer, r, the number of routes I run.
- Finally, the last r lines will contain the routes, consisting of comma-separated endpoints followed by the elevation change along that route.

Constraints

You cannot assume the graph is connected; there may be a place I cannot reach taking only routes that I know.

Output

- The output will consist of n lines.
- each line consists of a place name (in alaphabetical order), followed by a colon, a space, and the least elevation change to that place from my source.
- If there is no route to that place, print "NONE" instead.

See the sample output section and hackerrank for concrete examples

Sample Input 1	**Sample Output 1**
Home 9 Bookworks, Home, 95 Bookworks, Aquarium, -50 Home, Aquarium, 150 Home, Starbucks, 12 Home, Beach, -100 Beach, Starbucks, 55 Aquarium, Bookworks, 50 Starbucks, Carmel, 200 Carmel, Home, -100	Bookworks: 200 Home: 0 Aquarium: 150 Starbucks: -45 Beach: -100 Carmel: 155

4. Optimizing Routes (15 points)

https://www.hackerrank.com/contests/cst370-s19-hw5/challenges/optimizing-routes

I have my set ways of traveling to places I visit often. However, I'm curious if my routes are actually the best routes according to maps. Determine the difference in my route and the supposed best route for each of my destinations.

subsection*Starter code The code below will help you read input for this problem.

- C++: https://repl.it/@dsyang/Optimizing-Routes-cpp
- Java: https://repl.it/@dsyang/Optimizing-Routes-java
- Python: https://repl.it/@dsyang/Optimizing-Routes-py

Input

- The first line will contain a location, my home.
- The next line will contain an integer n, the number of places I visit often enough to have a set route, followed by how long my route to that place takes.
- The next line will contain an integer, r, the number of roads.
- Finally, the last r lines will contain the roads, consisting of comma-separated endpoints followed by the time it takes to travel the road.

Constraints

You can assume all the weights are positive (it takes time to travel a road) and that all locations have unique names. You may choose which appropriate path-finding algorithm to use to solve this problem.

Output

- The output will consist of n lines.
- Each line consists of a destination I visit often, followed by a space, followed by either "FASTEST" if my route is equal to or faster than the recommended route. "NO PATH" if there is no recommended route from my home to that location. Or an integer x representing how much faster the recommended route is.
- The n destinations will be ordered alphabetically, case-insensitive. This means Carmel will come before CSUMB.

See the sample output section and hackerrank for concrete examples

Sample Input 1

```
Home
4
Bookworks, 9
CSUMB, 16
East Village, 6
Carmel Beach, 21
17
Presidio, Bookworks, 3
Presidio, Aquarium, 3
Presidio, Colton Hall, 2
Presidio, Home, 5
Home, Carmel Village, 17
Home, REI, 12
REI, CSUMB, 3
Home, CSUMB, 16
Home, East Village, 6
Home, 17-Mile Drive, 10
Home, Colton Hall, 4
Home, Sushi Time, 10
CSUMB, Sushi Time, 7
Carmel Beach, 17-Mile Drive, 12
East Village, Colton Hall, 5
Colton Hall, Home, 4
Carmel Village, Carmel Beach, 4
```

Sample Output 1

```
Bookworks NO PATH
Carmel Beach FASTEST
CSUMB 1
East Village FASTEST
```

5. Essential Services (15 points)

https://www.hackerrank.com/contests/cst370-s19-hw5/challenges/all-pairs

King's Landing is trying to evaluate the quality of life of its citizens. One of the components is the time it takes each citizen to get to every essential service. Given the citizens and services, find the distance from each citizen to each service.

Starter code

The code below will help you read input for this problem.

- C++: https://repl.it/@dsyang/Essential-Services-cpp
- Java: https://repl.it/@dsyang/Essential-Services-java
- Python: https://repl.it/@dsyang/Essential-Services-py

Input

- The first line contains an integer, c, the number of citizens.
- The next c lines contain the citizens' names.
- The next line contains an integer, s, the number of services.
- The next s lines contain the services' names.
- The next line contains an integer, e, the number of direct routes.
- Each of the next e lines contains a direct route which consists of two comma-separated entities (citizens or servies) and a time it takes to get from one to the other.

Constraints

You can assume all the weights are positive (it takes time to get from one place to the other) and that all citizens and services have unique names.

Output

- The output will consist of $c + 1$ lines.
- The first line consists of the services in alphabetical order, space-separated.
- The next c lines consist of a citizen and their times from each service. You should have s space-separated integers, the time from each service in alphabetical order, followed by the citizen name (in alphabetical order).

See the sample output section and hackerrank for concrete examples

Sample Input 1	**Sample Output 1**
5 Cersei Robert Jamie Qyburn Tomin 2 Castle Church 9 Castle, Robert, 15 Robert, Qyburn, 155 Robert, Tomin, 15 Castle, Tomin, 35 Qyburn, Church, 15 Robert, Jamie, 30 Jamie, Church, 115 Robert, Cersei, 300 Cersei, Church, 60	Castle Church 220 60 Cersei 45 115 Jamie 170 15 Qyburn 15 145 Robert 30 160 Tomin