



Project 1: Shortest Path (Uninformed + Informed Search)

General Instructions for all problems

- No homework is accepted after the deadline
- You can use either Java or Python in your implementations.
- Please submit a document that shows your algorithm choice with a brief description of how it applies to the problem (1 to 4 sentences) and how to run your script.
- You must submit your implementation along with the document.

Input

You are given a random graph.

The “nodes” in a graph are located on a 10x10 2D “chess” board, consisting of 100 “squares”. Each square is itself of size 10 by 10 units. Squares are arranged like this:

0	1	2	3	...					9
10	11	12	13	...					19
...				...					
90	91	92	93	...					99

Each square itself is an area of size 10 x 10 units. So, we observe that if we only know which square the node falls in, that is itself not sufficient to identify the EXACT location of the node, but give some general guidance.

As a basic Geometry refresher, here is a basic Geometry fact: Two vertices that are in the same square can be up to approx. 14.1 units away. Another basic geometry fact: Two vertices that are in adjacent squares can be arbitrarily close to each other.

Input File Structure

Input consists of two files: v.txt and e.txt

v.txt includes the node data, which is vertex ID, and the “square” it falls on. This is given in this format:

```
1,72 // Vertex #1 falls in “square” 72
```

```
2,69 // Vertex #2 falls in “square” 69
```

e.txt includes the set of edges, which are fromVertex, toVertex, and the distance between them.

```
1,3,349.22 // Vertices 1 and 3 are 349.22 distance apart, respectively.
```

To Do

You should implement a program that reads this graph and takes two inputs: start and end points, and calculates the shortest path from start to end.

Implement an uninformed search (SSP/Dijkstra/APSP), and an informed search (A*) for this problem.

Compare the performance of uninformed and informed searches.