CS 180: Introduction to Artificial Intelligence

ME1 - Solution by Searching

Introduction

This machine exercise is an application of two algorithms frequently used in solving problems by searching - the greedy best-first search and the A* search.

Goal

Given a map, source, destination, and step costs for each action, find the path between the initial location and destination with the minimum path cost, using the greedy best-first search and A* search.

Input

The input of your program is a text file. The text file contains, in order, the map configuration, the two locations, and the step costs.

Map

The map is a matrix of zeros and ones. Ones correspond to *walls* while zeros correspond to *pathways*. No solution must contain a coordinate whose value is one. The dimension of the map is also encoded, preceding the matrix.

Locations

The two locations are represented by row-wise, zero-based coordinates.

```
Source: 2 2
Destination: 4 4
```

Step Costs

Each step cost will be encoded in a separate line.

```
Up: 4
Down 4
Left: 5
Right: 5
Diagonal: 6
```

Sample Input File

```
5 5
0 0 0 0 0
0 1 1 1 0
```

```
0 0 0 1 0
0 0 1 0 0
0 0 1 1 0
Source: 2 2
Destination: 4 4
Up: 4
Down 4
Left: 5
Right: 5
Diagonal: 6
```

Output

Your program must output the sequence of coordinates which make up the path with the minimum cost between the two locations. Output the sequences found by the two algorithms in two separate text files, named *astar.out* and *greedy.out*, respectively. Each coordinate in the sequence must be encoded in a separate line. After each sequence, output the total path cost.

Sample Output File

```
2 2
3 3
4 4
12
```

Submission

The deliverables for this ME are the following: the source code and a short documentation explaining your work. Email the deliverables to *kedelaspenas@up.edu.ph* with the subject *CS 180: ME1 <surname>*. Deadline of submission is on 16 December 2013 11:59PM.

Notes

- Work individually.

- Rules against cheating and plagiarism will be strictly observed.

- Use the Euclidean distance as heuristic for the A* search.