CMPE252: Artificial Intelligence and Data Engineering

Homework 1

Due: Sep. 13, 9PM

Instructions. You must work on it individually. Please follow the submission instructions exactly. You need to submit your assignment online by the due date. It is okay to scan your answers and create the pdf submission.

Problem 1 — 100 points

Implement Dijkstra's algorithm to solve for minimum cost paths on a given graph. We will consider only 2D grid graphs in this assignment. The edge costs are the distances between the two vertices. Each vertex can be connected to at most eight of its neighbors (N, NW, W, SW, S, SE, E, NE).

Files. Use your **SJSU Google account** to download supplemental files for this problem: `https://drive.google.com/drive/folders/12iGnOMY52mHQKYg-_nQ6jQG3lKweU6py?usp=sharing`

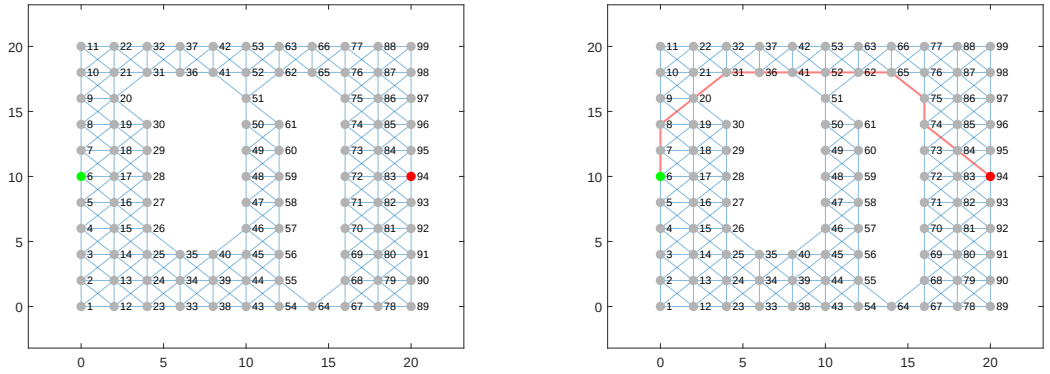
Input Description. You are given a grid environment. There are two input files corresponding to the given graph:

1. `input.txt`. This file has a specific format as given below:
   - The first line of the input file gives the total number of vertices, n, in the graph.
   - The second line gives the starting vertex index (indices go from 1 to n).
   - The third line gives the goal vertex index (indices go from 1 to n).
   - Starting from the fourth line, we have the edges in the graph specified in the form of an edge list. Each line specifies one edge: `i j wij` which indicates that there is a (directed) edge from `i` to `j` with a cost of `wij`. Note that this is a directed graph.
2. `coords.txt`. This file lists the x and y coordinates of the vertices starting with vertex numbered 1 on the first line.

Output Description. You may implement your algorithm using one of the following programming languages: MATLAB, C/C++, Python, or Java. The output file `output.txt` must have two lines that contains the vertices along the shortest path as well as the traveling cost from the starting vertex to the above corresponding vertices. The output file must have the following format:

- The first line must list the indices of the vertices on the shortest path only - from the start to the goal vertex. You need to choose the one with a smaller index if there are multiple options. For example, if there are two paths with indices " 1 3 2 ..." and "1 3 4 ..." that have the same traveling cost, you need to choose the first one.
- The second line must list the traveling cost from the starting vertex to the above corresponding vertices.

A sample input and output file are given on canvas. For example, the `input.txt` file contains 99 vertices. The `coords.txt` file specifies the actual coordinates of the 99 vertices starting with vertex numbered 1 to vertex numbered 99.



(a) (b)

Figure 1: Example

Submission Description. A sample input and output file are given. The requirements of your submission are as follows

1. You need to generate a video showing the result in each step using the provided inputs. Your video should demonstrate the result of your algorithm in each step that contains the following: the constructed graph and the paths that have been explored/updated until the current step. In the last step, you should plot the path you select according to the above criterion. Each part should be with a different color to differentiate. For example, in MATLAB, you can use the examples shown on this page (`https://www.mathworks.com/help/matlab/ref/graph.plot.html#buzeikk`) to display the graph. The video should be produced using code/package/plugin, rather than recorded by a external hardware (e.g., camera). Your video output should be named `hw1.mp4`. A sample image is shown in Figure 1.

2. You need to submit your source code. Your implementation must produce a single executable or single script named `hw1` that reads input file `input.txt` (will be put in the same directory as the executable or script) and produce a output file named `output.txt` in the same directory. You don't need to put the video-generating function into your code. You don't need to upload any `input.txt` or `output.txt` file. We will test your code on instances other than the provided sample. Make sure you follow the input/output conventions exactly.

3. You need to submit a pdf report named `hw1.pdf` describing your implementation. You should also include instructions on how to compile and run your code.

4. Please note that your code should ideally produce results within a few seconds when using the provided `input.txt` test case and no animation-generating function is included, given that the problem size is very small. During our code testing, if your code (without video-generating function) cannot generate a result within a 3-minute time-frame using the same level of problem size, your code submission will be deemed unsuccessful.

In conclusion, you submission should contain the following: `hw1.pdf`, `hw1.mp4`, and `hw1`. *Submission needs to be uploaded as a zip file.*