Artificial Intelligence and Data Engineering

Homework 2

Due: Sep. 29, 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.

Problem 1 — 100 points

Implement and compare the performance of *Dijkstra, A*, and Weighted A** algorithms to solve for minimum cost paths on a given graph.

Input Description. You are given a grid environment. You can use the two input files `input.txt` and `coords.txt` from the previous homework with the same format.

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 **12 lines** that contains the vertices along the shortest path as well as the traveling cost from the starting vertex to the above corresponding vertices for different algorithms, which will be specified later.

For the `output.txt` file, you should generate two lines for each of the following 6 algorithms with the same format as the previous hw:

1. Alg 1, line 1-2: Dijkstra,
2. Alg 2, line 3-4: A*,
3. Alg 3, line 5-6: weighted A* with $\epsilon = 2$,
4. Alg 4, line 7-8: weighted A* with $\epsilon = 3$,
5. Alg 5, line 9-10: weighted A* with $\epsilon = 4$,
6. Alg 6, line 11-12: weighted A* with $\epsilon = 5$.

Therefore, you should generate 12 lines, considering there are 6 algorithms here. You should use Euclidean distance as a heuristic for different A* algorithms.

In summary, you should generate two lines with the same format as the last homework for each algorithm. For example, for line 11, you should list the indices of the vertices on the generated path only - from the start to the goal vertex using weighted A* with $\epsilon = 5$. Ensure that the tie-breaking mechanism matches the one used in the previous homework. For line 12, you should list the traveling cost from the starting vertex to the above corresponding vertices using this algorithm.

Submission Description. You can refer to the `output.txt` file from the last to understand the format and the meaning of each of the two lines for each algorithm. But remember that you need to generate 12 lines for this homework. The requirements of your submission are as follows.

1. You need to generate a video showing the result in each step for different algorithms using the provided inputs. You can concatenate 6 different videos sequentially. However, a preferred way is to generate a video such that we can check the output of each algorithm at the same time for each time step. For example, you can generate a video that contains 2×3 (or 3×2) sub-videos such that each sub-video shows the results individually from each algorithm but with the same time-stamp. In the last step, your video should plot the final path generated by the corresponding algorithm. You should generate only **ONE** video named `hw2.mp4`.

2. You need to submit your source code. Your implementation must produce a single executable or single script named `hw2` that reads input file `input.txt` (will be put in the same directory as the executable or script) and produce an 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 `hw2.pdf` describing your implementation. You should also include instructions on how to compile and run your code. More importantly, you should generate a 2×6 table in this pdf file. The first row of the table should be the final cost associated with each algorithm. The second row of the table should be the total number of iterations taken by each algorithm to generate the corresponding output. You can add a table header if you want.

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 **for all the algorithms**, your code submission will be deemed unsuccessful.

In conclusion, you submission `hw2.zip` should contain the following: `hw2.pdf`, `hw2.mp4`, and `hw2`. *Submission needs to be uploaded as a zip file - hw2.zip.*