

MEAM 620 Homework 0

Find the Shortest Path!

Due: Wed, January 14th, 1:29 PM

1 Introduction

This assignment is a warmup exercise designed to get you to hit the ground running in MEAM 620! In the class we will need to solve path planning problems for robots. In preparation for this, you will familiarize yourself with graph search algorithms and implement Dijkstra's Algorithm, that yields the shortest distance path on a given weighted graph.

2 Background reading

Familiarize yourself with the graph search program and the algorithms used for (a) breadth-first search; (b) depth-first search; (c) Dijkstra's algorithm; and (d) A* search. Articles on wikipedia.org and the references therein are a great starting point. All the relevant files for your homework are available from the course website, <https://alliance.seas.upenn.edu/~meam620/wiki/>.

3 The assignment

(a) Implement Dijkstra's algorithm Write a MATLAB program and submit your code via **turnin**. Please see `shortestpath.m` for more details on inputs and outputs. Please note:

- In addition to returning the path, `shortestpath()` should return the total costs of returned path.
- One of the main considerations of any planner is how fast it is. Part of your grade will be determined by how quickly your planner runs. The [Matlab profiler](#) will help you considerably. Also see this [Mathworks page](#). For those of you know how to use C and MEX, please do **not** use those for this assignment.

(b) Discussion Provide a short (less than one page), **typed**, discussion of Dijkstra's Algorithm by comparing it to other graph search algorithms. Would (i) breadth-first search; (ii) depth-first search; and (iii) A* search have worked for this particular assignment? Why?

4 Grading

For this assignment your grade will be determined by automated tests. You can see an example of the types of tests we will run in `shortestpath_test.m`. These tests should run and pass (no command prompt outputs) if you type `shortestpath_test` from the Matlab prompt when you are in the directory that contains your code. We assure you that your code will be run against more tests than the ones we have given you.

5 Submission

(a) Dijkstra's algorithm When you are finished, upload your code to Eniac and submit via **turnin**. If you are unfamiliar with turnin, please see [this description](#) from CIS 520. All references to `cis520` should be

meam620 instead. The project name for this assignment is titled “assignment0” so the command to submit should be

“turnin -c meam620 -p assignment0 -v *”. Your turnin submission should contain:

1. A README file detailing anything we should be aware of.
2. Files `shortestpath.m` as well as any other Matlab files you need to run your code.

Shortly after submitting you should receive an e-mail from `meam620@seas.upenn.edu` stating that your submission has been received. You can check on the status of your submission at <https://alliance.seas.upenn.edu/~meam620/monitor/>. Once the automated tests finish running, you should receive another e-mail containing your test results. This report will only tell you whether you passed or failed tests; it will not tell you what the test inputs were or why your code failed. Your code will be given at most 5 minutes to complete all the tests. There is no limit on the number of times you can submit your code.

Please do not attempt to determine what the automated tests are or otherwise try to game the automated test system. Any attempt to do so will be considered cheating, resulting in a 0 on this assignment and possible disciplinary action.

(b) Discussion Please submit one printed page at the beginning of the first lecture. Please note the deadline and the policy for late assignments. All late submissions (homework and projects) except this first homework, will be penalized 20%. Thereafter, all submissions (including this first homework) will be penalized 2% each day.