UNIVERSITY OF COLORADO - BOULDER
Robotics Program

COEN 5830 (ROBO 5000) - Intro to Robotics

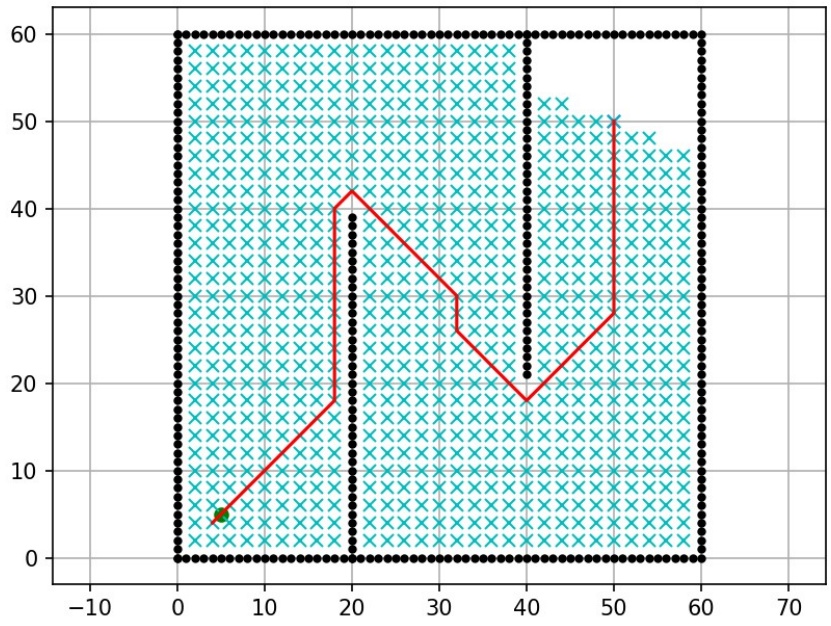
Homework #2 (Assigned: 9/22, Due: 10/4 11:59pm on *Canvas*)
Motion Planning

Instructions

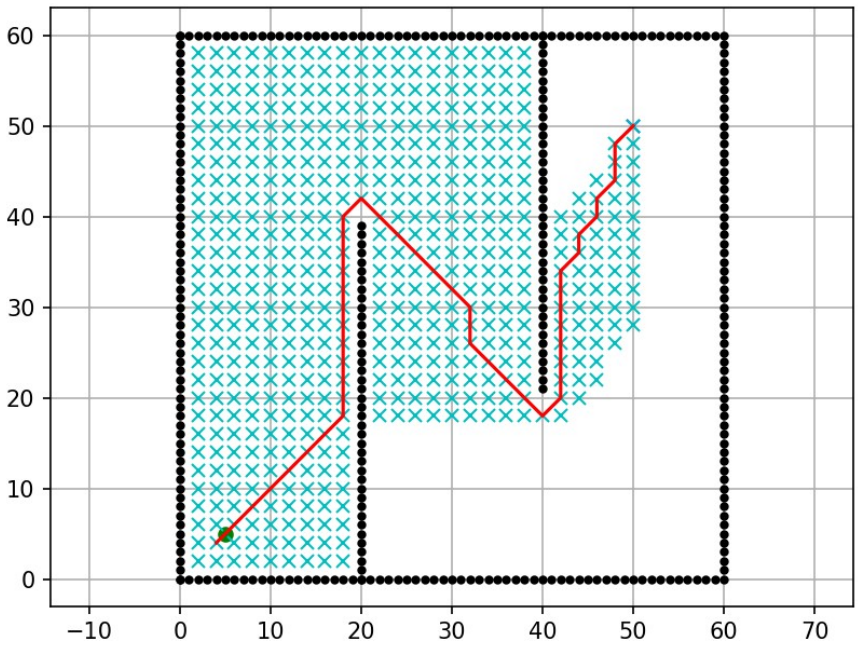
This homework assignment has a written component and all questions should be completed through Python code or commenting answers in the python file (no pdf submission is necessary). For written components, please add comments to the end of your file. Be sure to comment code so the grading staff can easily follow your logic. **You are encouraged to discuss your work with other students, but submitted work must be your own. Please indicate on the first page of your answer sheet who your collaborators for this homework are.**

Please complete all programming questions in a single .py file with the naming convention "Last-Name_FirstName.py". For example, my submission file would be called "Beuken_Leopold.py".

1. [5 points] A template has been provided to implement and visualize Dijkstra's algorithm via dijkstra.py. A brief psuedocode has been commented in the python file to help in the implementation. Finish the implementation such that the following output is obtained:



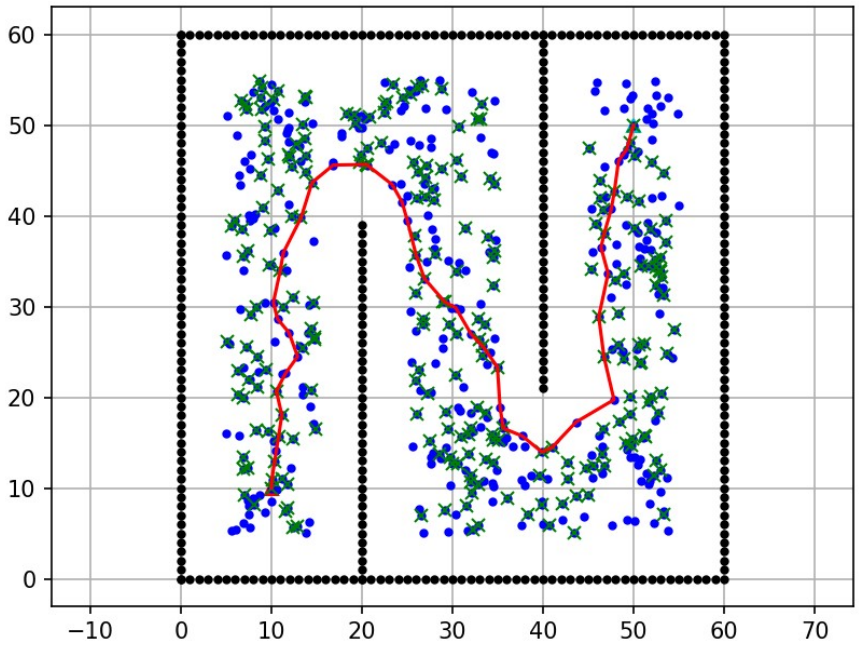
2. [5 points] A template has been provided to implement and visualize A* algorithm via a_star.py. A brief psuedocode has been commented in the python file to help in the implementation. Finish the implementation such that the following output is obtained:



3. [10 points] Change the environments for the A* and Dijkstra's implementation such that the time taken for both the algorithms is the same. Use python's time library to show that the time taken by both is approximately the same (± 0.2 seconds) . (Note: The heuristic function must not be altered.)

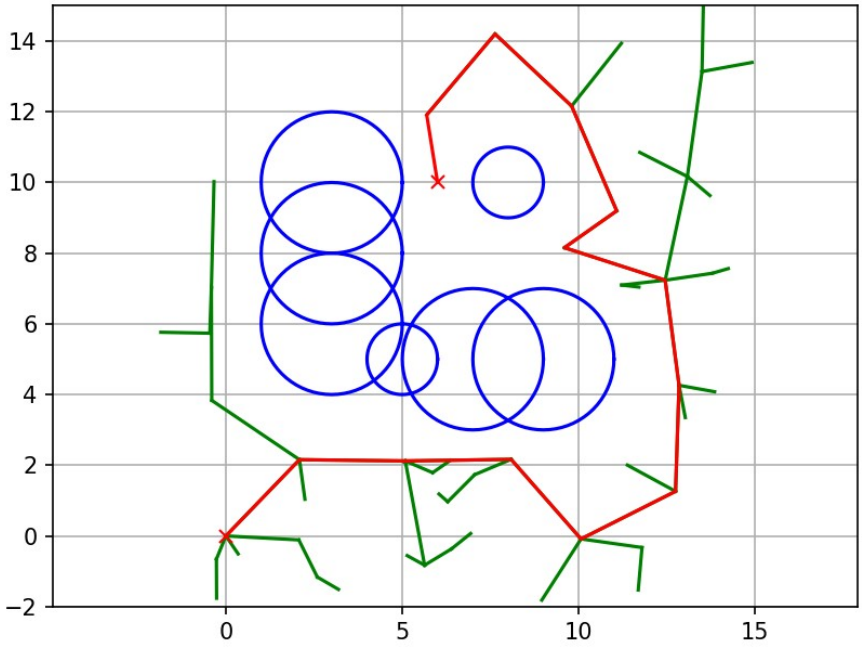
4. [2 points] Is it possible for A* to be slower than Dijkstra's at finding a solution? If so, give an example.

5. [5 points] In probabilistic_road_map.py, the code for PRM generation is already completed. Using the PRM, A template has been provided to use Dijkstra's on the existing PRM to find the solution. Finish the implementation such that a similar output is obtained (Since PRM is probabilistic, the exact output isn't expected. However, the path between the start and goal must be found):



6. [5 points] What is the average cost of the solution over 5 trials for the above implementation. What simple parameter change in PRM would result in the reduction of the average cost of solution provided by Dijkstra's? Demonstrate by changing the appropriate parameter in the code.

7. [10 points] A template has been provided to implement and visualize RRT algorithm via rrt.py. A brief psuedocode has been commented in the python file to help in the implementation. Finish the implementation such that a similar output is obtained (Since RRT is probabilistic, the exact output isn't expected. However, the path between the start and goal must be found):



8. [3 points] Does the above implementation work if the value of *expand_dis* is changed from 3.0 to 0.1? Why or why not? If not, what parameter can be changed to ensure the implementation works with expand_dis being 0.1?