# A* Implementation Documentation

Rose Gebhardt

February 15, 2022

## 1 Code Structure

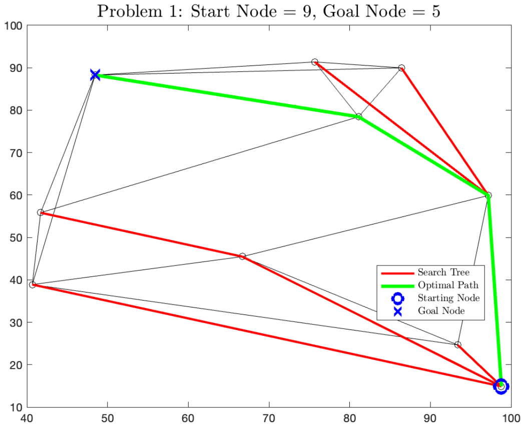
All files originated from the provided C++ starter code.

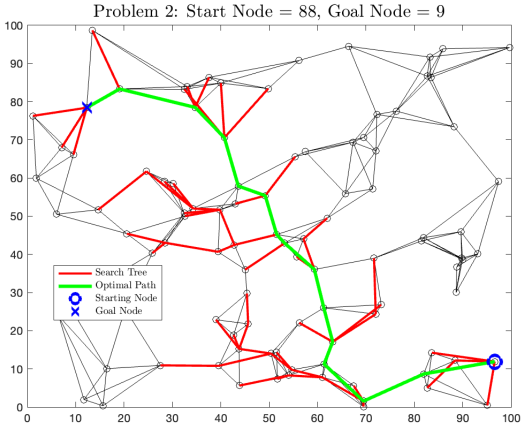
- `heap.cpp` and `heap.h` files are used to create a priority queue class. The only changes made to these files were the suppression of some print statements.
- `graph.h` is used to create a graph class which can read and parse input data files, store the data in the appropriate structure, and write the resulting search tree and path to a text file. The only changes from the original are the addition of `costStart` attribute to each node, all initialized at zero, and the suppression of some print statements.
- `simple_graph_search.cpp` uses the previous two data structures to search for a path. This file originally implemented a random search but was changed to run A*. The heuristic used was the Euclidean distance between the node from the goal.
- `display_search.m` plots the nodes and edges, the search path, the optimal path, and the start and goal nodes. This file was edited slightly to loop through five problems, mark the start and goal nodes, and to save the figures as `.png` images.
- `heap_test.cpp` and `plot_graph.m` are used for debugging, but not directly needed to running A* or plotting the results. These files were not changed from the original.

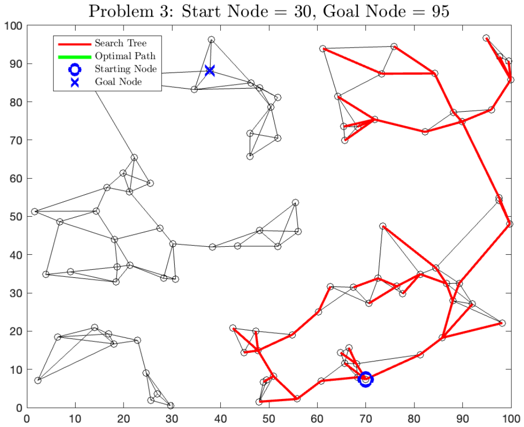
To get the search path and optimal path in a text file, run `./run_simple_graph_search` in terminal. To plot these results, run the `display_search.m` MATLAB script.

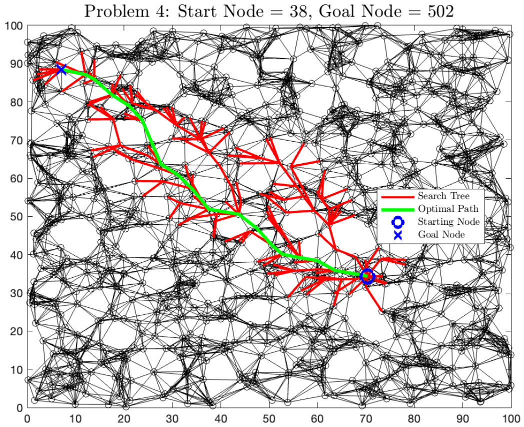
## 2 Results

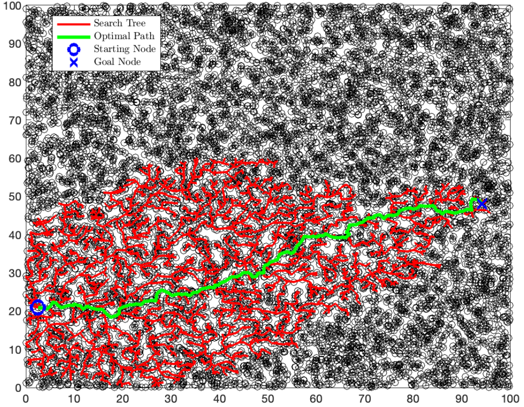
Results from each problem are shown below. The only case that did not have a solution was problem 3, which should be indicated in the terminal when running the executable.


Problem 1: Start Node = 9, Goal Node = 5


Problem 2: Start Node = 88, Goal Node = 9


Problem 3: Start Node = 30, Goal Node = 95


Problem 4: Start Node = 38, Goal Node = 502


Problem 5: Start Node = 38, Goal Node = 5002