Citations

For this project, I used the following resources to understand how to implement A* search in the grid:
http://www.redblobgames.com/pathfinding/a-star/implementation.html - gave an idea of using queues to pop the latest nodes. Has a small mockup implementation of the A* search.

https://www.youtube.com/watch?v=KNXfSOx4eEE - very helpful video that demonstrates how A* search actually works in the grid search with obstacles. Has similar Concepts with some minor details.

Part 2

1) As we change the value of w, we observe that the number of expansions start to vary. The larger the value of w, the smaller is the number of nodes our search expands by looking at the way the robot behaves.
2) W=1 gives us weighted A* search from the results of the path traversed by the robot. If we look at the formula (w=1) gives us f(s) = g(s) + 1 * h(s) which is essentially the formula for the weighted A* search. In the given configuration though, it seems like for all w values where w>=1, the robot is guaranteed to find the shortest path where. Consequently, uniform cost search expands more nodes than the rest of all algorithms because it doesn't account for heuristics and looks only at the costs, and since cost doesn't necessarily lead to the target, its path as a result gets longer.
3) As w increases, the number of expansions decreases. It follows the pattern of Greedy Best-First Search where it looks only at the best h(s) value possible without considering the cost known so far. As a result, it expands the most promising node and picks choice that is the closest to the target. Hence, as we increase the value of w, we decrease the chances of getting the path with the least cost but expand less nodes at the same time. For the given configuration, Greedy actually gets to the goal faster than A* search. Uniform cost search has the worst results because it expands too many nodes due to absence of heuristics.
4) During the experiment, I believe that the path length should be the same for all w >= 1 because it acts as A* in that case, and A* gives the optimal path if its heuristic is admissible and consistent with accordance to our implementation. But when w = 0, we have larger path length due to the way uniform cost search formes new nodes in the path.