* 1) Briefly explain how the algorithm works (N.B. This is a very common interview question).
* 2) What is an example of an admissible heuristic that would help with implementing an informed  
  search algorithm for this problem?
* 3) Show sample output from step 1 (a 4x4 cost matrix given a start and destination state)
* 4) Show sample output from step 2 for the example used in Q2.
* 5) Roughly how much time did you spend programming this lab?

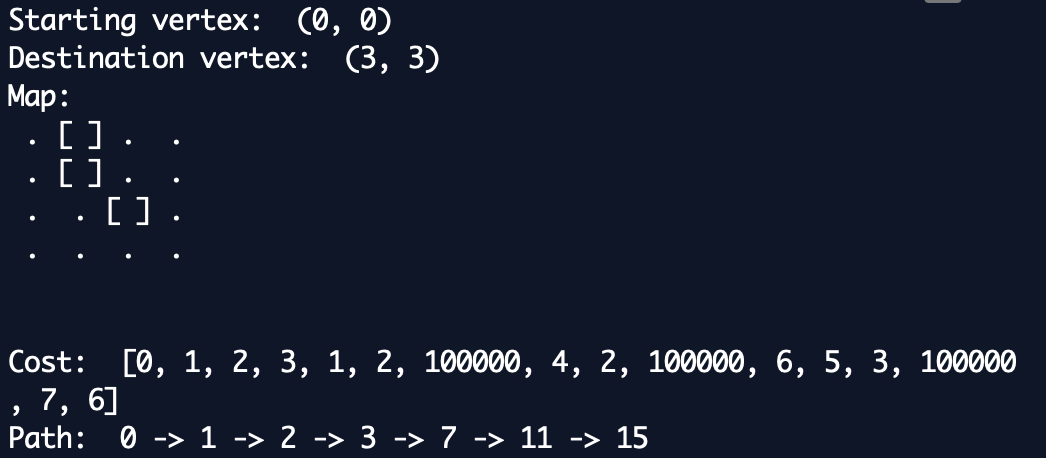
Lab 6

Lucas Laughlin: [lula5767@colorado.edu](mailto:lula5767@colorado.edu)

Casey Tran: catr5879@colorado.edu

Jordan Smart: [josm9339@colorado.edu](mailto:josm9339@colorado.edu)

1. The algorithm starts by setting the distance to each node from the starting node to INF except the starting node which is 0. The algorithm then hits every node with the shortest distance first and updates the cost of every neighbor of the node if the cost of the current node plus the cost to the neighbor is less than the current cost of the neighbor node. If a neighbor node cost is updated then the current node is then stored as previous node for the neighbor. Just before the algorithm is complete the path is constructed by adding the previous node of the destination node to list then adding previous node of the previous node and so on until we reach the starting node. We then reverse the list and we have a path.
2. Placing an inversely proportional weight on the bearing error so that paths that are closer to a bearing error of 0 have a higher weight. This way paths that may have a smaller cost but move the robot in the opposite direction of the goal will be discounted.\



|  |  |  |  |
| --- | --- | --- | --- |
| 3 | inf | 7 | 6 |
| 2 | inf | 6 | 5 |
| 1 | 2 | inf | 4 |
| 0 | 1 | 2 | 3 |

1. Path: 0 -> 1 -> 2 -> 3 -> 7 -> 11 -> 15
2. We spent around 8 hours on this lab