Dynamic Programming: Cost-to-Come

Peter Conant — 012155624

October 28, 2024

1 Requirements

My implementation of DP cost-to-come was developed in VSCode on a windows machine using Python (v3.12.4) and Numpy.

2 Operating the code

The script can be run using command `python 012155624.py`.

If you wish to change the start and end values on the graph from those retrieved from input.txt, you can do so by changing the `start` and `end` variables on lines 74-75.

3 Implementation

My implementation of cost-to-come is completed in a triple for loop with k representing the number of steps taken; i representing the node number we are searching for; j represents the neighbor. I use a n size vector to keep track the best value for each node and another n sized vector to keep track of the best neighbors. An n by n adjacency matrix hold the neighbors and the values of edge connecting the neighbors. My implementation does not keep track of neighbors and instead has to search through the row of the adjacency matrix to find edges that end with the current node. If I needed to reduce the time complexity of this function I would create a dictionary for each node where the first key would be the end node and the second key would be the start node and the value associated would be the cost of the edge.