

Lab 12

Exercise 1: Modify the code for the Bellman-Ford single-source shortest path algorithm so that it takes as input the origin x and destination y and returns the actual shortest *path* from x to y and not just the length of the shortest path. You can assume that y is reachable from x and that there are no negative weight cycles.

Exercise 2: Modify the code for Floyd-Warshall (either the recursive or iterative approach) to keep track of a matrix `next[][]` so that `next[u][v]` is some intermediate vertex on the shortest path from u to v (or the Python value `None` if there is no intermediate vertex). Now write a recursive procedure `findPath` which takes u , v , and the `dist` and `next` matrices, and returns the shortest path from u to v as a list. For example, if the shortest path from 0 to 3 is $0 \rightarrow 1 \rightarrow 3$, then `findPath` should return `[0,1,3]`.

Exercise 3: Modify the code for Floyd-Warshall to return -1 if there exists a negative weight cycle somewhere in the graph. Otherwise, it should return a matrix of distances as before.