CIS 477 (Fall 2022) Disclosure Sheet

Name: Kevin Lopez

HW # 6

Yes No Did you consult with anyone on parts of this assignment, including other students, TAs, or the instructor?

Yes No Did you consult an outside source (such as an Internet forum or a book other than the course textbook) on parts of this assignment?

If you answered Yes to one or more questions, please give the details here:

By submitting this sheet through my Blackboard account, I assert that the information on this sheet is true.

This disclosure sheet was based on one originally designed by Profs. Royer and Older.

Kevin Lopez

CIS477

11/24/22

HW6

1. Text

   Description automatically generated
   1. To do this algorithm, you start at node s and do a breadth first search to all adjacent nodes. By that notion, S would be connected to every possible node of U by a single edge. Yet this is incomplete because not every node in vertex set V is available in U, U is a part of V but not completely connected through V as it has less nodes.
2. Text

   Description automatically generated
   1. You can solve this problem with a dag pathfinding algorithm. You assign the nodes with the probabilities of slipping on them. The edges are only connected if they are M units of distance of each other and only face forward. You then multiply each of the paths that are connected and surpass R distance and find the lowest probability of slipping.
3. Text

   Description automatically generated
   1. C[i] is the cheapest way to get to shop i. You can do this by creating the subproblem of finding the next cheapest bus. At each station starting at station 0. If the cost of getting to station j and taking another bus is less than staying on the current bus, you take the next bus. You keep doing this for all n bus stations to the last station i.
   2. You can solve this with a DAG pathfinding algorithm. In this case the nodes are represented as the n bus stations and the edges are weighted with the cost Cij of getting to the next bus station. Starting at bus station 0, you find all the possible paths to get to the final bus station. You then choose the cheapest solution and find your answer.
4. A screenshot of a computer

   Description automatically generated with medium confidence
   1. The dag will be a longest pathfinding search algorithm however it will have a caveat. Instead of one value per node, it will have two. In this case, it will be the width and length. The edges will only point to the next node if the width and length are less than the node previous to it. Then you will find the longest series of edges. Another way of doing this is by making the edges be equal to the height and finding the largest sum of edges in the DAG.
   2. To do this algorithm, you find the maximum amount of ways to stack the boxes and choose the one with the largest height. You iterate and organize the boxes based on length and width in increasing order, but starting with the first element as the bottom box. Then you add up the heights based on the list and they fit the conditions. You continue iterating and finding the combinations based on each box. Then choose the largest height
5. Text

   Description automatically generated
   1. This is similar to the scheduling program, except you are looking for the most overlaps instead of scheduling them in separate groups. You can order the frequencies by most numbers overlapped. This also ensures the fewest number of laser probes because it combines the frequencies that can be tested together.