Stefan Theard PSID: 1208198 Project #1 COSC 3320 Problem#1 3rd due date

My advice on phase 2 was to recheck the problem definition and consider all the parameters.

The examples I believe I missed:

(a): The case when the distance from the last house back to Jonny’s house is not a part of the objective function.

(b): The case when the distance from the last house to Jonny’s house is a part of the objective function.

Further I need to prove my algorithms correctness and derive its time complexity. I’ll reiterate the algorithm here so it will be more clear since it is typed instead of written, I’ll prove it, analyze it, and then finally go over (a) and (b).

Shortest Cycle Algorithm:

Graph graph;

VertexNode current;

Path path; // global variable that stores the path to compare and be replaced by a //shorter path should one exist

shortestCycle()

{

While(!isComplete(path))

{

If(hasUnvisitedNeighbor(current, path))

{

VertexNode next = selectUnvisited(current, path);

generateRoadTo(next, current, path) //updates path w/ new edge and node

current = next; //move to next house

}

Else

{

VertexNode next = selectRandomNeighbor(current, path);

removeRoadTo(next, path)

/\* This is necessary because if the path is incomplete and does not have an unvisited neighbor, we’ve made a cycle that is complete but does not include every node, or house in this example which is invalid\*/

generateRoadTo(next, current, path);

path = findLastNode(next, current, path);

/\*findLastNode() will trace the stored path starting from next without passing through current and store that result into path if the new path is a shorter distance than the current path\*/

}

}

}