Greg Daniels
A00789340
CS5600 Homework 3

Problem 1:
(a)
When the cost to every node is equal, best first search will perform the same amount of work as the breadth first search. Because all of the costs will be equal.
(b)
In best first search the next node expanded is always the one chose to be the best based on some heuristic, with each expanded node being placed in a priority queue based on that heuristic function. If that function is simply the number of nodes to the top of the tree, then it will be the same as depth first search.

(c)

Problem 2:
(a) The relaxed version of tsp allows us to visit every vertex more than once (not create a hamiltonian cycle) a minimum spanning tree connects all remaining nodes and a tour can be constructed from the tree by following it around in a circular manner and returning to the beginning.

(b)
If nodes A, B , And C are all in a straight line. the minimum spanning tree will be exactly half of the optimal tour. this is because every edges is used in the minimum spanning tree and you must return to the starting node

(c) This question is vague. It sounds like a programming question, but this is written homework.

```
The way I would go about doing this
loop until I have generated the desired number of cities
    add a city to my graph with a random x and y coordinate between -1 and 1.
    return the graph
```

(d) Use prims or kruskals algorithm to generate a mst of the remaining nodes. As you are traversing the graph, at every step check to see if the cost so far plus the cost of the minimum spanning tree is greater than the best so far. If so you can terminate on that branch because the mst is a lowerbound on the rest of the tour.

Problem 3:
A depth first search would be better. If one is physically traversing the maze then depth first search would be better because it would involve much less walking than would a breadth first search. breadth first search requires the use of a queue. so at each intersection you come upon you would remember which options are available and then go back to the next on on the list. This would result in significantly more walking than if you used a depth first search.

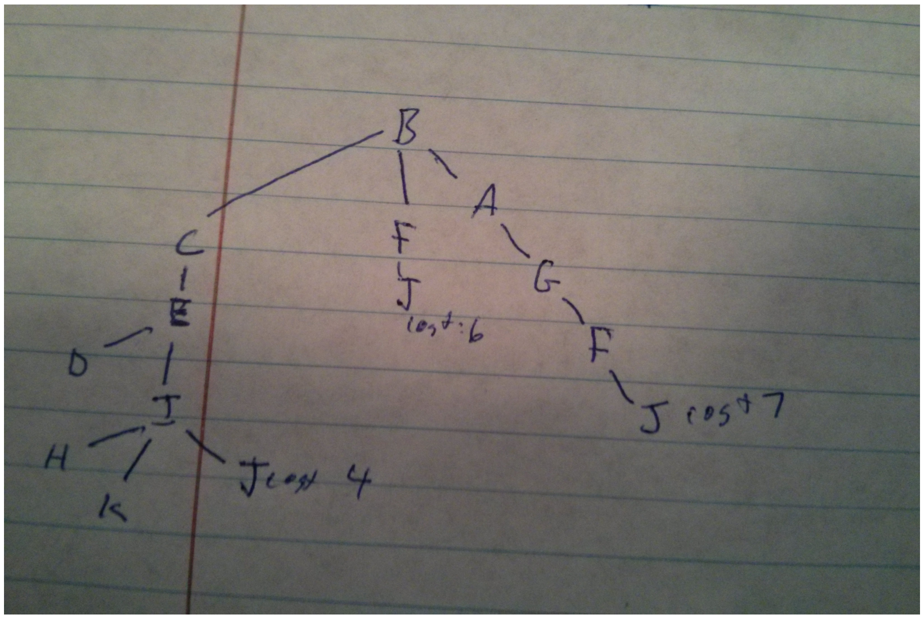
Problem 4:

(a) abcedihjfgk

(b) abgcfejdihk

(c) First this question is extremely vague. What is meant by solution? Cost or the whole path that it traverses? If we are going cost then the best and worst cases are 4 because a simple depth first search will always return the optimal path. If we are talking terms of computation before finding a solution, then we have 2 steps in the best case and in worst case 4.

(d) Same problems as above. What is meant by solution? If we are talking cost then best would be 4, because unless we are terminating when we find a solution, the best case and worst case are 4. but if we are talking computation before finding a solution(any solution) then best case is 5 nodes expanded and worst is also 5 nodes expanded. Look at the tree I drew. Rewrite these questions because they don't make any sense.


B
A
C
F
G
E
J
cost:6
F
D
J
J cost 7
H
J cost 4
K