Cos3751 assignment

Question 1

1.1 A single agent environment describes a type of environment that is observed and affected by one agent and thus a multi agent environment is one that is observed and affected by multiple agents. Observability being a factor that plays we say the agents must affect or be in an environment for them to count as agents.

1.2 A stochastic environment is random at best and chaotic at worst. A stochastic environment is described as an environment who's state changes independently of the current state and/or the agents actions. A deterministic environment's state is totally dependent on the current state and the actions of the agent.

1.3 The game of chess is fully observable as it allows an agent to perceive the current state, the rules of the game and even measure the solution probabilities.

Question 2

2.1

- Initial State
- Actions
- Transition Model
- Goal Test
- Path Cost

2.2 Search space is the entire set of states that an agent can explore to find a solution in a problem. Goal space is the set of states that an agent tests to see if it has achieved its goal.

2.3 To take into account all the explored states in a problem.

2.4

- Queue – This type of queue is the normal queue that sets objects stored in it in the order they were added… first in first out
- Priority Queue – This queue type uses weighted objects to in order of importance. The ones with greater priority are push forward.
- Stack – This type of queue sets items based on the way they were added but as a stack, meaning the first one in will be the last one out

2.5

- Completeness – this measures the ability of an algorithm to find a solution if one exists.
- Optimality – this measures if the algorithm find the most optimal solution to a problem
- Time Complexity – The amount of time it takes for the algorithm to find a solution if one exists.
- Space Complexity – The memory used by the algorithm to solve said problem

Question 3

3.1 State <- Position(Person, Side_ofRiver, action)
It is appropriate because it describes the position of the person relative to the side of the river, allows for a goal test to be applied and allows for actions to change state
Example : state <- Positon(Ché, Left)

3.2 Start : State <- Positon(Andile, Left, Stand(LeftSideofRiver))
Goal : State <- Position(Andile, Right, InBoat())

3.3 Ok

3.4 **Function** River-Cross-Search(problem) returns a solution, or failure

The functions would take the amount of people on one side the, the weight necessary to sink the boat and search a solution of how the people would get from one side of the river to the other side.

i.e it would for the first state take both the boys to the other side then bring back one boy and then one large adult would take the boat to the other side and so on. Searching for the optimal amount of trips to take for the right amount of people to cross the river and reach the goal scenario.

Question 4

4.1 A tree search looks at all possible paths to find a solutions while a graph search algorithm removes redundant paths to find a solution.
4.2 For starters a breadth first search is an instance of a general graph-search algorithm which means it searches the entire successor list of a root node before exploring others. We use this algorithm to speed up the process of finding a solution and removing redundant paths in our search.
4.3 There are about 1.3 trillion = 16! / 2 distinct states for that type of puzzle and about 16^d search nodes. This is a lot

Question 5

5.1 Ok
- Limit = 0: O
- Limit = 1: OME
- Limit = 2: OMDCELB
- Limit = 3: OMDKICJHELF

And then it stops at the goal on limit 3

Question 6

6.1 The uniform cost search uses a priority list that takes the lowest cost path as the priority. It adds up the total cost for a path and uses the lowest value to find a solution. The uniform cost search is like a driver calculating the shortest path to a destination and using that.
6.2 SO
- C -> F = 4
- C –> F -> H = 13
- C -> F -> H -> G = 25
- C -> A = 20
- C -> A -> B = 31
- C -> E = 24
- C -> E -> D = 46
- C -> F -> H -> G = 25

Then the algorithm stops and uses the C -> F -> H -> G = 25

They are expanded CFHABEDG

Question 7

7.1 An consistent heuristic measures the ability of a heuristic to stay consistent across the whole scope of it generating successor nodes such that the estimated cost of reaching a goal from its successor is greater than or equal to the consistency function over the root node and its successors on actions plus the heuristic of the successor. An admissible heuristic is that never overestimates the cost to reach the goal.

Question 8

8.1 A = 10 -> The estimated shortest path heuristic.

Visited = { A , D, K }

The first three nodes to be expanded will always be the ones from the initial state and they yield these values:
A -> B = 7 + 12 => 21
A -> C = 8 + 4 => 12
A -> D = 3 + 4 => 8

As the algorithm dictates the nodes that will be added to the frontier by order of importance \ will be {B, C, D} We expand D first and it yields:
A -> D -> K = 3 + 4 + 2 => 9
A -> D -> L = 3 + 6 + 1 => 10
A -> D -> I = 3 + 7 + 4 => 14

We then add D to the visited list and remove it from the frontier and then by order of importance K, L and M are added showing a new frontier {B, I, C, L}, K. We then choose to expand K and it yields:
A -> D -> K -> M = 3 + 4 + 2 + 1 = 10
A -> D -> K -> E = 3 + 4 + 5 + 4 = 16

After expanding all the nodes from K we add it to the visited list and add 2 new nodes to our list making
it {B, E, I, C, M, L}. We chose to expand M before L because that path took less nodes to have a 10 path
cost so it take priority. It yields:
A -> D -> L -> N = 3 + 6 + 3 + 0 = 12

it is added to the frontier and we see it doesnt take priority over the path cost for M so it takes a new position making the list {B, E, I, C, L, M}. The new path has the cost:
A -> D -> K -> M -> N 3 + 4 + 2 + 1 + 0 = 10

The algorithm ends here as this is the goal state of the algorithm .

8.2 No it isn't enough, depending on the type of search you need to be adding nodes that will lead to a solution i.e nodes that will be expanded opportunistically for a better state.

8.3 {B, E, I, C, L, M}

8.4 It inconsistent, the graph takes up many values but it should never be greater than the value of the estimate heuristic every node value and their successor should essentially agree with the h(n) <= c(n, a, n') + h(n) inequality and from the beginning we have A -> D = 8 i.e 10 <= 8 is not true for this state and thus defies the law making the heuristic applied inconsistent.

Question 9

9.1 A global maximum is the highest value in the entire function and the local maximum is the highest value in a small location within the function it will always be smaller than the global maximum. The global minimum is the lowest value in a function and the local minimum is the smallest valley within a neighbourhood within a function but is isn’t lower than than the global minima.

9.2 Function Simulated-Annealing(problem) returns a solution or failure.
Takes a problem and randomly populates the problem space and then swaps values till a solution is found.

9.3 A hill-cliimbing search is a function that follows the direction of increasing value without thinking about anything else. A simulated annealing search is described as the hill-climbing approach mixed with random walk. This means you take the ability to follow increasing value but removing the weakness of being stuck at the top because it cannot go to decreasing values by giving it the ability to pick random move that put it in a better situation. Yes it would have been a good fit as it is exactly how simulated annealing would deal with the situation… apply random figures and switch between then till one that makes sense is found and accepted. Classic simulated annealing.

9.4