# Theoretical Problems:

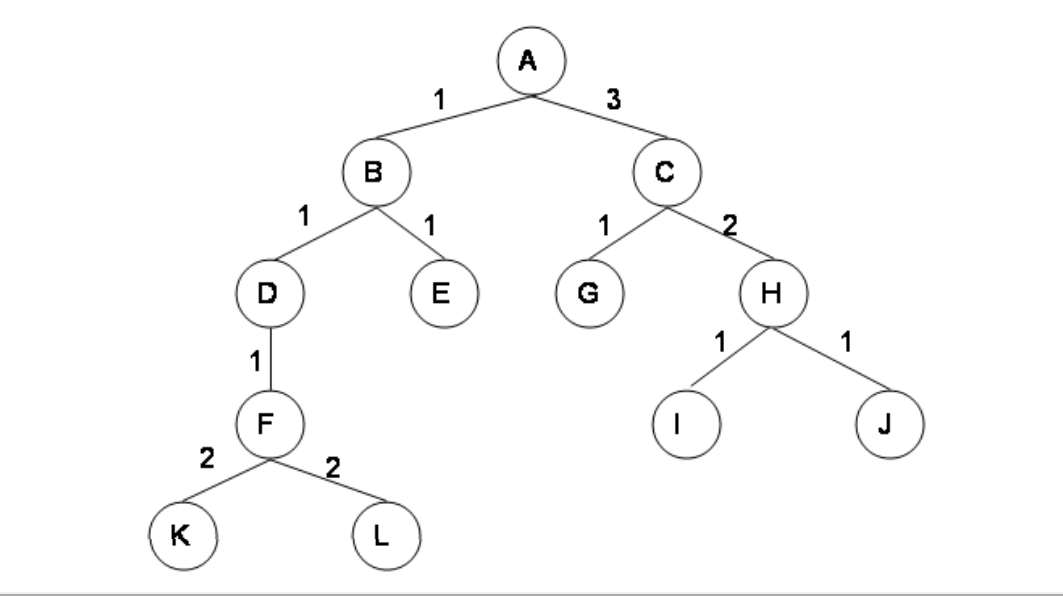
**Submission Deadline - 11:55PM, 9th August, 2017.**

**Name**:

**Roll**:

**You need to submit a .pdf file containing your solutions for the following problems. Mention your name and Roll No. on the top of the document. The file name should be <Roll No>\_assignment1\_ th.pdf. For eg. 13CS30024\_assignment1\_th.pdf. You are free to use any editor of your choice. Resize the tables according to your needs.**

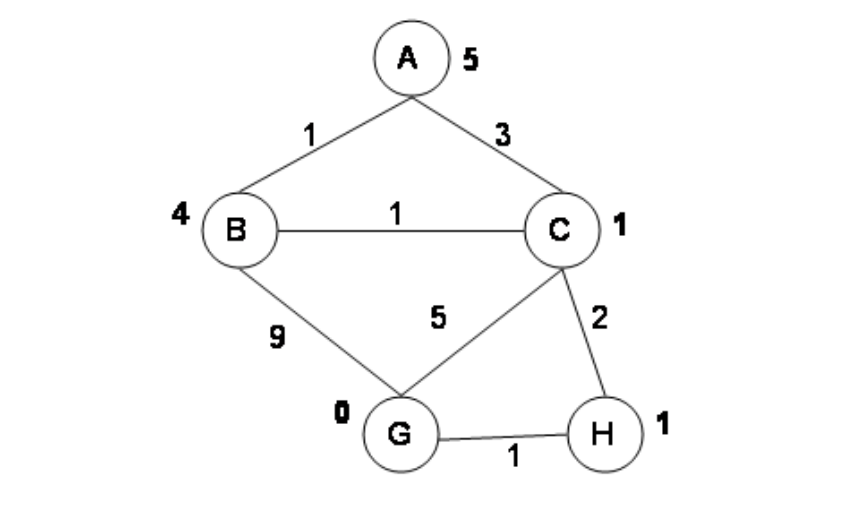
1. Describe in brief an autonomous vehicle as an intelligent agent. Give a **PEAS** specification for the task environment.
2. Prove the following:
   1. A\* search is complete
   2. A\* search is optimal.
3. Weighted A\* can be described as best-first search with: f(n) = g(n) + w · h(n) or as f(n) = (1-w) · g(n) + w · h(n) . Using one of these formulations, provide a bound on the maximum sub-optimality of a path that weighted A\* can return as a function of w.
4. Consider the tree shown below. The numbers on the arcs are the arc lengths



Assume that the nodes are expanded in alphabetical order when no other order is specified by the search, and that the goal is state G. No visited or expanded lists are used. What order would the states be expanded by each type of search? Stop when you expand G. FIll the boxes with sequence of states expanded by each search.

|  |  |
| --- | --- |
| **Search Type** | **List of states** |
| Breadth First |  |
| Depth First |  |
| Iterative Deepening Search |  |
| Uniform Cost Search |  |

5. Consider the graph shown below where the numbers on the links are link costs and the numbers next to the states are heuristic estimates. Note that the arcs are undirected. Let A be the start state and G be the goal state.



Simulate A\* search with a strict expanded list on this graph. At each step, show the path to the state of the node that’s being expanded, the length of that path, the total estimated cost of the path (actual + heuristic), and the current value of the expanded list (as a list of states). Please transcribe (only) the information requested into the table given below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Path to State** | **Expanded Length of Path** | **Total Estimated Cost** | **Expanded List** |
| A | 0 | 5 | (A) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

6. Answer the following questions with respect to Problem 5.

1. Is the heuristic given in Problem 5 admissible? Explain.
2. Is the heuristic given in Problem 5 consistent? Explain.
3. Did the A\* algorithm with strict expanded list find the optimal path in the previous example? If it did find the optimal path, explain why you would expect that. If it didn’t find the optimal path, explain why you would expect that and give a simple (specific) change of state values of the heuristic that would be sufficient to get the correct behavior.

7. The water jug problem can be stated as follows: you are given two jugs of capacities 4 litres and 3 litres. You also have a pump that can be used to fill either jug with water, and you can empty the contents of either jug at any time. Your goal is to get exactly 2 litres of water in the 4 litre jug.

1. Formulate this problem as a state-space search describing the state, move-generator and the goal checker.
2. Suppose that it costs Rs. 5 every time the pump is used, Rs. 2 every time you fill the 4 litre jug and Rs 1 every time you fill the 3 litre jug. Find the lowest cost solution to this problem.

8. Decide whether the following statements are true or false. If true, explain why. If false, give a contradicting example. Recall that B is the average branching factor and L is the length of the shortest path from start to goal.

1. Bi-directional BFS is always faster than BFS when B ≥ 2 and L ≥ 4.
2. A\* search always expands fewer nodes than DFS does.
3. For any search space, there is always an admissible and consistent A\* heuristic.
4. IDA\* does not need a priority queue as in A\*, but can use the program stack in a recursive implementation as in DFS.