Assignment 5

The due date for submitting this assignment has passed. As per our records you have not submitted this assignment.

BEGIN GROUP

1) For all infinite graphs with finite branching factor and arbitrary positive non-zero edge costs, which of the following algorithms will always find a path to the goal if one exists?

- A*
- Best First Search
- Branch and Bound Search
- Breadth First Search
- Depth First Search
- Dijkstra's Algorithm that is able to access the graph on a need-basis
- None of the above

No, the answer is incorrect.
Score: 0
Accepted Answers:
Breadth First Search

2) For all infinite graphs with finite branching factor and with edge costs greater than a small positive non-zero constant \( \varepsilon \), which of the following algorithms will always find a path to the goal if one exists?

- A*
- Best First Search
- Branch and Bound Search
- Breadth First Search
- Depth First Search
- Dijkstra's Algorithm that is able to access the graph on a need-basis
- None of the above

No, the answer is incorrect.
Score: 0
Accepted Answers:
A*  
Branch and Bound Search  
Breadth First Search  
Dijkstra's Algorithm that is able to access the graph on a need-basis

3) For all graphs described in the previous question, which of the following algorithms will always find an optimal path if the start and goal nodes live in the same connected component?

- A*
- Best First Search
- Branch and Bound Search
- Breadth First Search
- Depth First Search
- Dijkstra's Algorithm that is able to access the graph on a need-basis
- None of the above

No, the answer is incorrect.
Score: 0
Accepted Answers:
Branch and Bound Search  
Dijkstra's Algorithm that is able to access the graph on a need-basis
4) The figure shows an infinite graph with finite branching factor and \textit{arbitrary positive non-zero} edge costs. For this graph, which of the following \textbf{1 point} algorithms will find an optimal path from S to G?

The MoveGen returns nodes in alphabetical order.
The heuristic function \( h(N) \) is defined as: \( h(An) = \frac{1}{2^n} \) and \( h(\text{other nodes}) = 1 \).
Use alphabetical order to break ties

- A*
- Best First Search
- Branch and Bound Search
- Breadth First Search
- Depth First Search
- Dijkstra's Algorithm that is able to access the graph on a need-basis
- None of the above. Because of the MoveGen these algorithms will refine S, A1, A2, A3, A4, A5 and so on and will never return from the infinite branch.

No, the answer is incorrect.
Score: 0
Accepted Answers:
A*
Best First Search
Branch and Bound Search
Breadth First Search
Dijkstra's Algorithm that is able to access the graph on a need-basis

5) Consider the following values that can be associated with a node N. Let S be the start node and G be the goal. \textbf{1 point}

- A: number of hops from S to N
- B: \( 1 - A \)
- C: actual path cost from S to N
- D: estimated path cost from S to N
- E: actual path cost from N to G
- F: estimated path cost from N to G
- G: actual cost from N to its nearest neighbour
- H: number of hops from N to G
- I: \( C + E \)
- J: \( C + F \)
- K: \( D + E \)
- L: \( D + F \)
- M: NONE (none of the above values are used)

Given that each search algorithm refines the candidate with the \textit{lowest value} associated with each candidate, identify the value used by each of the following algorithms.

- Depth First Search uses M
- Depth First Search uses B
- Breadth First Search uses A
- Breadth First Search uses M
- Best First Search uses D
- Best First Search uses F
- Branch and Bound uses C
- Branch and Bound uses D
- Algorithm A* uses J
- Algorithm A* uses L
- Dijkstra's Algorithm uses J
- Dijkstra's Algorithm uses C

No, the answer is incorrect.
Score: 0
Accepted Answers:
Depth First Search uses B
Breadth First Search uses A
Best First Search uses F
Branch and Bound uses C
Algorithm A* uses J
Dijkstra's Algorithm uses C
The figure shows a map with several locations on a grid where each tile is 10km x 10 km in size. In this map, S is the start node and G is the goal node, the locations are connected by two-way edges (roads). Each edge has a cost and the cost is the same in both directions.

For this map, MoveGen returns nodes in alphabetical order. When several nodes have the same cost, use alphabetical order to break ties.

When the same node occurs multiple times with the same cost then use the open-sequence-number to break ties, i.e., refine the node occurrence that is oldest (was placed earliest in time) in the OPEN list.

Where needed, use Manhattan distance as the heuristic function.

BnB-1: is the simple Branch And Bound algorithm covered in the lecture, which allows cyclic refinements/paths like SASA...SA in the search-tree.

BnB-2: is a variation on BnB-1, that prevents cyclic expansions, i.e., it does not refine a node that already occurs in the path (and partial path) from the root to the current node. Therefore, it keeps each path in the search tree (and each partial path) free of duplicate nodes.

6) The search tree after 15 node-refinements by BnB-1 is shown below. Run BnB-1 and populate this search tree with node labels, node costs, the order in which nodes were added to OPEN list, the order in which nodes are refined, etc.

Let S be the first node to be refined. Now determine the 5 nodes from the 6th node to the 10th node that are refined by BnB-1. List the node labels as a comma separated list and enter the nodes in the order it was refined. DO NOT enter extraneous characters like spaces, dots, brackets, extra commas, etc.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) S,P,S,S,S
(Type: String) S, P, S, S, S

7) For the 5 nodes listed in the previous question, list the costs of those nodes as a comma separated list of 5 natural numbers. Use the same node order used in the previous question. DO NOT enter extraneous characters like spaces, dots, brackets, extra commas, etc.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) 50,70,70,70,70
(Type: String) 50, 70, 70, 70, 70

8) In the search tree shown for BnB-1, the number of times node S appears (including refined and unrefined cases) is _______.

No, the answer is incorrect.
Score: 0
9) In the search tree shown for BnB-1, the number of unrefined nodes is _______.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: Numeric) 7

10) The search tree after 15 node-refinements by BnB-2 is shown below. Run BnB-2 and populate this search tree with node labels, node costs, the order in which nodes were added to OPEN list, the order in which nodes are refined, etc.

Let S be the first node to be refined. Now determine the 5 nodes from the 6th node to the 10th node that are refined by BnB-2. List the node labels as a comma separated list and enter the nodes in the order it was refined. DO NOT enter extraneous characters like spaces, dots, brackets, extra commas, etc.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) P, U1, U1, V1, W

11) For the 5 nodes listed in the previous question, list the costs of those nodes as a comma separated list of 5 natural numbers. Use the same node order used in the previous question. DO NOT enter extraneous characters like spaces, dots, brackets, extra commas, etc.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) 70, 70, 70, 70, 85

12) In the search tree shown for BnB-2, the number of times node S appears (including refined and unrefined cases) is _______.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: Numeric) 1

13) In the search tree shown for BnB-2, the number of unrefined nodes is _______.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: Numeric) 18

14) In the map, let S be the first node to be refined. Now determine the 5 nodes from the 6th node to the 10th node that are refined by A*. List the node labels as a comma separated list and enter the nodes in the order it was refined. DO NOT enter extraneous characters like spaces, dots, brackets, extra commas, etc.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) T, U, V1, M, U1
15) For the 5 nodes listed in the previous question, list the costs of those nodes as a comma separated list of 5 natural numbers. Use the same node order used in the previous question. DO NOT enter extraneous characters like spaces, dots, brackets, extra commas, etc.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) 245,245,250,270,270

1 point

16) For the map, what is the path found by A* algorithm. List the path (starting from S and ending in G) as a comma separated list of node labels. DO NOT enter extraneous characters like spaces, dots, brackets, extra commas, etc.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) S,V,V1,Z1,Z,J,G
(Type: String) S, V, V1, Z1, Z, J, G

1 point

17) What is the cost of the path found by A* algorithm?
Enter the cost as a natural number. DO NOT enter dots or other extraneous characters.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: Numeric) 290

1 point

18) For the map, starting from S, run Best First Search to its completion. What is the path found by Best First Search algorithm? List the path (starting from S and ending in G) as a comma separated list of node labels. DO NOT enter extraneous characters like spaces, dots, brackets, extra commas, etc.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) S,O,Q,M,E,B,G
(Type: String) S, O, Q, M, E, B, G

1 point

19) What is the cost of the path found by Best First Search algorithm?
Enter the cost as a natural number. DO NOT enter dots or other extraneous characters.

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: Numeric) 490

1 point

20) In the map, if an agent using A* algorithm were to traverse from S to G, and then for its return journey from G to S, if it recomputes the path by treating G as the start node and S as the goal node, then the agent will ...

- take the same path back
- take a different path back
- get stuck on node I
- behave in arbitrary manner

No, the answer is incorrect.
Score: 0
Accepted Answers:
take the same path back

END GROUP