CZ3005: Artificial Intelligence

Lab 1

Submitted By:
Royce Ang Jia Jie
U1840416D
TSP2

Question One

Reference:

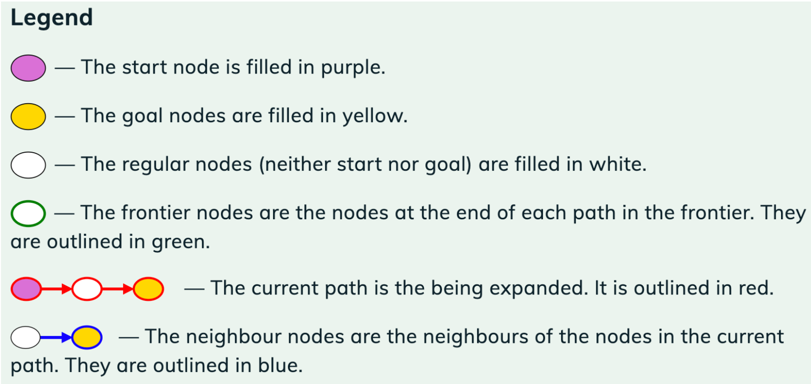


Figure 1 Legend from AI Space

1(a): A graph where DFS is more efficient than BFS

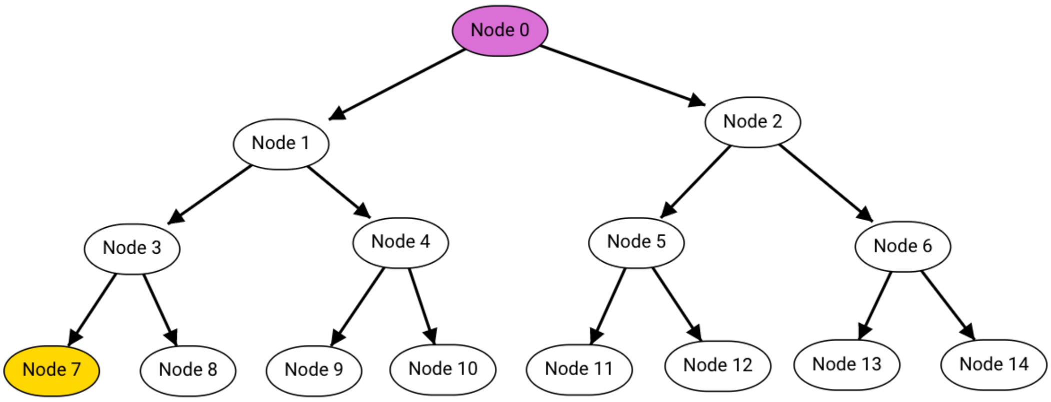
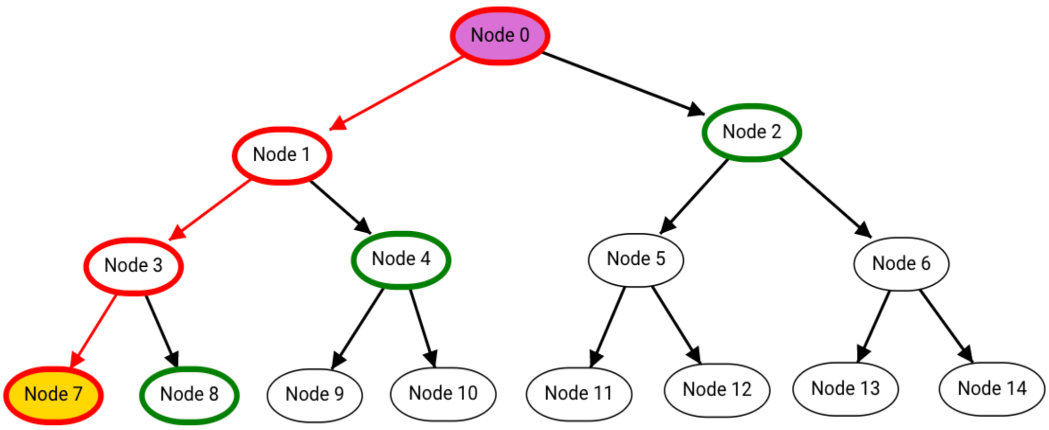
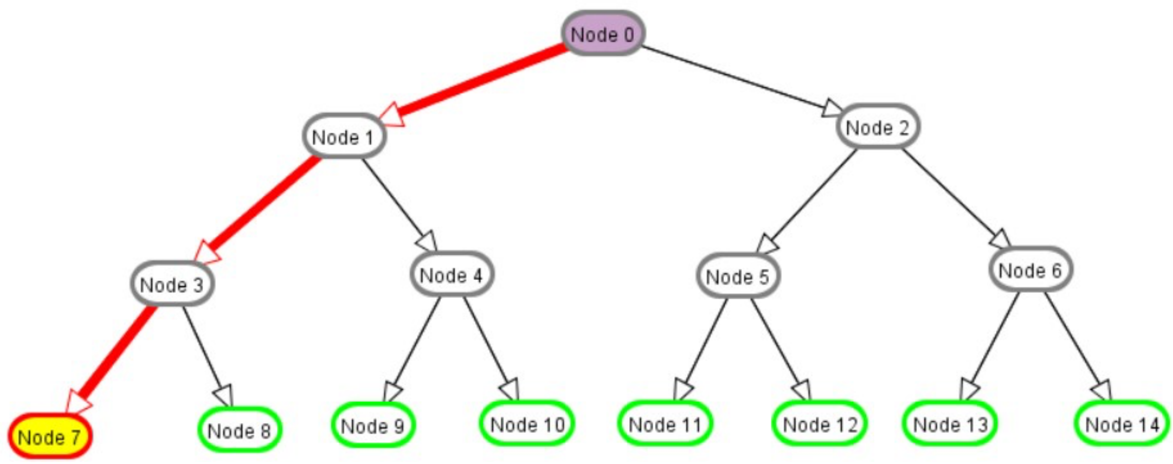


Figure 2 Reference Graph, Start Node = 0, Goal Node = 7

DFS



Explanation:
Path Taken: Node 0 (Start) → Node 1 → Node 3 → Node 7(Goal) Cost: 3.0 with equal weight across all traversing edges.
Total Nodes Expanded: 4

BFS

Explanation: Path Taken: Nodes 0 → Node 1 → Node 3 → Node 7 (Goal) Nodes expanded: 8

1(b): A graph where BFS is much better than DFS.

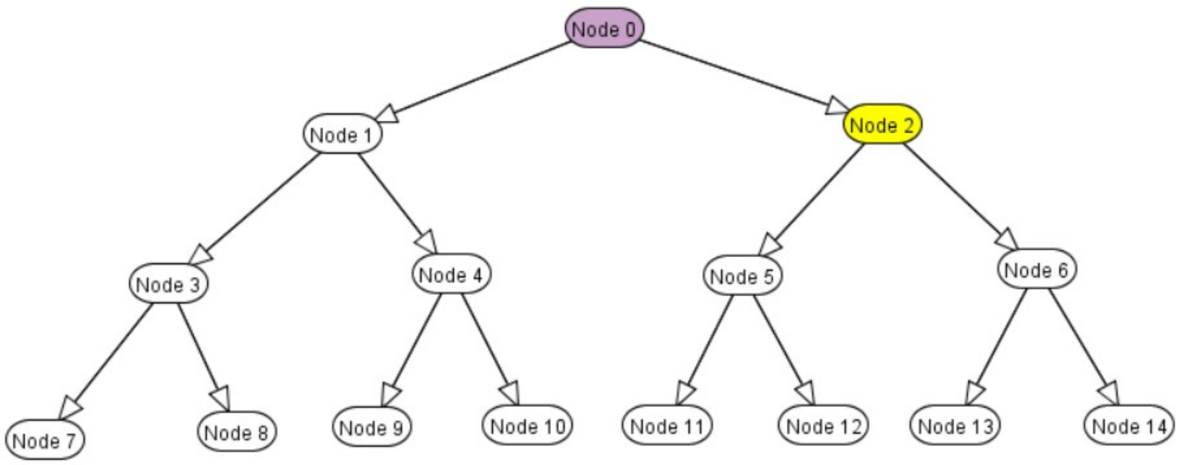
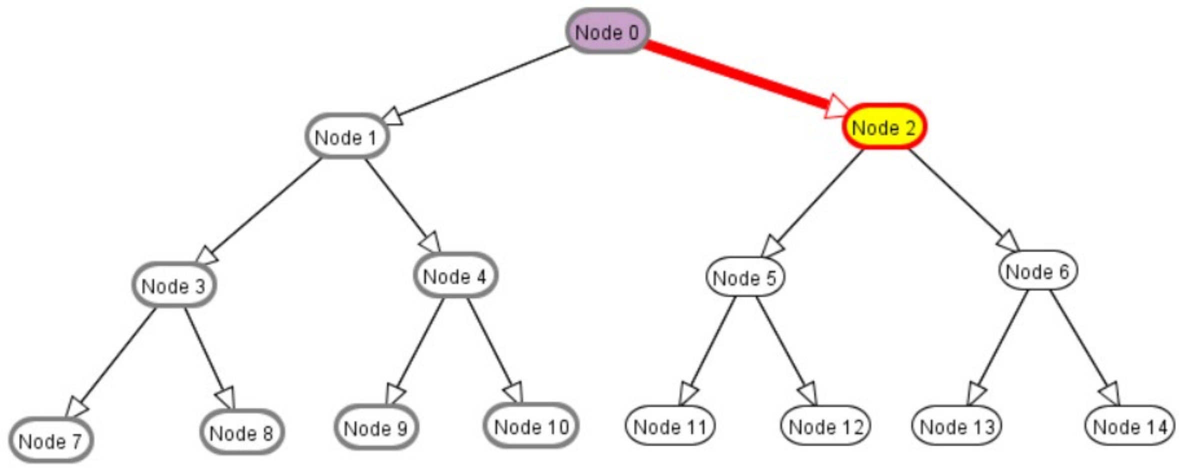
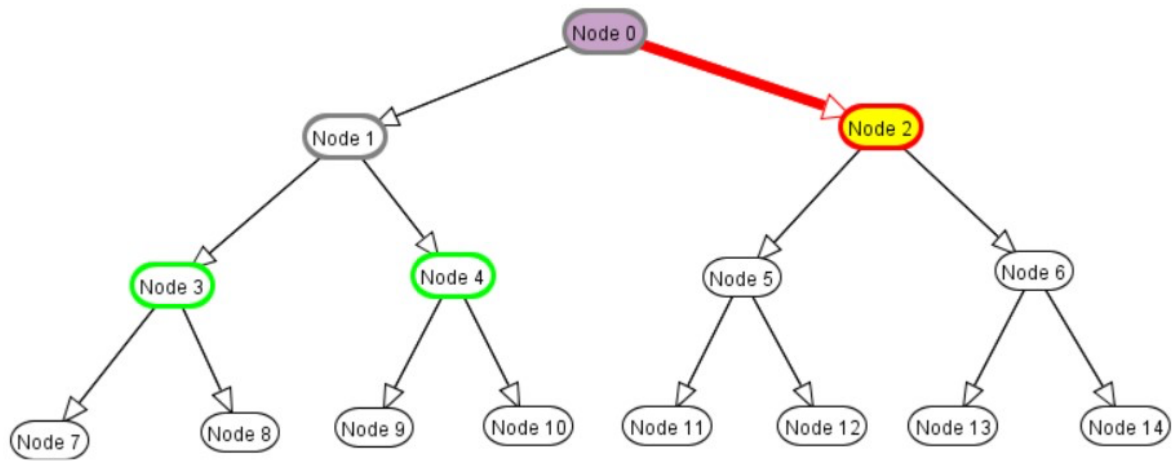


Figure 3 Reference graph for question 1b

DFS

Explanation: Path Taken: Node 0 → Node 2 (Goal) Nodes expanded: 9
BFS


Explanation: Path Taken: Node 0 → Node 2 (Goal) Nodes expanded: 3

1(c): A graph where A* Search is more efficient than DFS or BFS.
Given that heuristic, h(n), is the actual distance from any Node(N) to Goal Node.

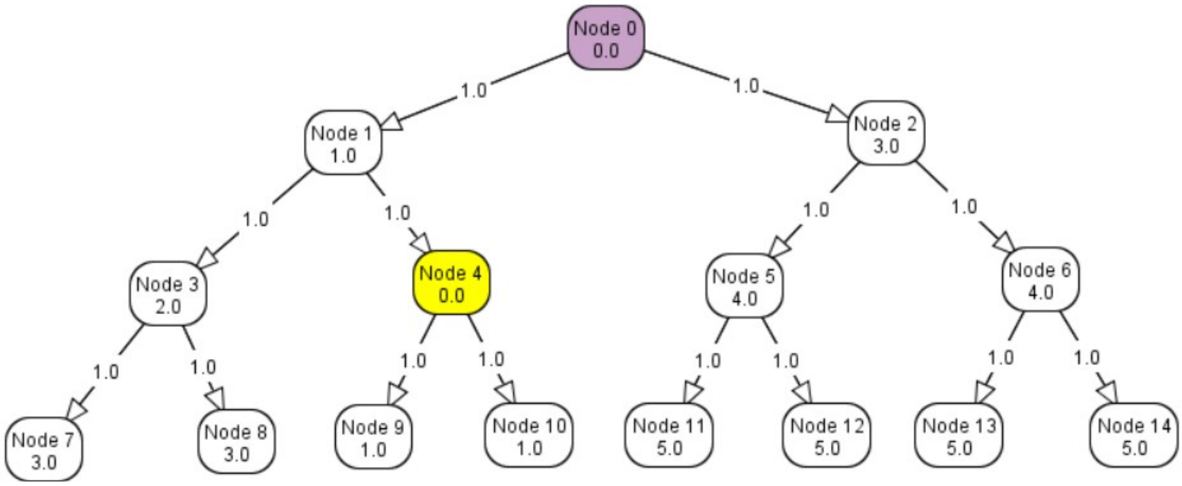
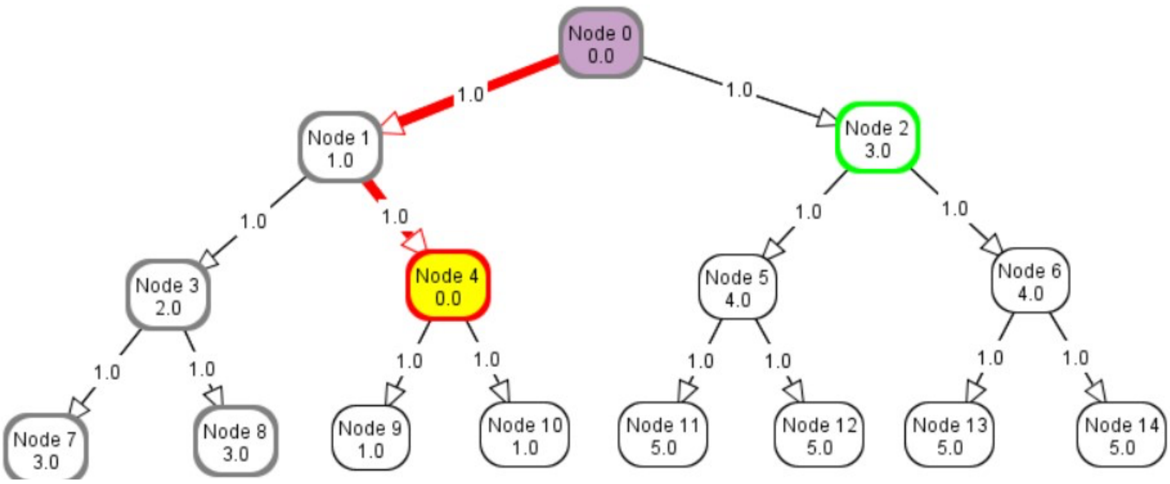
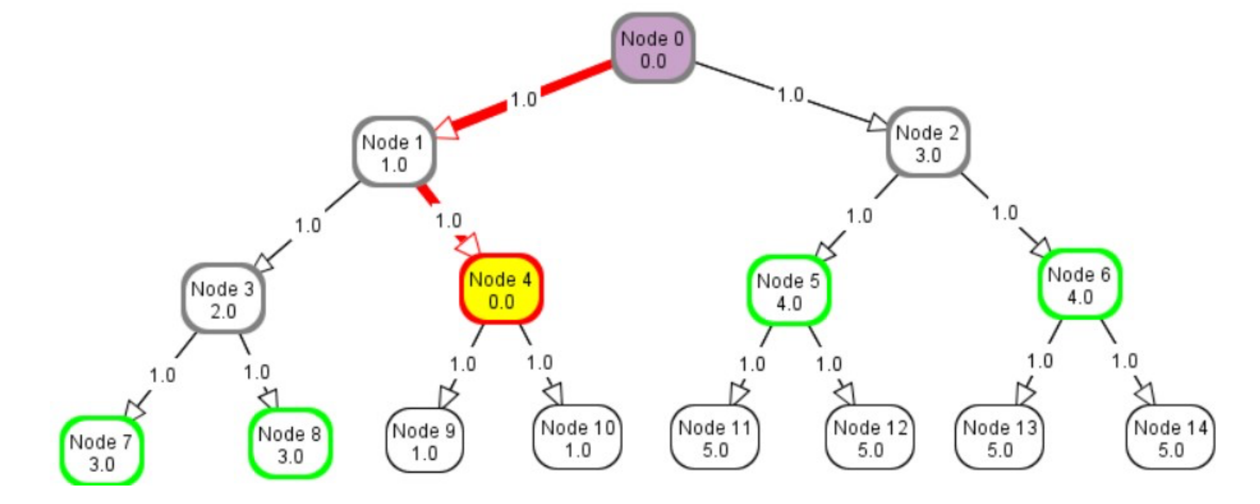


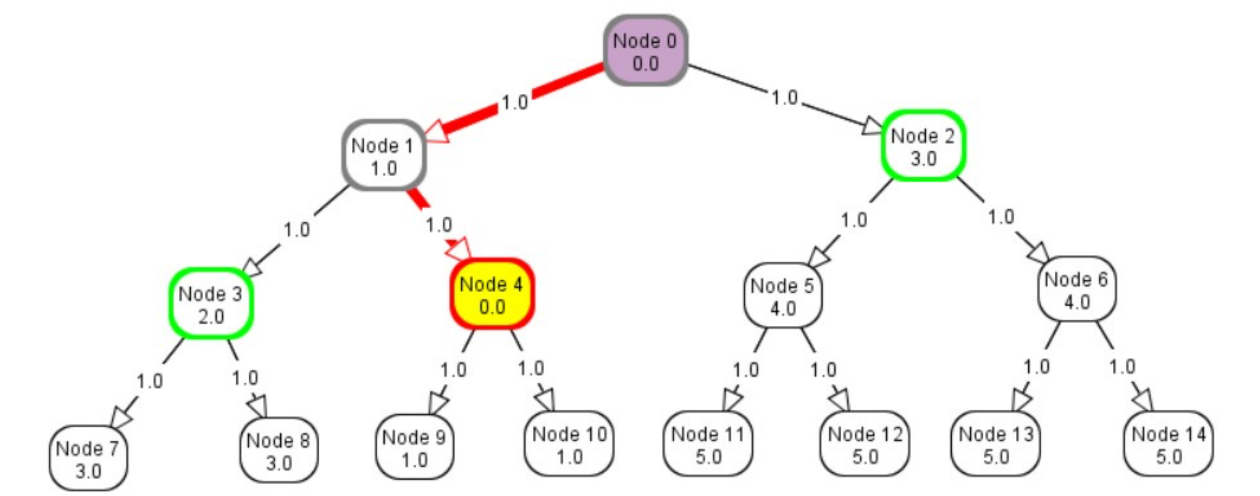
Figure 4 Reference graph for question 1c.

DFS

Explanation: Path Taken: Node 0 → Node 1 → Node 4 (Goal) Nodes expanded: 6
BFS



Explanation:
Path Taken: Node 0 → Node 1 → Node 4 (Goal)
Nodes expanded: 5

A*



Explanation:
Path taken: Node 0 → Node 1 → Node 4 (Goal)
Nodes expanded: 3

1(d): A graph where DFS and BFS are more efficient than A*

Given heuristic, h(n), is the absolute of the difference of (index of goal node – index of current node). Additionally, given that the goal node is positioned on the left-hand side of the tree, the heuristic will increase the bias towards the right-hand side of the tree than the left-hand side of the tree. Formula: h(n) = | index of goal node – index of current node |

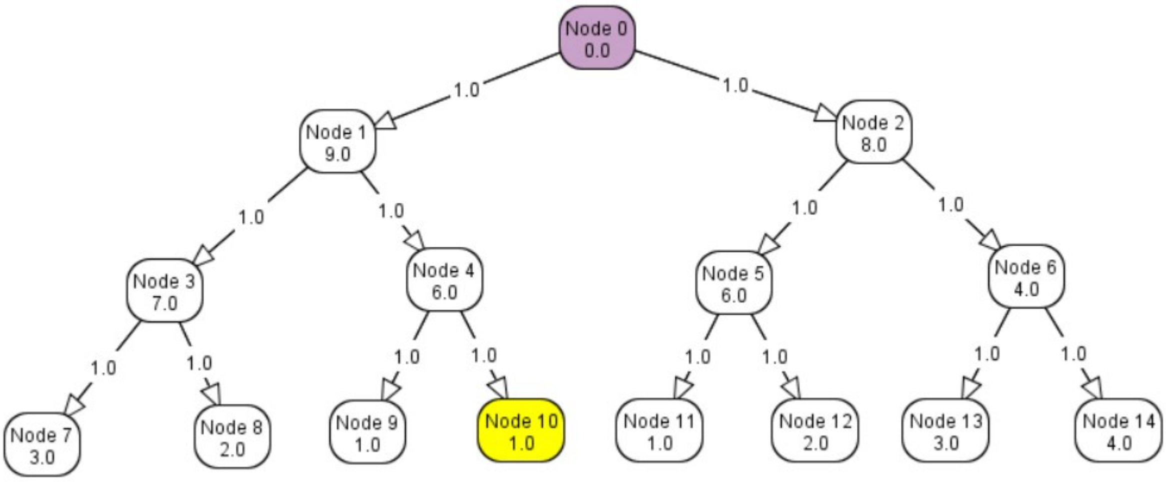
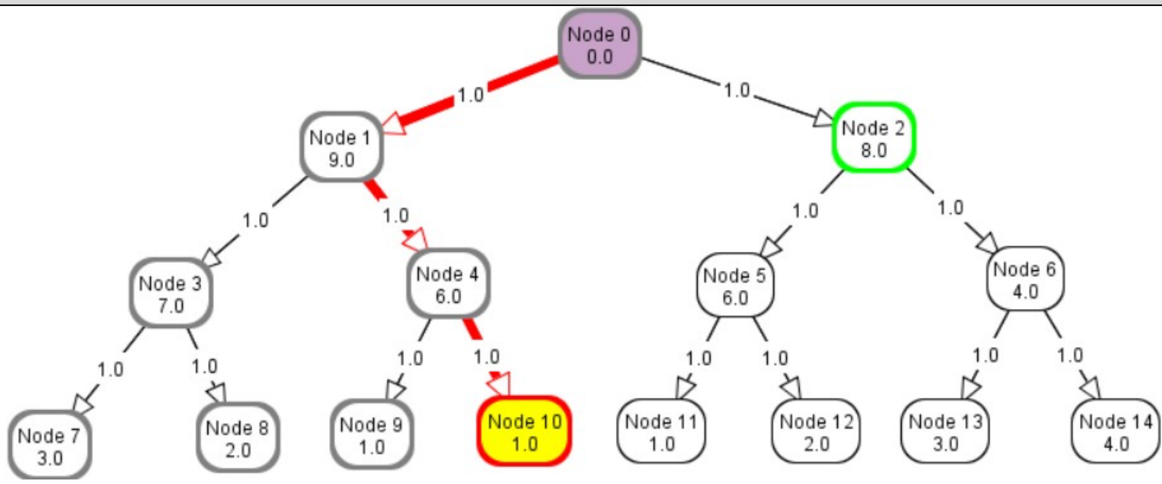
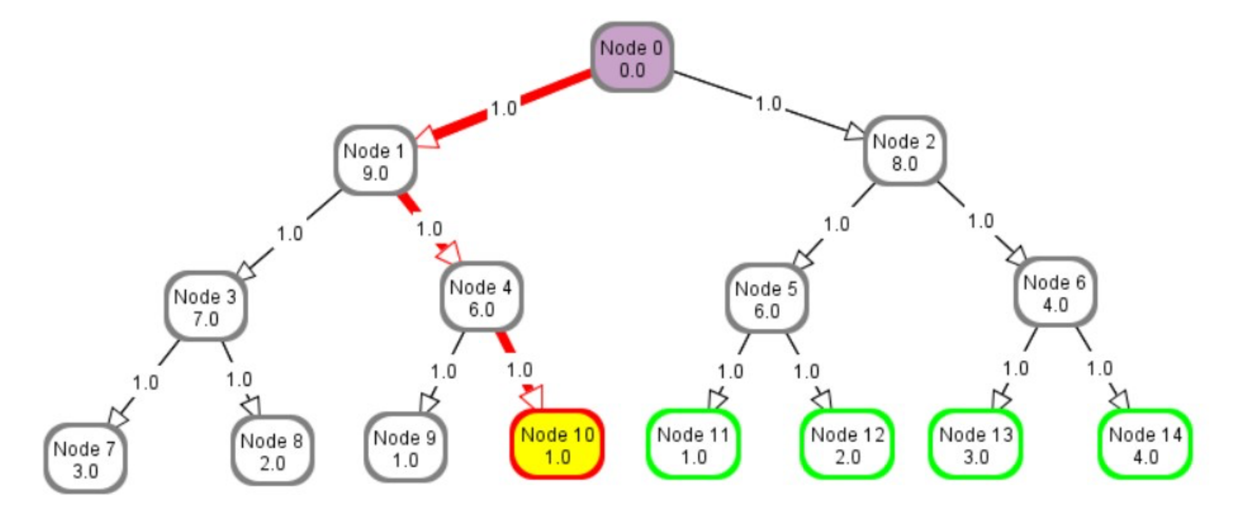


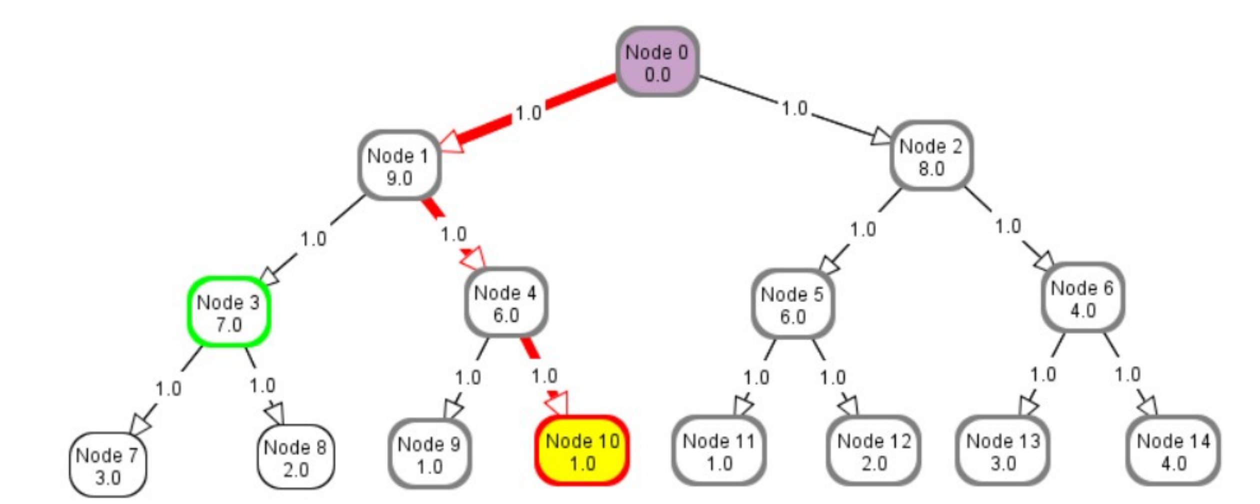
Figure 5 Reference graph for question 1d.

<table>
<tr><th>DFS</th></tr>
<tr><td></td></tr>
<tr><td>Explanation:
Path taken: Node 0 → Node 1 → Node 4 → Node 10 (Goal)
Nodes expanded: 8</td></tr>
<tr><th>BFS</th></tr>
</table>



Explanation:
Path taken: Node 0 → Node 1 → Node 4 → Node 10 (Goal)
Nodes expanded: 11

A*



Explanation:
Nodes expanded: Node 0 → Node 1 → Node 4 → Node 10 (Goal)
Nodes expanded: 12

Question Two

Conjecture: The closer h(n) is to goal node, G, the more efficient A* search will be, thereby promoting a more accurate heuristic.

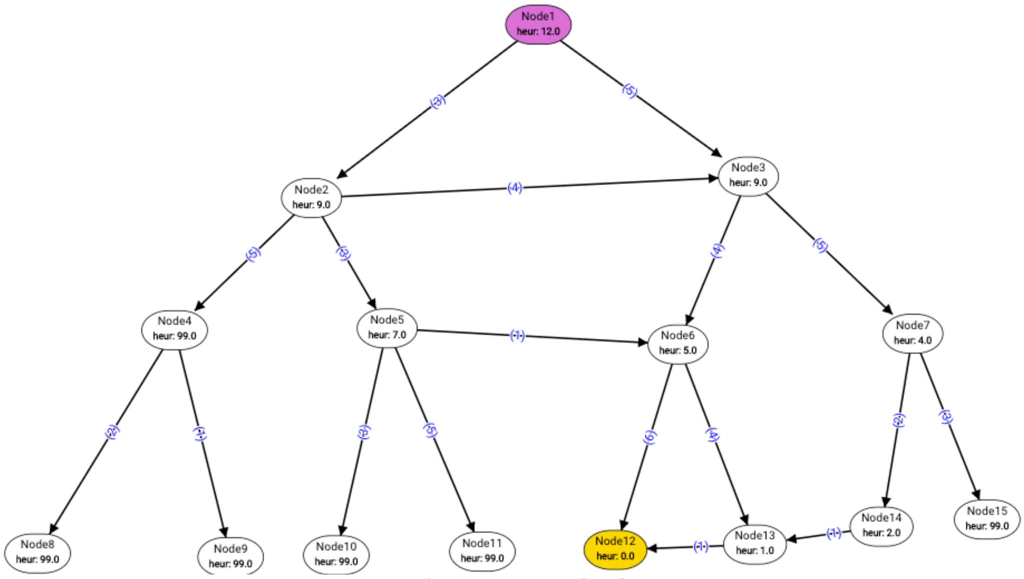


Figure 6 Reference graph for Question 2

Case 1: h(N) = D(N, G); for any given node N has heuristic of the same cost/distance to the goal.

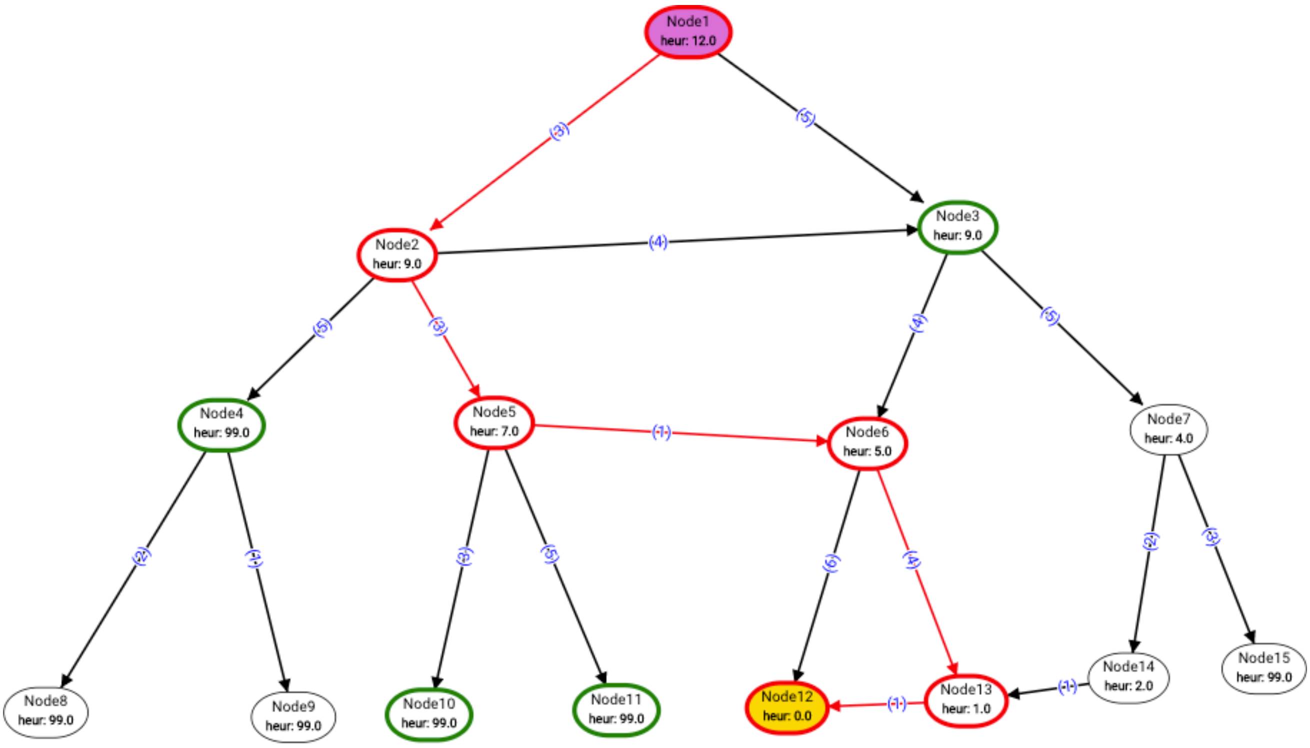


Figure 7 h(N) = D(N, G)

Node 1 (Start) → Node 2 → Node 5 → Node 6 → Node 13 → Node 12 (Goal) (Cost 12)
Total Nodes Expanded: 6 (1, 2, 5, 6, 12, 13)

Case 2: h(N) = 0.5*D(N, G); for any given node N has a heuristic that is half of the cost/distance to goal.

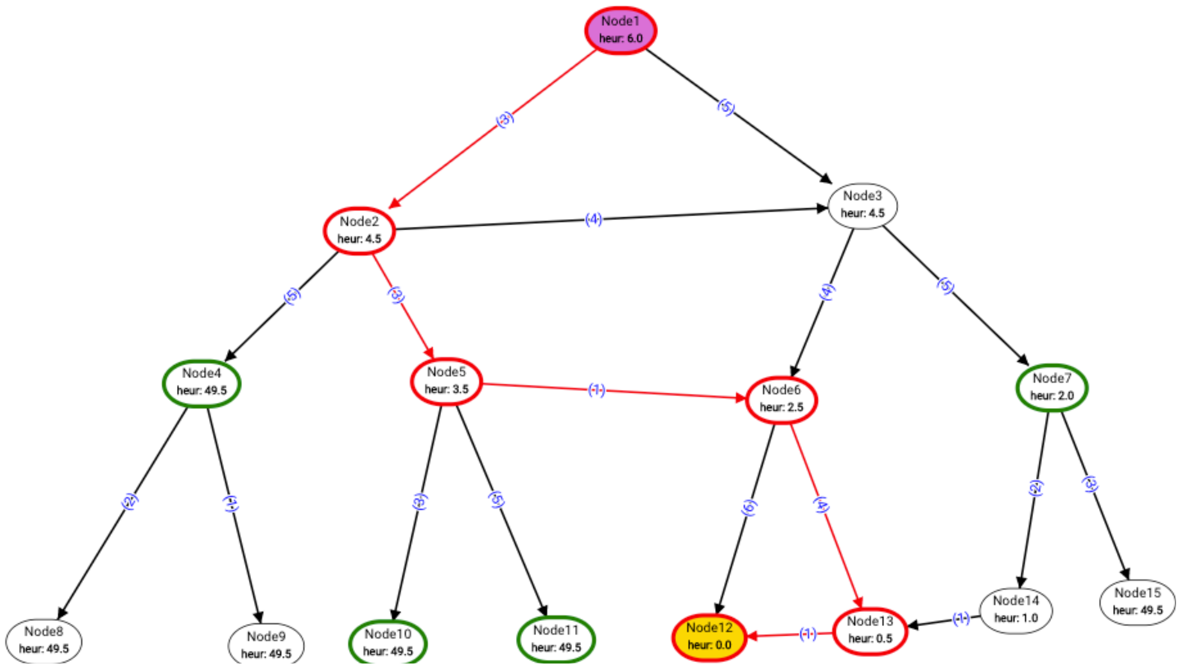


*Figure 8 h(N) = 0.5*D(N, G)*

Node 1(Start) → Node 2 → Node 5 → Node 6 → Node 13 → Node 12(Goal) (Cost 12)
Total Nodes Expanded: 7 (1, 2, 5, 6, 3, 13, 12)

Case 3: h(N) = 0.1*D(N,G); for any given node N has a heuristic that is a tenth of the cost/distance to goal.

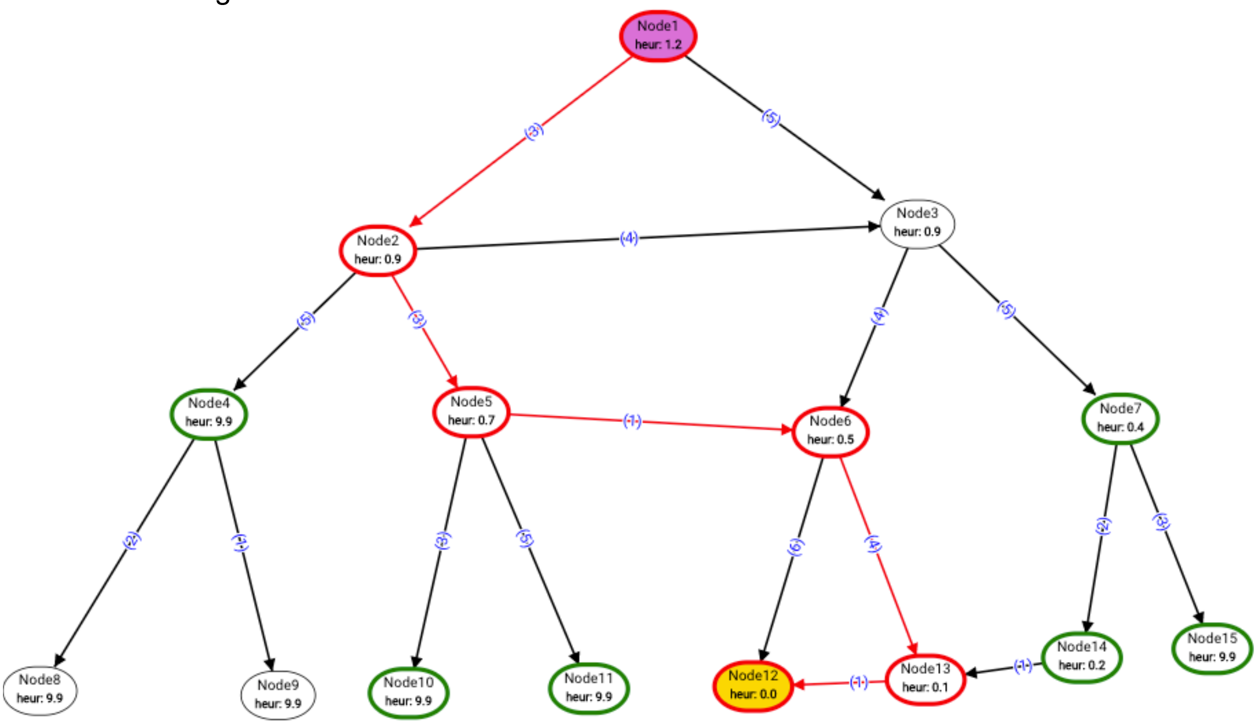


*Figure 9 h(N) = 0.1*D(N,G)*

Node 1(Start) → Node 2 → Node 5 → Node 6 → Node 13 → Node 12(Goal) (Cost 12)
Total nodes expansion: 8 (1, 2, 3, 5, 6, 7, 13, 12)

Summary	h(N) = D(N, G)	h(N) = 0.5*D(N, G)	h(N) = 0.1*D(N,G)
Total Expansion	6	7	8

Observation: We can see that as we scale heuristic, h(N), further away from the actual Goal, the less efficient the A* Search becomes. The total expansions increased from 6 to 8 nodes.

2(a)

- Reducing h(N) when h(N) is already an underestimate would make A* Search less efficient but equally optimal.

Prove:

- Assuming that we have a perfect heuristic that returns actual path cost to goal, h*(N), and we currently are using $h_2(N)$, but chose to further reduce $h_2(N)$ to $h_1(N)$.
- Then, $h_1(N) < h_2(N) \le h^*(N)$, where h_2 is a better heuristic than h_1.
- If A^*_1 uses $h_1(N)$ and A^*_2 uses $h_2(N)$, then every node expanded by A^*_2 will also be expanded by A^*_1.
    - We know that in general f(G) = h(G) + g(G), when h(G) is 0 at the goal node, f(G) = g(G)
    - Therefore, f1(G) = f2(G) = g(G).
    - And, *f1(N) = h1(N) + g(N) < f2(N) = h2(N) + g(N)* ≤ g(G).
    - To conclude, this means that every node expanded by A^*_2 will have path cost no greater than the actual cost to goal G. And every node expanded by A^*_2 will also be expanded by A^*_1 since $f_1(N)$ is a smaller cost compared to $f_2(N)$.

2(b)

- A* with h(N) that is of exact distance from n to a goal will always be efficient.
- Theoretically, the purpose of an underestimate is to "bias" the search from node, N, to goal, G.
- Therefore, having a h(N) = exact distance to Goal, would reflect the "true" distance to the goal. Thereby, reflecting the "true bias" from node, N to goal, G.
- To be more specific, assuming that we have a h(N) = cost(N,G). And that, cost(N,G) ≤ g(G).
- Then, for all f(N) = cost(N,G) + g(N), the search would always follow cost(N,G). Therefore, reflecting the true goal towards goal, G, from node, N.

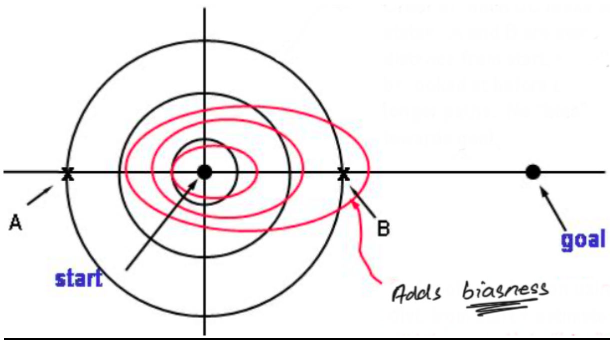


Figure 10 Underestimation, h(N), will add bias to Goal, G.

- Additionally, if there are multiple paths, as long as the frontier is scheduled in a first-in-last-out (FILO/LIFO) order for nodes with the same f(N), it is efficient.

2(c)

If all h(n) is not an underestimate, then h(n) is no longer admissible and therefore, A* Search's optimality and efficiency will not be guaranteed.
Prove:

- By mathematical definition of admissibility, $H(X,G) \leq D(X,G)$, where for any given X to a Goal, the heuristic of H(X,G) is always lesser than or equal to the actual distance D (X,G).
- Optimality is guaranteed with admissibility or underestimate.
- Assuming, A* has expanded path to goal node G.
- Then, A* has expanded all nodes N whose cost $f(N) < g(G)$. Since heuristic is admissible, at goal node G, then $h(G) = 0$. Therefore, $f(G) = g(G) + h(G) = g(G)$.
- Then, we can also be guaranteed that every unexpanded node N, has **$f(N) \geq g(G)$.** That is, every unexpanded node N must have f(N) greater than or equal to the actual path cost to G.
- However, if all h(n) is not an underestimate or not admissible, $\neg (H(X,G) \leq D(X,G)) =$ **$H(X,G) > D(X,G)$**
- Therefore, $\neg (f(N) \geq g(G)) = f(N) < g(G)$; that is for every unexpanded node N, there exist unexpanded nodes where **$f(N) < g(G)$.**
- This results in non-optimality of A* Search.

Visual Prove Case:

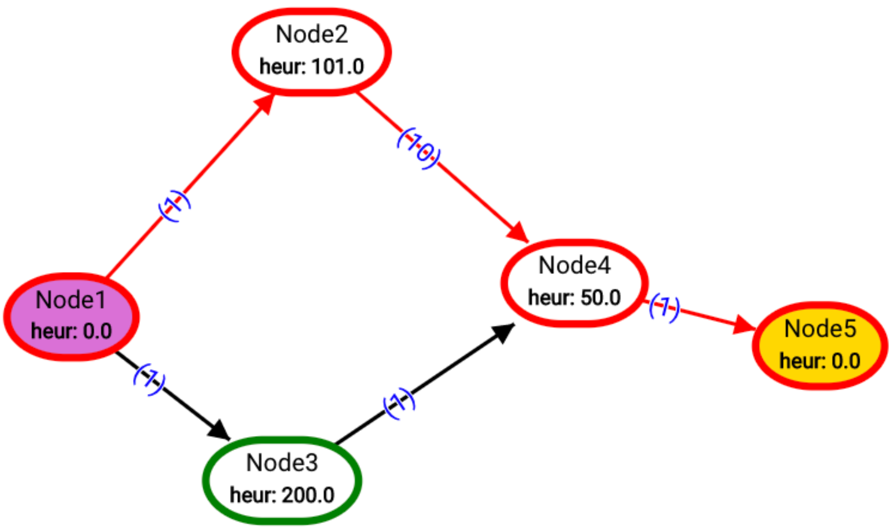


Figure 11 All heuristic is an overestimate

Non-Optimality (given that all heuristic is an overestimate):
Goal Path taken: Node1 → Node2→ Node4→Node5 (Cost 12)
Nodes expanded: 6
However, this isn't the optimal path, there exist an optimal path to goal node which is supposed to be Node1→Node3→Node4→Node5 (Cost 3)