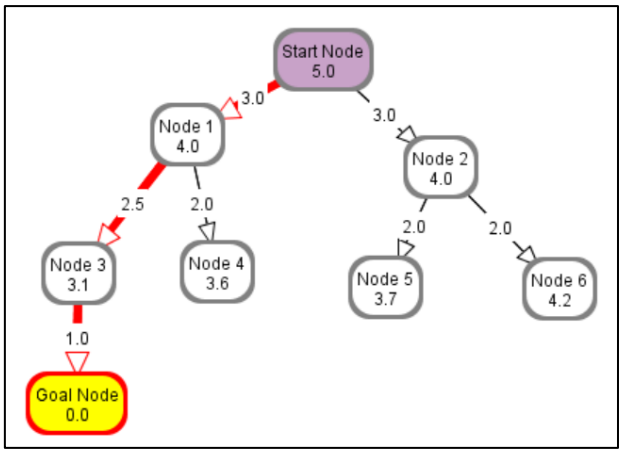
Lab Exercise 1

Problem Solving

Done by: Mock Jin Wei Tutorial/Lab Group: TSP 1

Question 1:

Q1 (a): Give a graph where DFS is much more efficient than BFS:



**Search Order is from Left to Right*

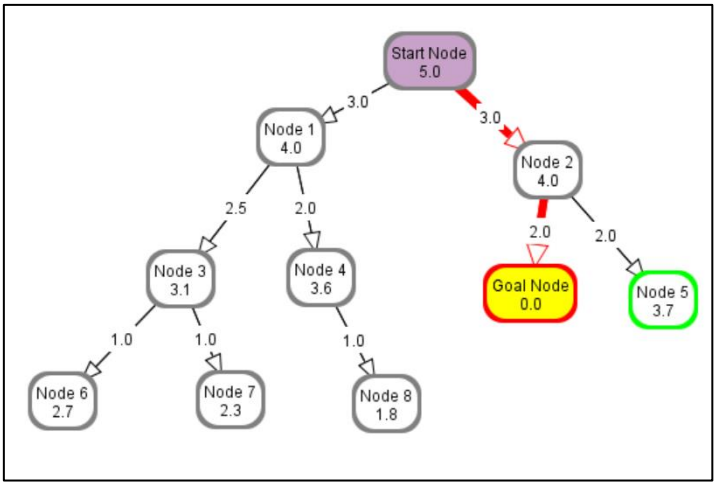
As seen from the above graph, DFS is performs better than BFS as:

- DFS expands **4** nodes to reach the Goal Node
  - Start Node → Node 1 → Node 3 → Goal Node

- BFS expands **8** nodes to reach the Goal Node
  - Start Node → Node 1 → Node 2 → Node 3 → Node 4 → Node 5 → Node 6 → Goal Node

Hence, DFS is better as it expands lesser nodes than BFS.

Q1 (b) Give a graph where BFS is much better than DFS.



**Search Order is from Left to Right*

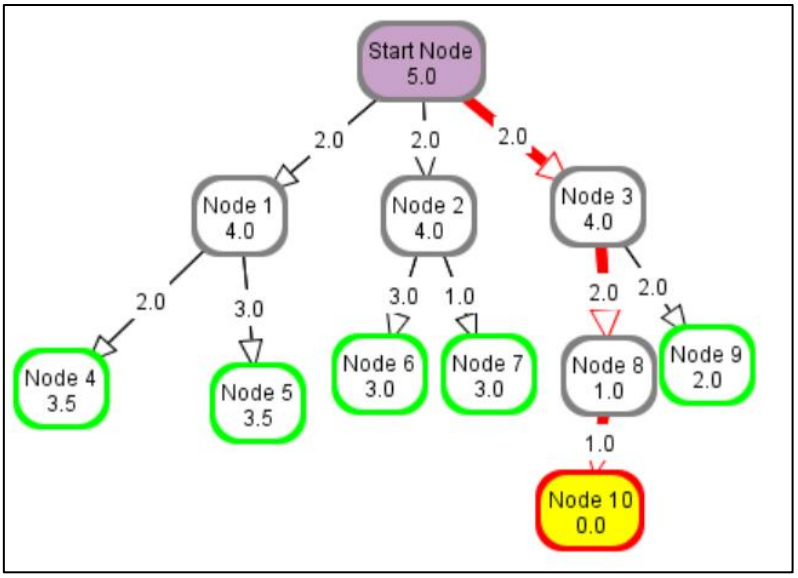
Based on the given graph above, BFS is more efficient than DFS as:

- BFS expands **6** nodes to reach the Goal Node
  - Start Node → Node 1 → Node 2 → Node 3 → Node 4 → Goal Node

- DFS expands **9** nodes to reach the Goal Node
  - Start Node → Node 1 → Node 3 → Node 6 → Node 7 → Node 4 → Node 8 → Node 2 → Goal Node

Since BFS expands lesser nodes that DFS to reach the Goal Node, BFS is a better searching algorithm here.

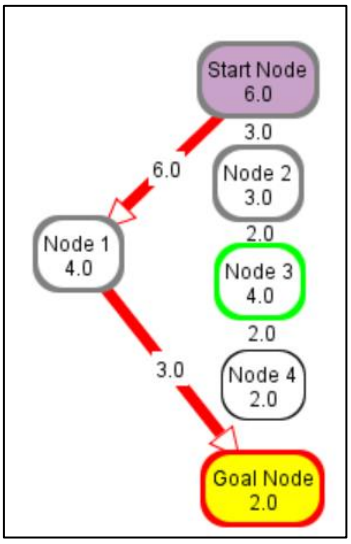
Q1(c) Give a graph where A* search is more efficient than either DFS or BFS.



With references to the above graph, the following result is observed:

- Under DFS, the algorithm will expand a total of **10** nodes before it reaches the Goal node.
- As for BFS, the algorithm will need to expand **11** nodes to reach the Goal node.
- A* search will requires the algorithm to expand **6** nodes before it reaches the Goal node.
- Hence, the graph above had shown that the A* search is much more efficient than DFS and BFS
  - 6 Nodes (A*) < 11 (BFS) or 10 (DFS) Nodes

Q1 (d) Give a graph where DFS and BFS are both more efficient than A* search.



The above graph shows the scenario whereby both DFS and BFS is much more efficient than A* search. This is because:

- DFS expands only **3** nodes to reach the goal node.
- BFS will expands **4** nodes to reach the goal node.
- A* search expands **6** nodes to reach the goal node.

---End of Question 1---

---Question 2 on the next page---

Question 2:

Q2 (a) What is the effect of reducing *h(n)* when *h(n)* is already an underestimate?

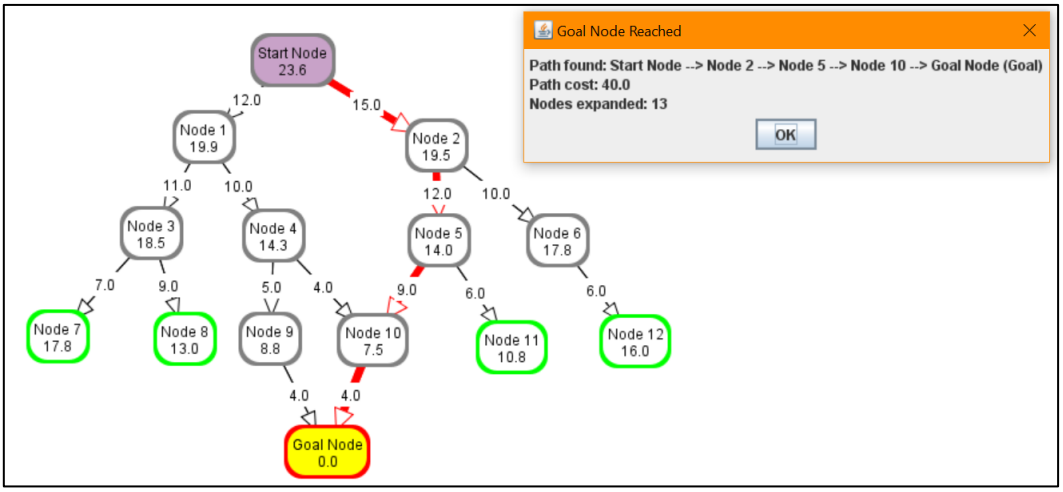


**Fig.1*

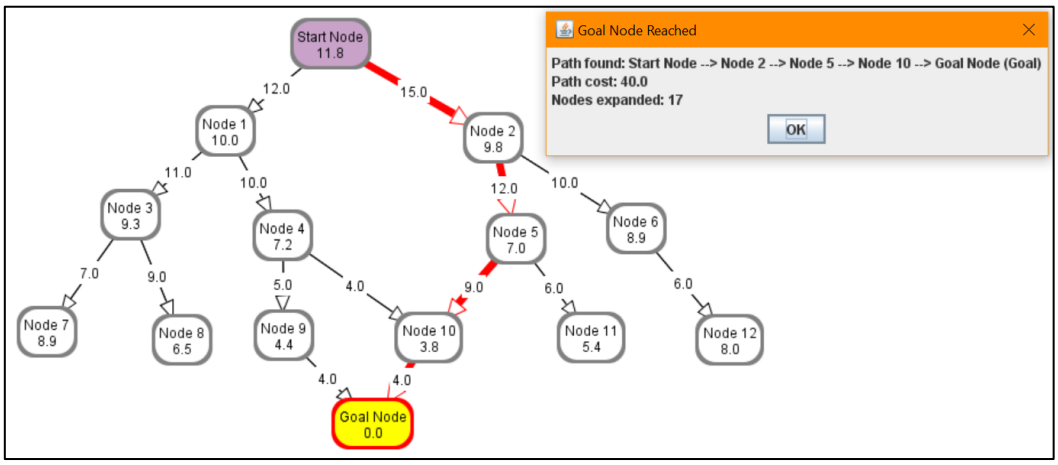
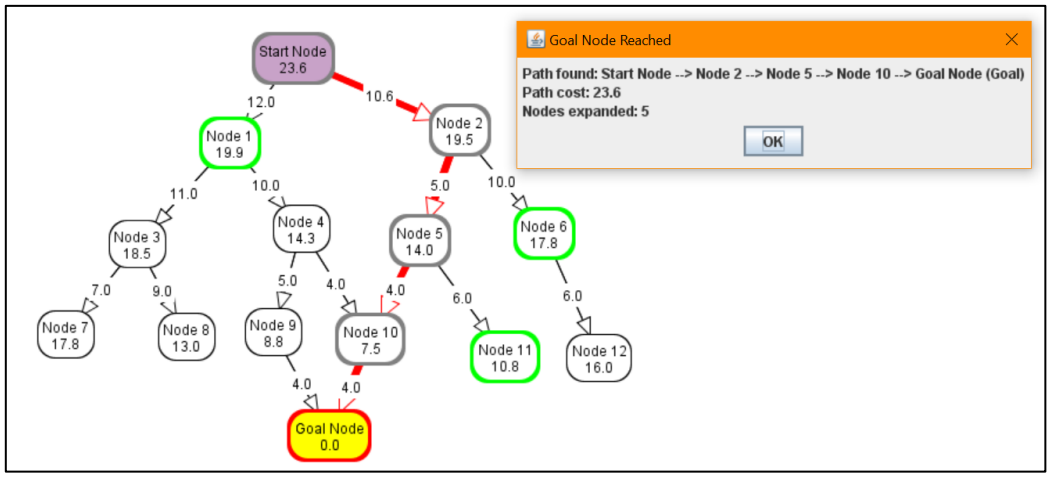


**Fig.2. h(n) value is half of the value shown in Fig.1*

- Based on Fig.1 and Fig.2 that is shown above, it can be observed that:
  - For Fig.1
    - The path found is: **Start Node → Node 2 → Node 5 → Node 10 → Goal Node**
    - Path cost is: 15.0 + 12.0 + 9.0 + 4.0 = **40**
    - Total Nodes expanded: **13**
  - For Fig.2
    - The path found is: **Start Node → Node 2 → Node 5 → Node 10 → Goal Node**
    - Path cost is: 15.0 + 12.0 + 9.0 + 4.0 = **40**
    - Total Nodes expanded: **17**
- Hence, it can be conclude that reducing *h(n)* when *h(n)* is already an underestimate will result in:
  - The optimal path will still always be found no matter how low *h(n)* is.
  - However, in return, as the value of *h(n)* gets lower, A* search will expands more nodes than before. This could make the A* search slower.

Q2 (b) How does A* perform when *h(n)* is the exact distance from *n* to a goal?



- With reference to the above graph,
  - *h(n)*: **23.6**
  - Path Cost: 10.6 + 5.0 + 4.0 + 4.0 = **23.6**
- Upon running the A* search when *h(n)* is the exact distance from *n* to a goal, the algorithm will take the best available path and will not expand any other nodes. This could possibly makes the A* search more efficient and faster.

Q2 (c) What happen if *h(n)* is not an underestimate?

- When *h(n)* is not an underestimate, there could be chances that either *h(n)* will be of an exact distance from *n* to a goal or it can be an overestimate.
- In the case of *h(n)* being the exact same distance from *n* to a goal, the result will be the same as the observation made in Q2 (b).
- If *h(n)* happens to be an overestimate, it is possible that A* search will not find the optimal path as the algorithm may expands other unnecessary nodes. However, this is very dependent on the given problem itself and different results might be found for different problems. Various graphs such as the one below has been drawn to come to the conclusion

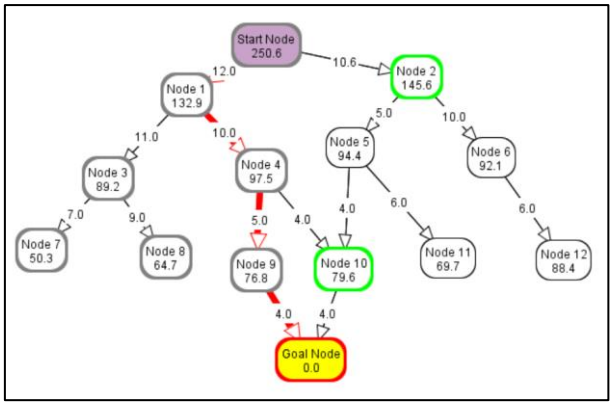


**Fig.3 .h(n) is an overestimate*

---End of Question 2---

---End of Lab 1 Exercise---