

**CZ3005 Artificial Intelligence**

**Lab Exercise 1**

Name: Loh Yi Xuan Renice

Matriculation No.: U1822247D

Date: 25 September 2020

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**NANYANG TECHNOLOGICAL UNIVERSITY**

**Question One [55 marks]**

**For each of the following, give a graph that is a tree (there is at most one arc into any node), contains at most 15 nodes, and has at most two arcs out of any node.**

**(a) Give a graph where Depth-first Search (DFS) is much more efficient (expands fewer nodes) than Breadth-first Search (BFS). [10 marks]**

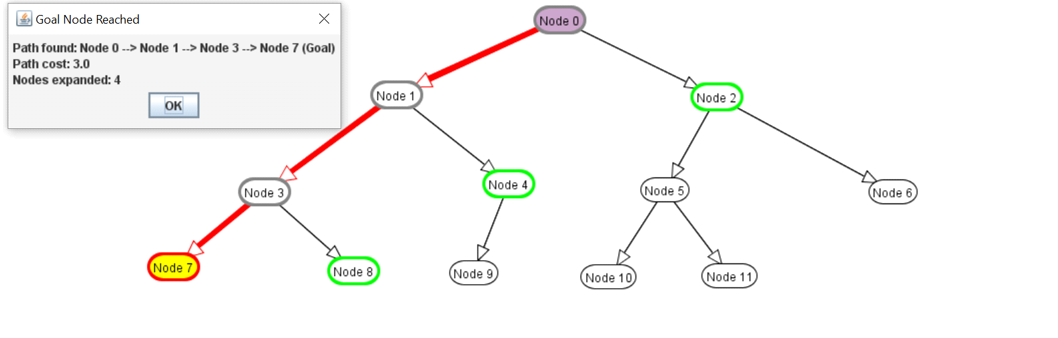


Figure 1.1.1: DFS

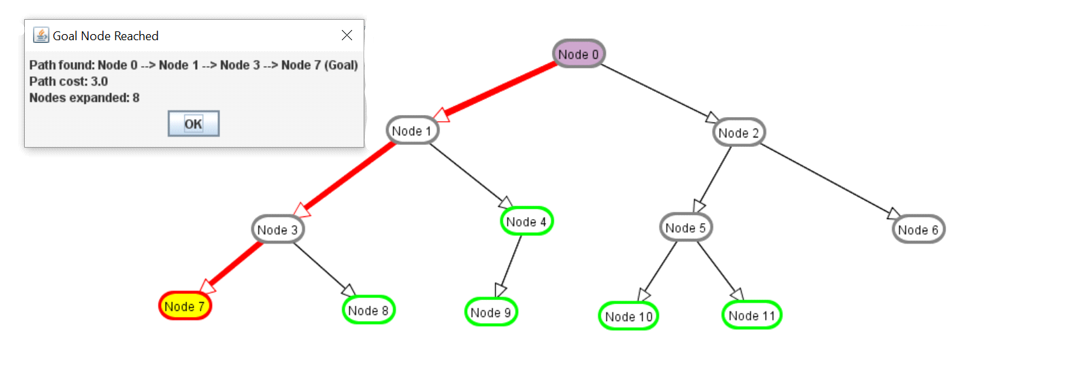


Figure 1.1.2: BFS

**(b) Give a graph where BFS is much better than DFS. [15 marks]**

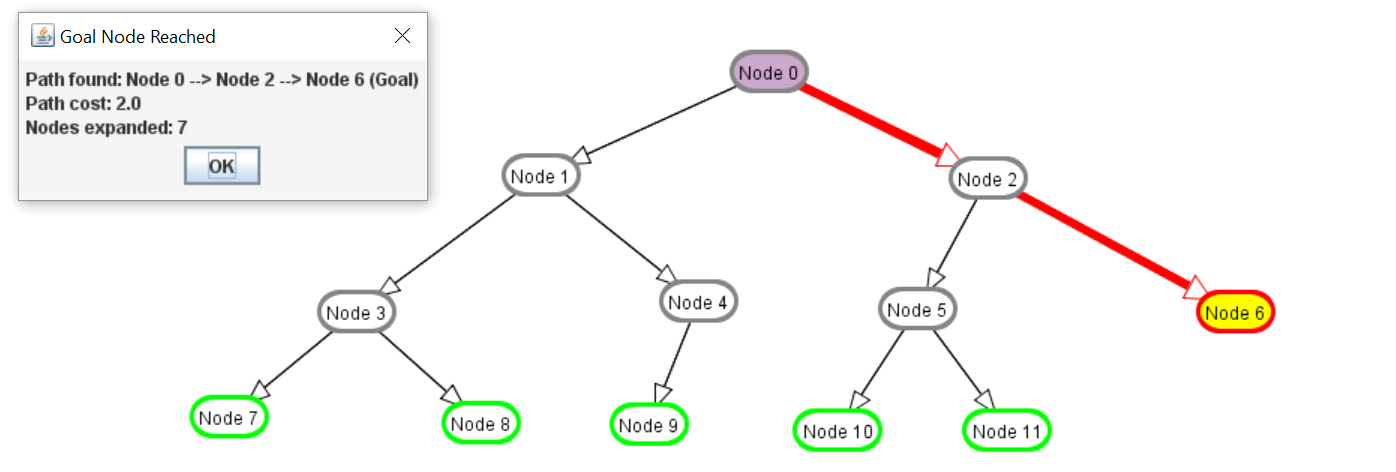


Figure 1.2.1: BFS

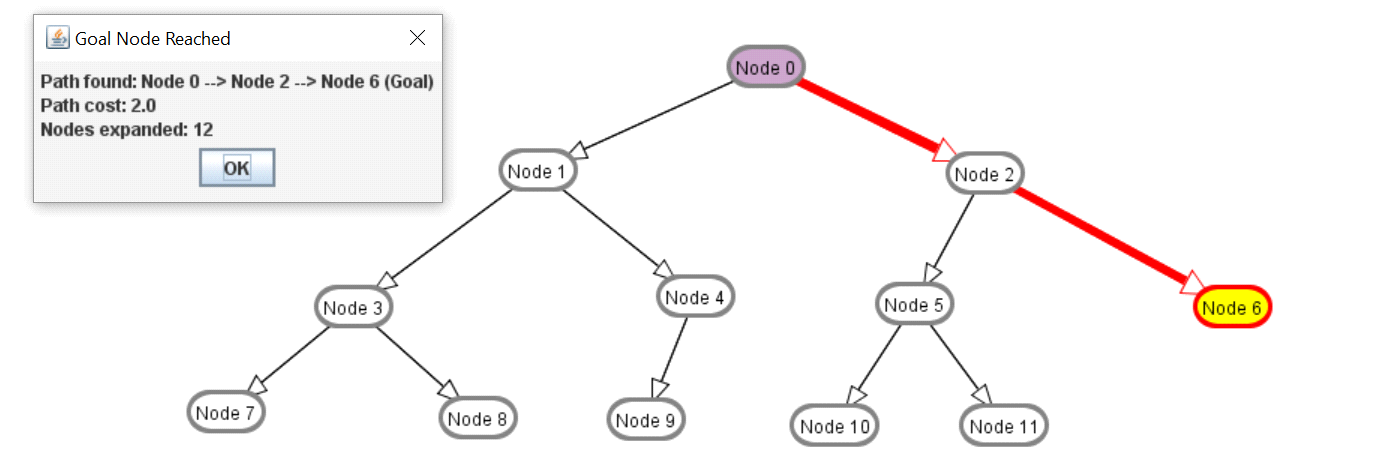


Figure 1.2.2: DFS

**(c) Give a graph where A\* search is more efficient than either DFS or BFS. [15 marks]**

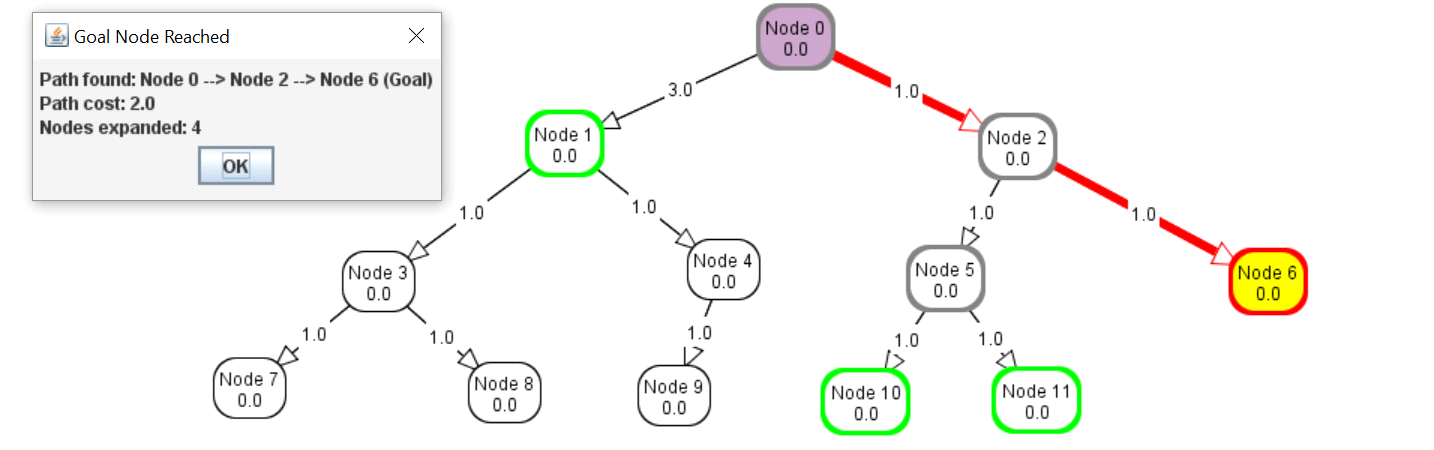


Figure 1.3.1: A\* Search

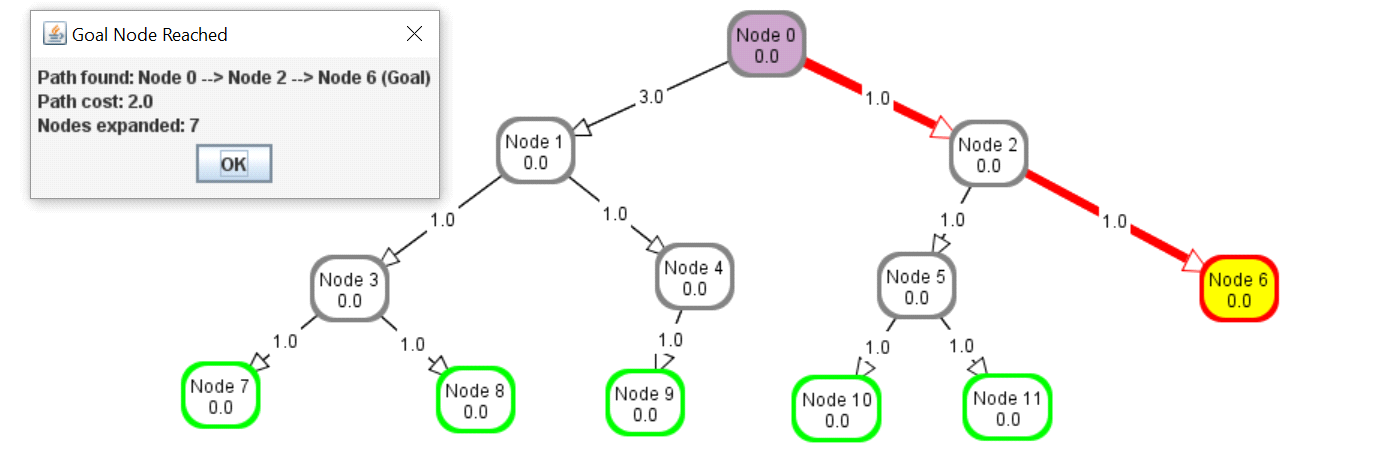


Figure 1.3.2: BFS

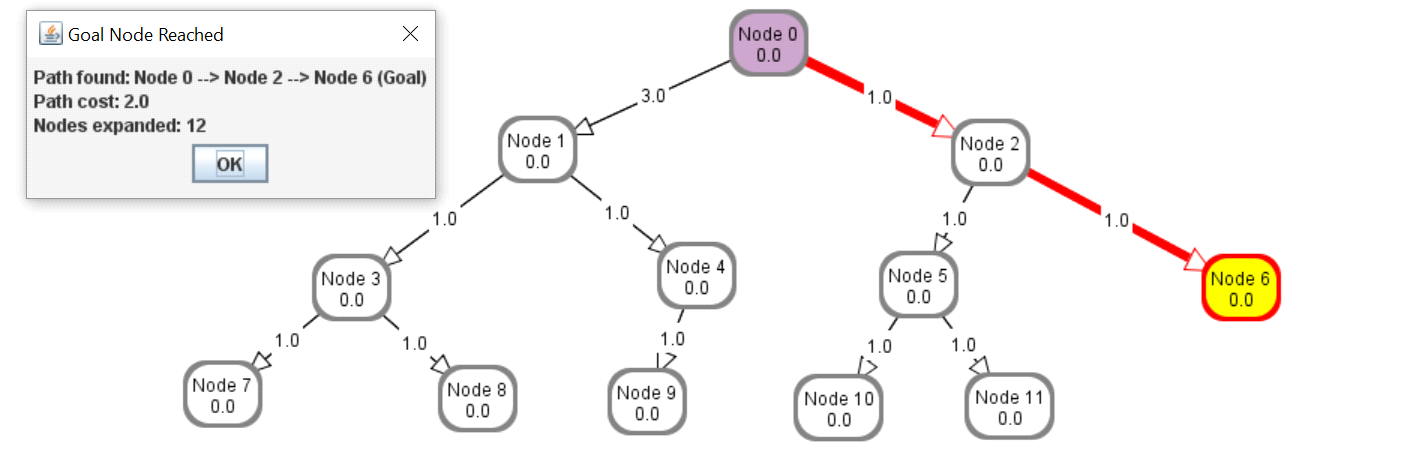


Figure 1.3.3: DFS

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

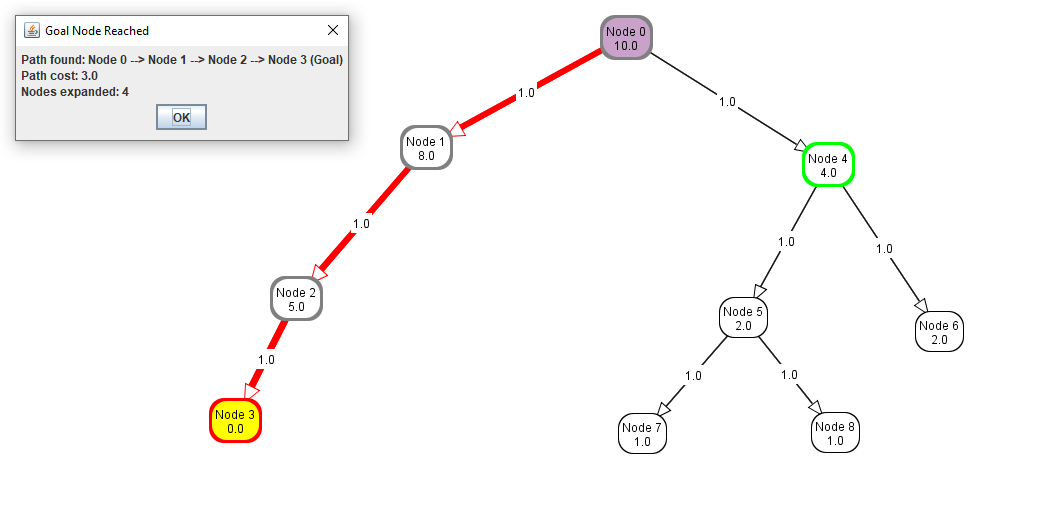


Figure 1.4.1 DFS

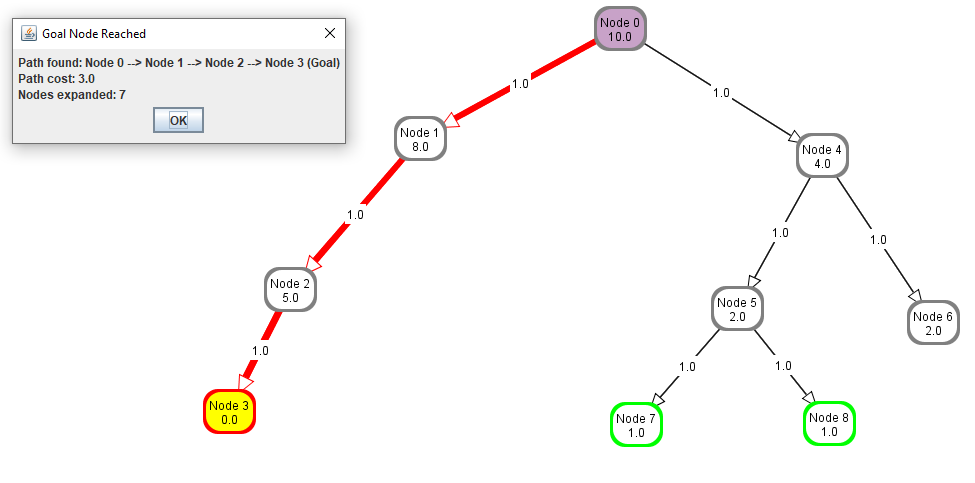


Figure 1.4.2 BFS

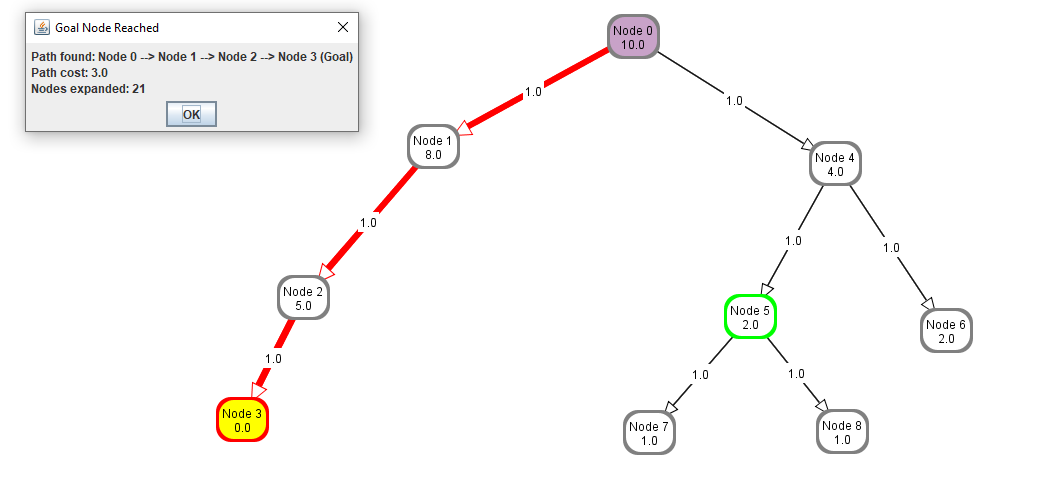


Figure 1.4.3 A\*Search

**Question Two [45 marks]**

**Progress is made in science by observing a phenomenon of interest, making hypotheses, and testing the hypothesis either empirically or by proving theorems. For this question you are to think about the effect of heuristic accuracy on A\* search. That is, you are to experiment with, and think about how close h(n) is to the actual distance from node n to a goal affects the efficiency and accuracy of A\*. To get full marks you must at least invent one (plausible, nontrivial) conjecture and either prove it and show some empirical evidence for your answer or show that it is false. Your answers need to be precise (e.g., don’t say “it works better”, but say something like “it always works better”, “it sometimes works better” or “it works better in a majority of cases”).**

**(a) What is the effect of reducing h(n) when h(n) is already an underestimate? [15 marks]**

The effect of reducing h(n) when h(n) is already an underestimate will increase the number

of node expanded. However, optimal solution will still be found.

The following graph node and heuristic value is automated generated:



A\* search with normal heuristics: 5 nodes expanded, and optimal solution is found.

A\* with 0.97x heuristic value: 5 nodes expanded but optimal solution is still found.

A\* with 0.95x heuristic value: 5 nodes expanded but optimal solution is still found.

A\* with 0.5x heuristic value: 8 nodes expanded but optimal solution is still found.

A\* with 0.1x heuristic value:11 nodes expanded but optimal solution still found.

This finding shows that, not necessary reducing h(n) when h(n) is already an underestimate

will increase the number of nodes. The reason is that a different optimal path may be found.

One true statement is that any sub-optimal path explored with *h1* will be explored

with *h2* if *h2(n) <= h1(n) <= cost(n,g)*, where *cost(n,g)* is the actual cost from *n* to *g*.

**(b) How does A\* perform when h(n) is the exact distance from n to a goal? [15 marks]**

The performance of A\* depends on what happens when there are multiple optimal paths to the goal. If the frontier acts as a stack for nodes with equal *f*-values, then it will proceed to the goal without expanding any node off a single optimal path.

**(c) What happens if h(n) is not an underestimate? You can give an example to justify your answer. [15 marks]**

This means that h(n) could be an overestimate or a correct estimate. The correct estimate will not affect the search. On the other hand, if h(n) is an overestimate, A\* has no idea when it can stop exploring a potential path as there can be path with lower actual cost but higher estimated cost than the best currently known path to the goal.

For example, given that h(n) is an overestimate and the algorithm reach the goal node by some other path with the distance less than f(n) then it will never look at node n because h(n) is seemingly worse than an established path. Thus, even if the actual distance via n is optimal, the path will be overlooked, hence A\* may not guarantee to produce optimal behaviour.