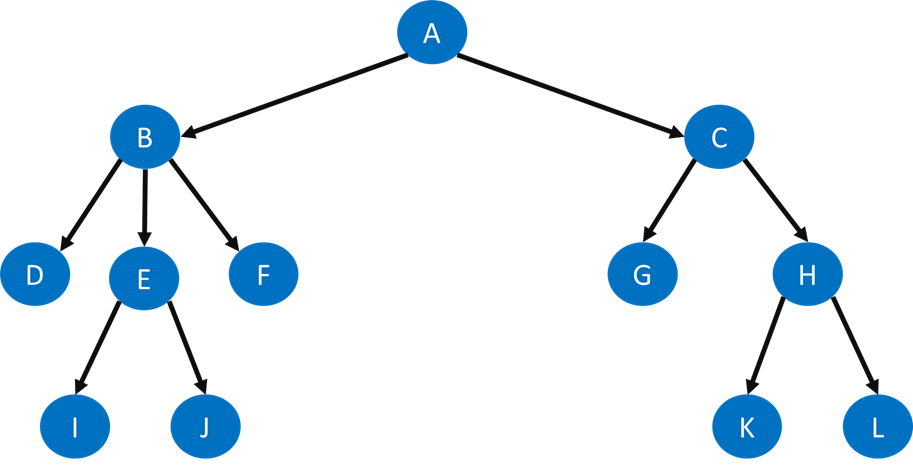
Assignment 3

Depth vs Breadth First Search

1. For the graph below, show the sequence in which the nodes will be expanded when performing:
   1. Breadth First Traversal
      1. [A, B, C, D, E, F, G, H, I, J, K, L]
   2. Depth First Traversal
      1. [A, B, D, E, I, J, F, C, G, H, K, L] - (without revisits)
      2. [A, B, D, B, E, I, E, J, E, B, F, B, A, C, G, C, H, K, H, L] (with revisits)



1. Assuming that all edges have the same cost, will Breadth-First Search always return the most cost-optimal path to a target node?
   1. Yes.
2. What about Depth-First Search?
   1. No, it returns the first solution which may not be the most cost-optimal solution.
3. Explain why.
   1. Breadth-first search finds the cheapest route, but is memory intensive. Depth-first search tries to solve the memory problem of breadth-first search, but in doing so doesn’t necessarily find the cheapest route. It returns the first solution it finds, and there may be another path that is cheaper that it hasn’t found yet. It searches each branch of the tree all the way to the end, which may be a possible solution, just not the cheapest. It only has to store information for the current path though.
4. Explain why Breadth-First Search requires more memory than Depth-First Search when traversing very large trees.
   1. It has to store all the information about all the nodes it visits until it reaches a goal state.