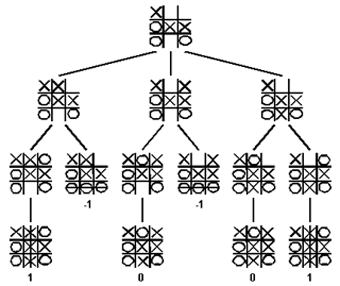
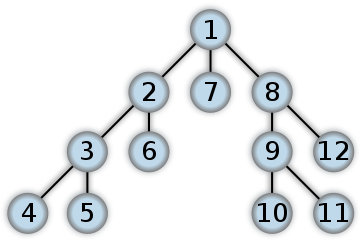
# Usages of Depth-First, Breadth-First and Best Fit Algorithms

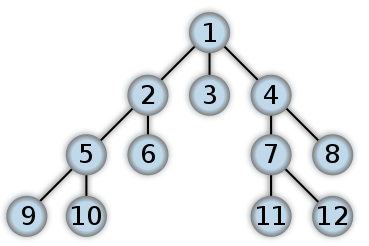
# Depth-First:

1. Relies on stack to store vertices.
2. Suitable in graphs or trees where the target is deep and away from the root.
3. Suitable in games because it explores a possible solution till its end; if it leads to a win than the algorithms stop, if not then it backtracks and tries another solution.
4. It has a linear space complexity of O(bd). (lower than both Breadth and Best)
5. It doesn’t provide the shortest path between two nodes.

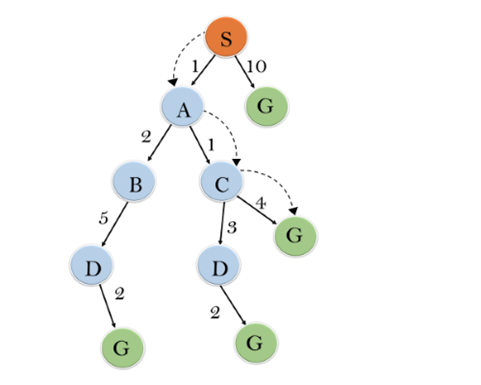


# Breadth-First:

1. Relies on queue to store vertices.
2. Suitable in graphs where the target is close to the root.
3. Guaranteed to find the shortest path between two nodes (in an unweighted graph)
4. Space complexity is exponential b^(d+1)



# Best-fit:

1. Uses an evolution function to decide which adjacent node is most promising.
2. It can switch between both Depth-First and Breadth-First, gaining the adv. of both.