

EE599

Hw6

<https://github.com/Wenhaocui/HW6.git>

1.

A tree is an undirected graph.

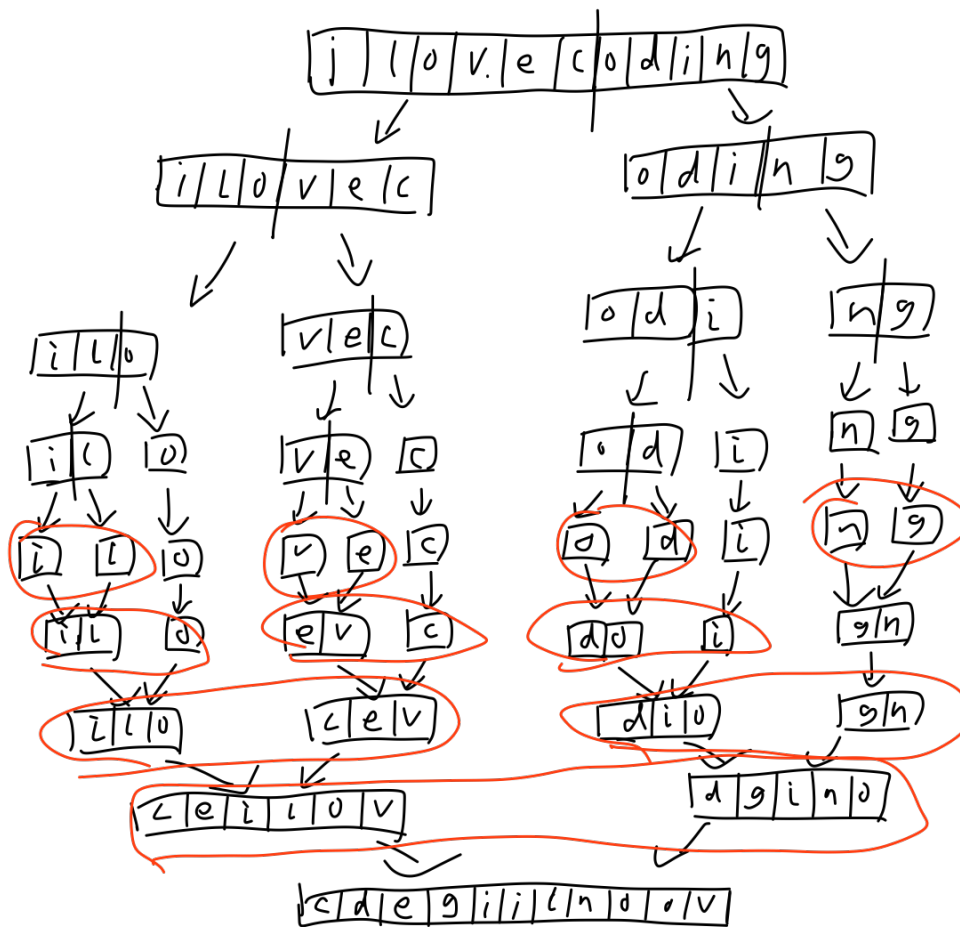
A tree is a connected graph.

A tree is an acyclic graph.

In a tree, there is a path from each vertex to all other vertices.

A simple graph is a graph that does not have more than one edge between any two vertices and no edge starts and ends at the same vertex. (no self-loops and no multiple-edges)

2.



3.

Runtime complexity is $O(V+E)$ Because I need to check each vertices is visited or not, which means all the graph vertices, and go through adjacent nodes of each checked nodes, which means all the graph edges.

4.

Runtime complexity is $O(V+E)$ Because I need to check each vertices is visited or not, and finished when each nodes is visited, and go through adjacent nodes of each checked nodes, and if traverse from this root does not connect with any other nodes (because directed), we check the result vector to check if there are all nodes in there. if it is not, start the unvisited node as root. which means we need to traverse all the graph nodes and edges.

5.

If each path is not blocked, which means we might traversed whole maze, because I switch the number 1 to 2 whenever I went through one piece, which means the worst case runtime complexity is $O(n*m)$ (n is row, m is column)

6.

Runtime complexity is $O(n)$

For loop from low to high is linear and swap is constant.