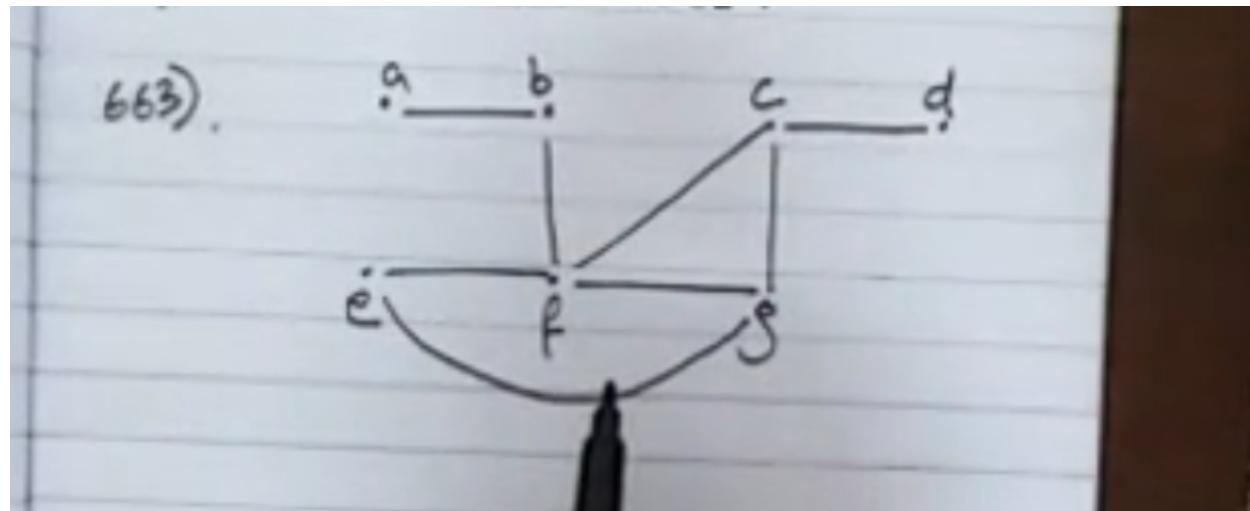
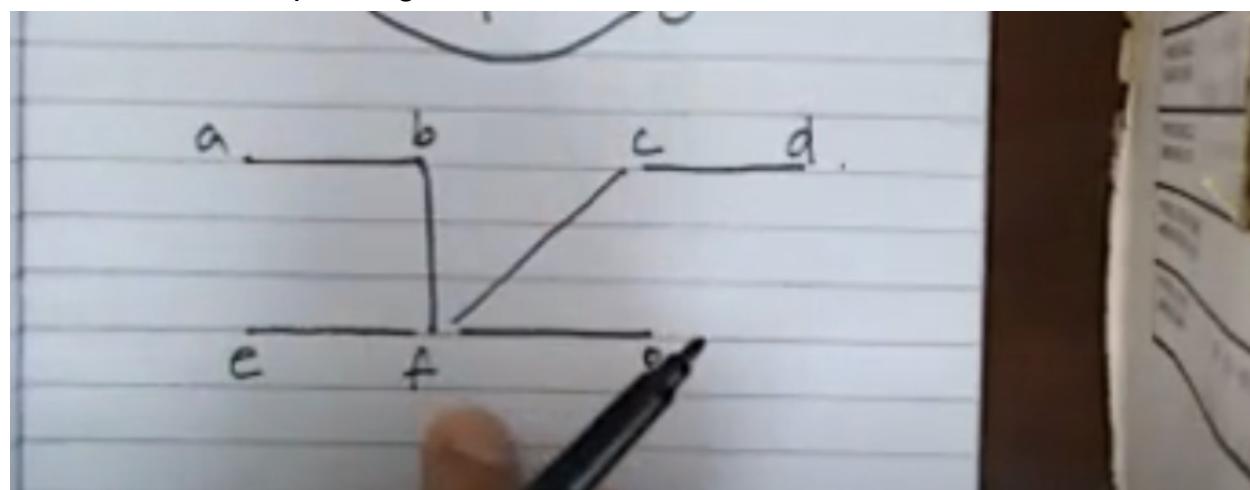


## Discrete Lecture # 29

- Spanning Tree
  - Connected
  - Do not have a loop
  - Undirected
  - Sub graph
  - All vertices
- Example

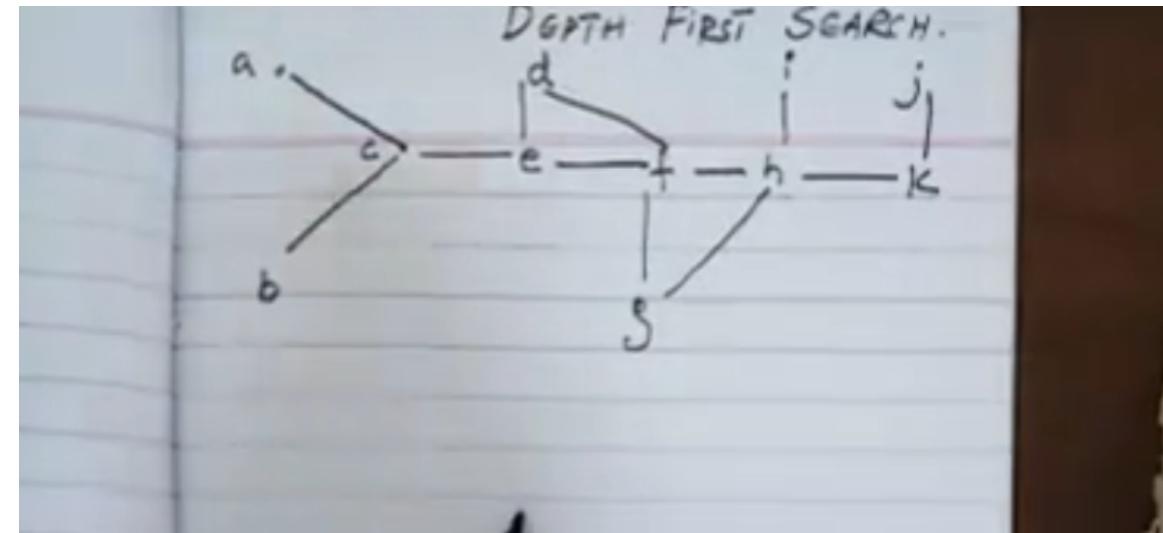


- A graph made from this with all the spanning trees conditions will be considered as its spanning tree

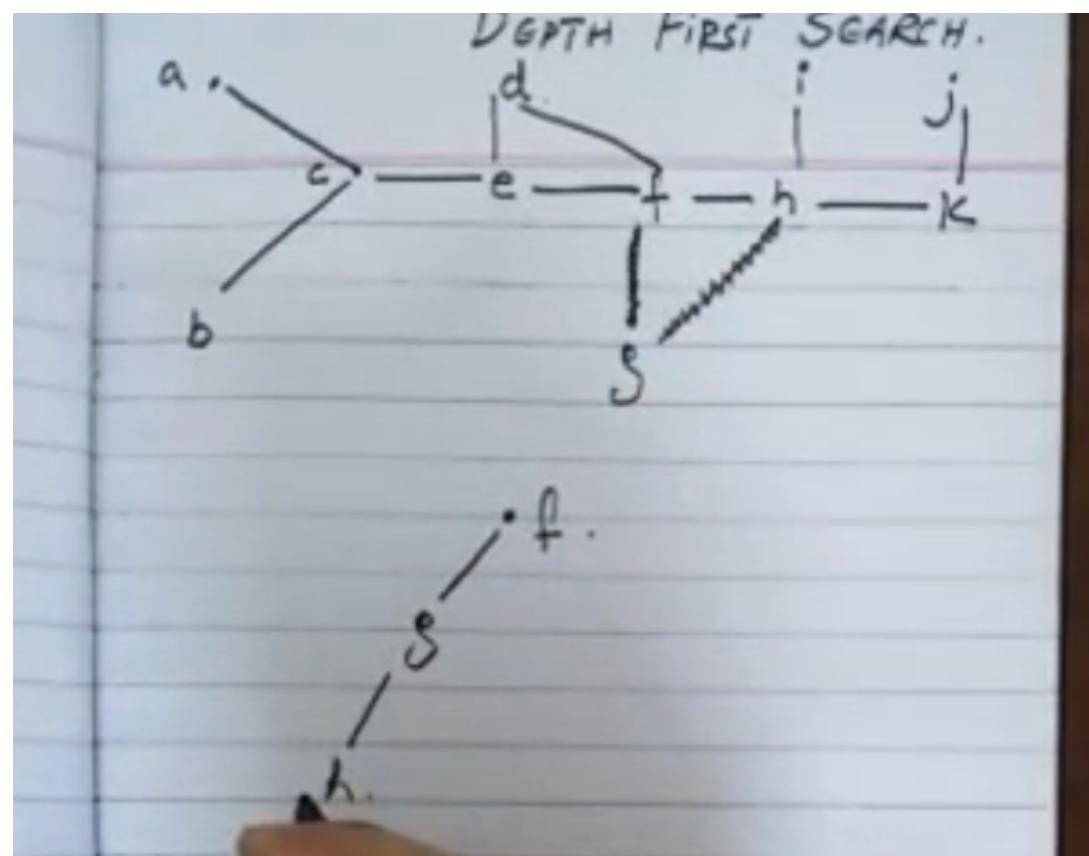


- Connected
- No loop
- undirected
- Is a sub graph

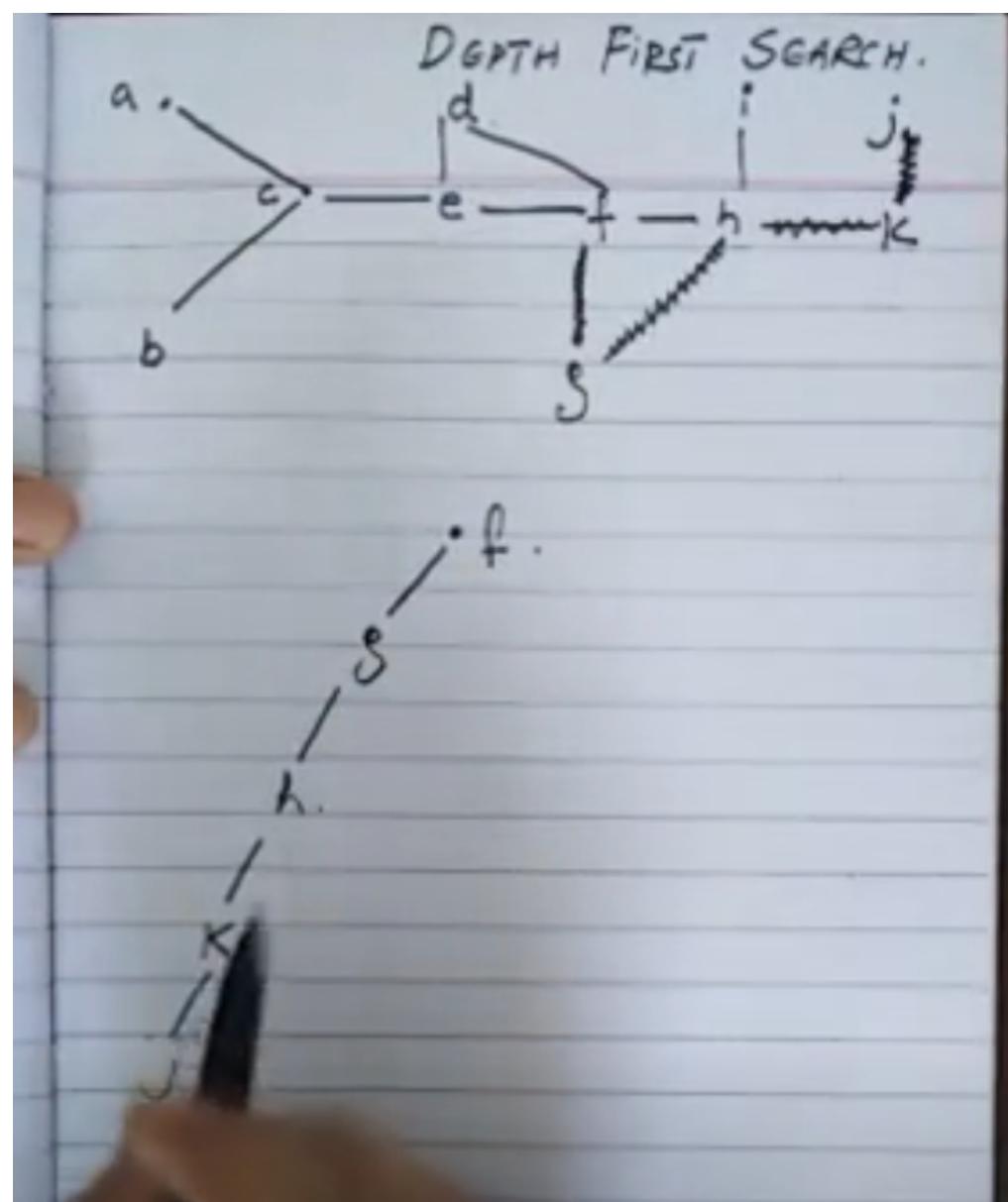
- Contains all vertices
- Algorithms for constructing spanning trees
  - Depth first algorithm



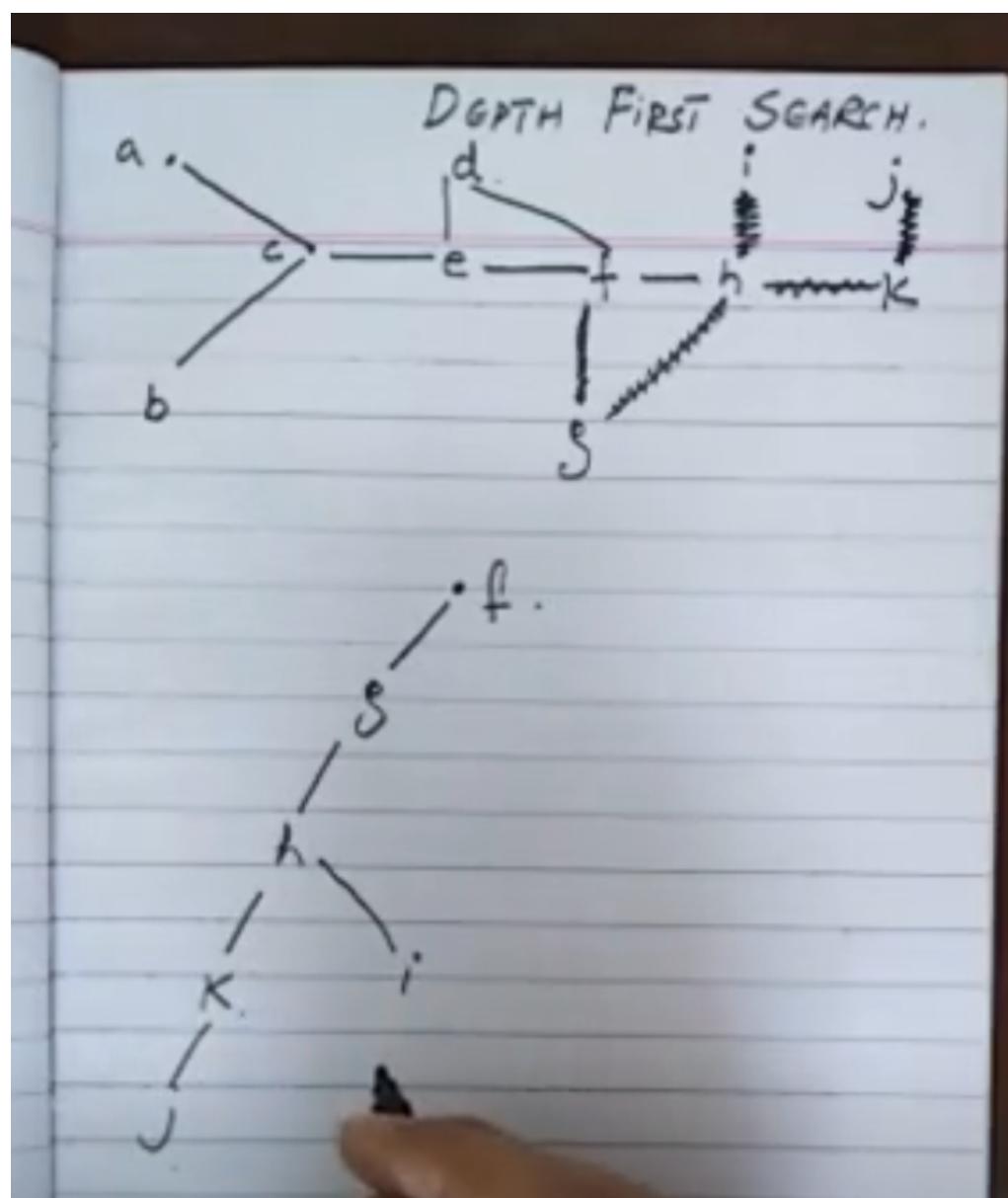
- Pick any vertex with greater degree
- Traverse towards a vertex and you cannot go back once you cover this edge



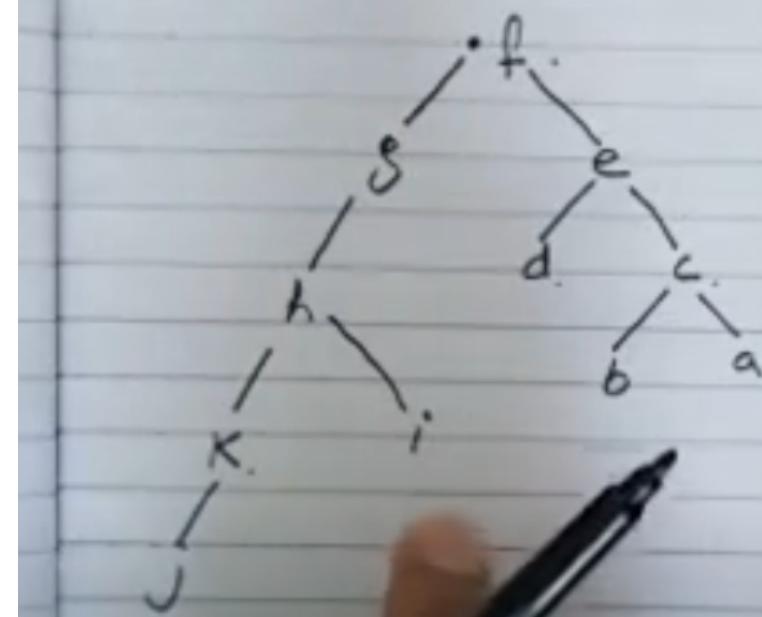
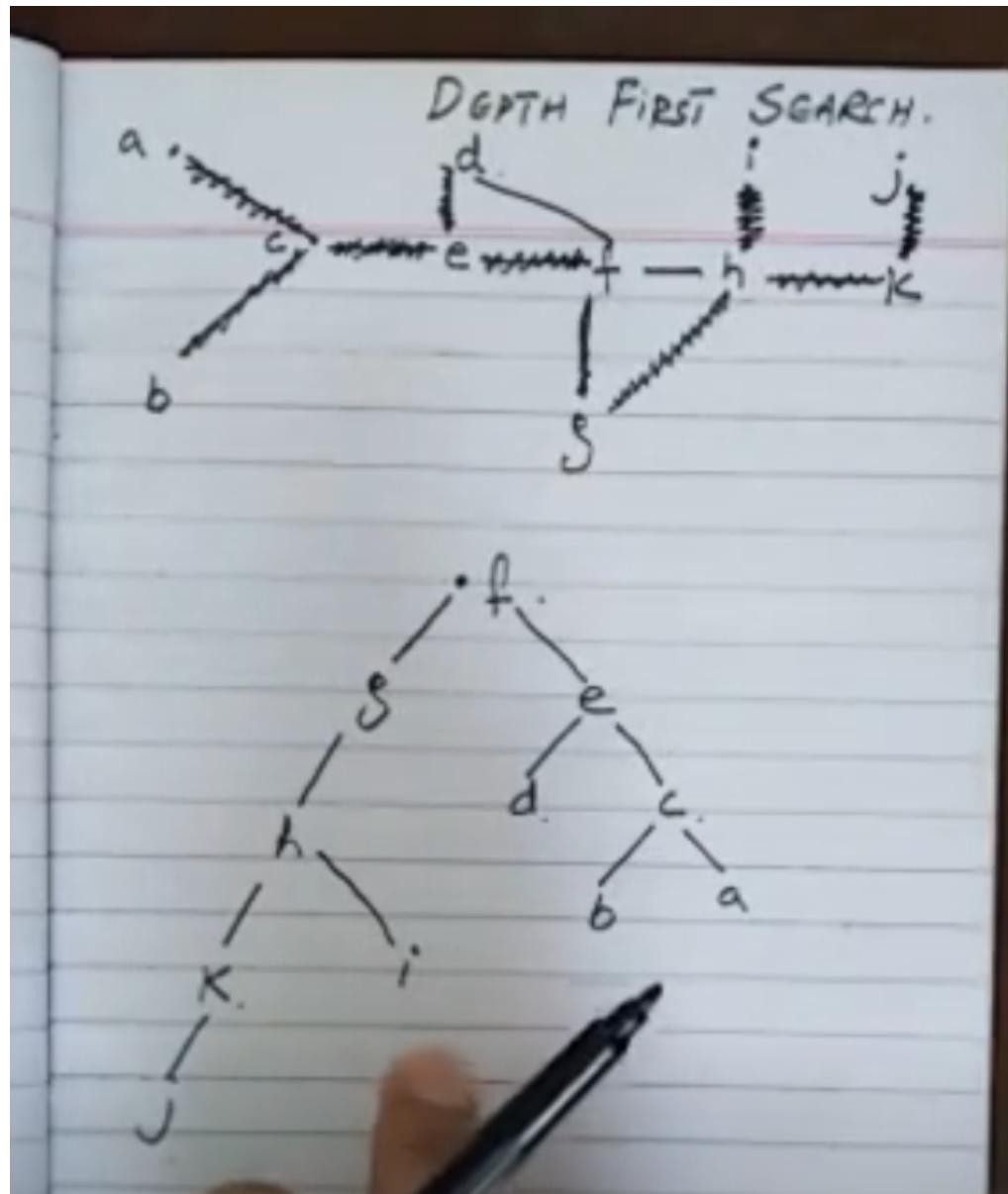
- We chose f
- Traversed to g
- Cannot go back to f
- We have only one path to h we go there
- Now from H we only two path to i and to k because we cannot go back to f as we have already covered it and dont wanna cover it again



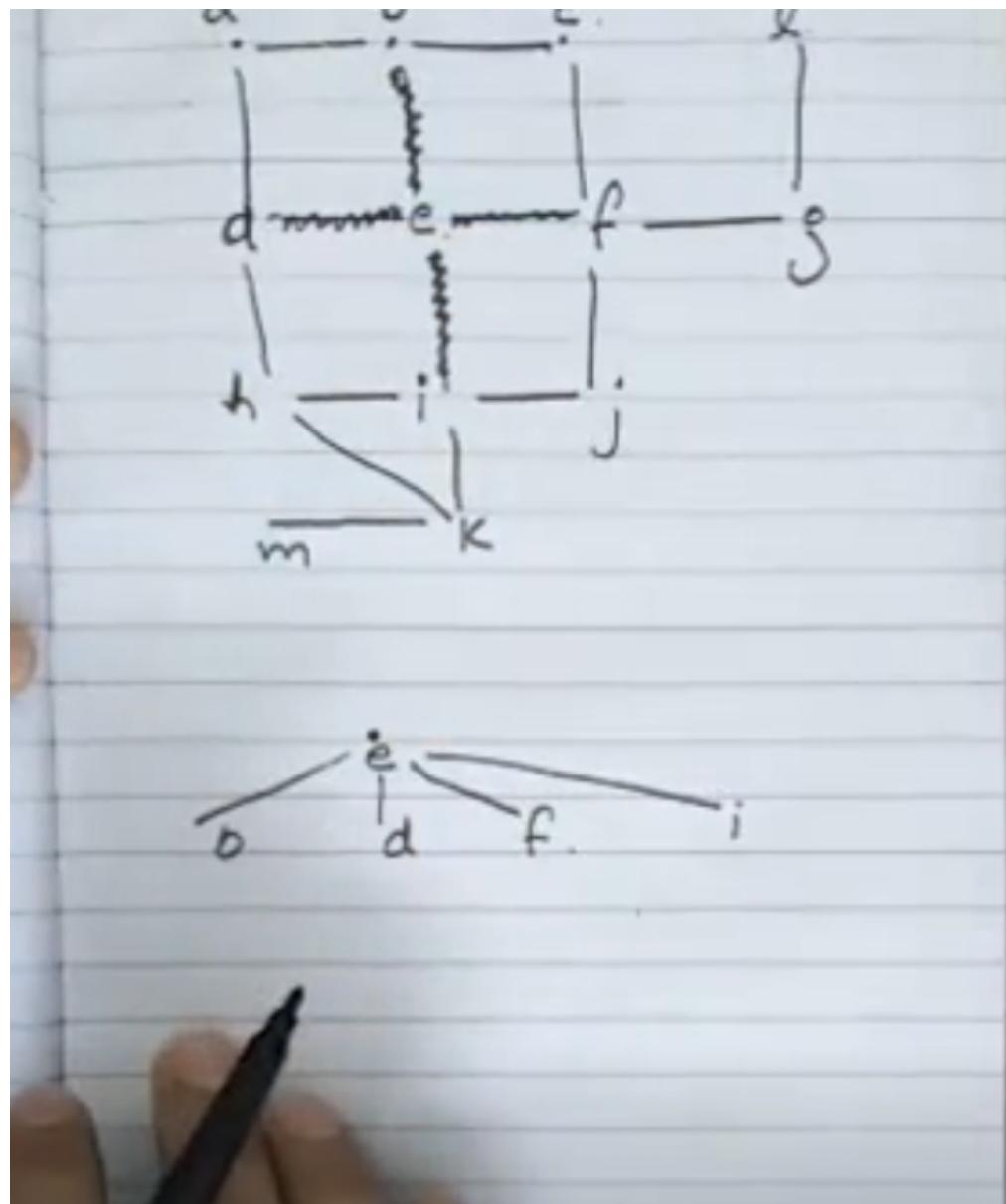
- Now we reach a DEAD END and now we have to back track to h as we have one undiscovered path



- Now we are again at DEAD END
- We back track to h now we cannot cover f as f is already covered
- We back track to g and to f
- Continue the same process and you will make this

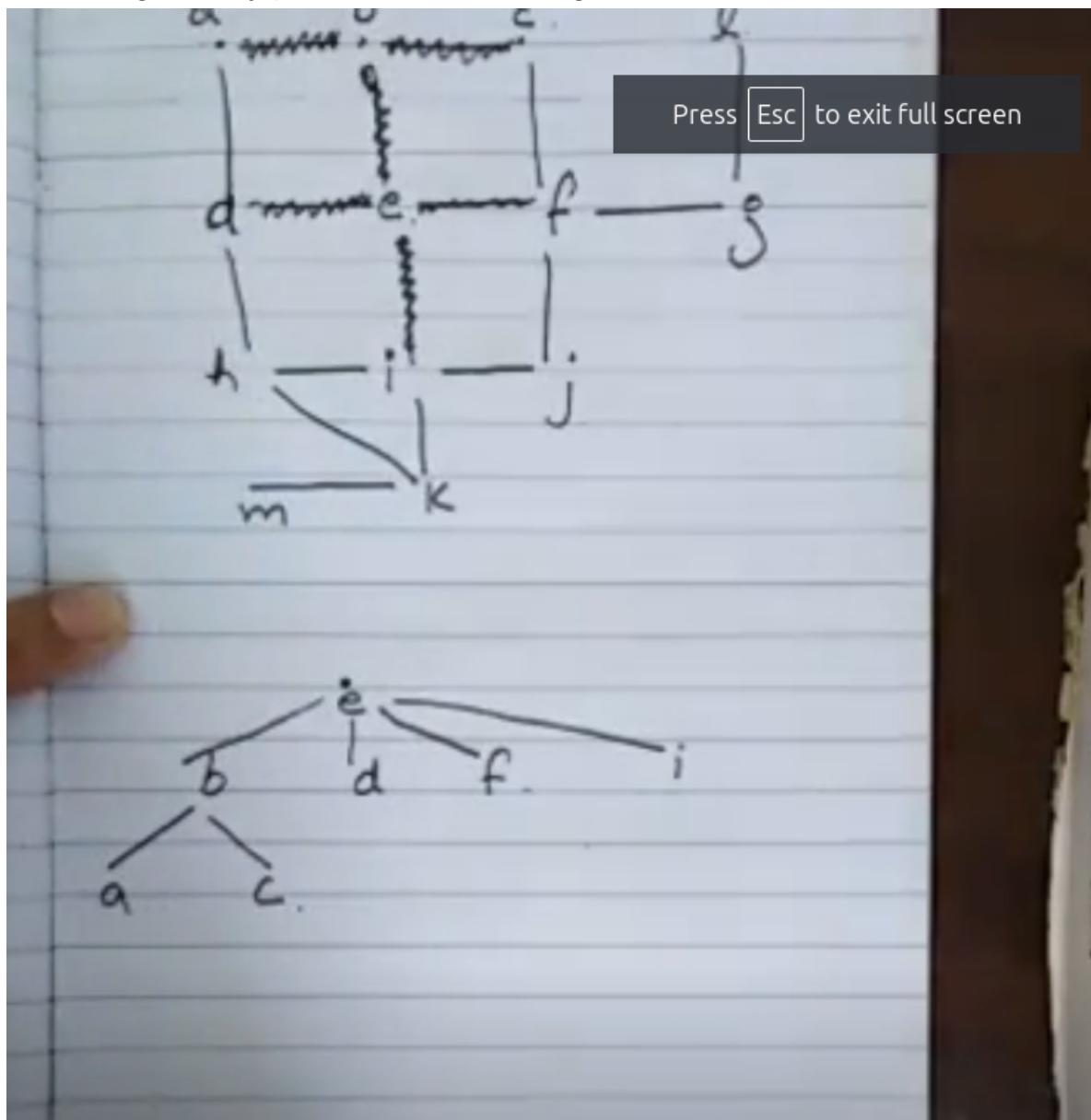


- This is a depth first graph
- It also a sub graph
- Depth first search wont give you always a unique tree
- Breadth first algorithm

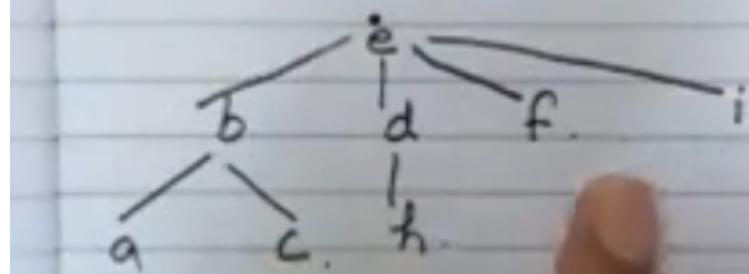
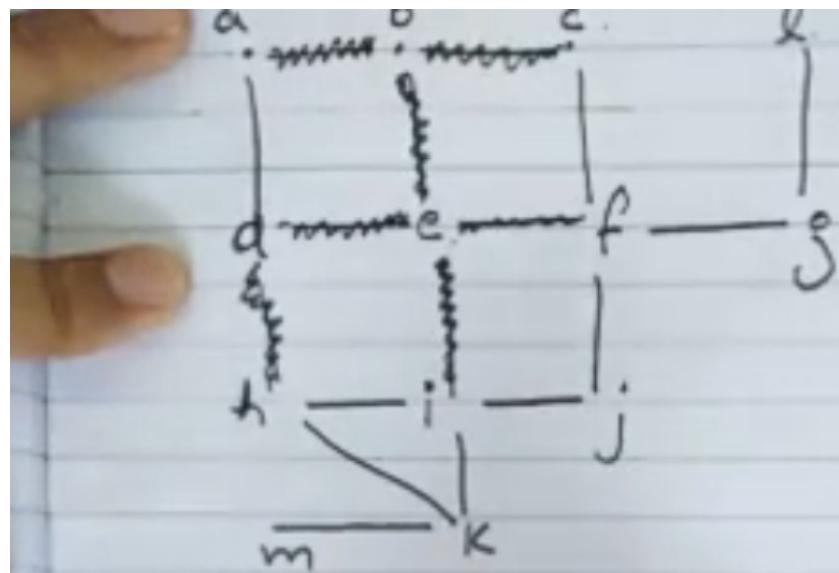


- 
- In breadth first algo we choose a vertex
- Then we make all the adjacent vertices childs of the chosen root vertex
- When all adjacent vertices are covered

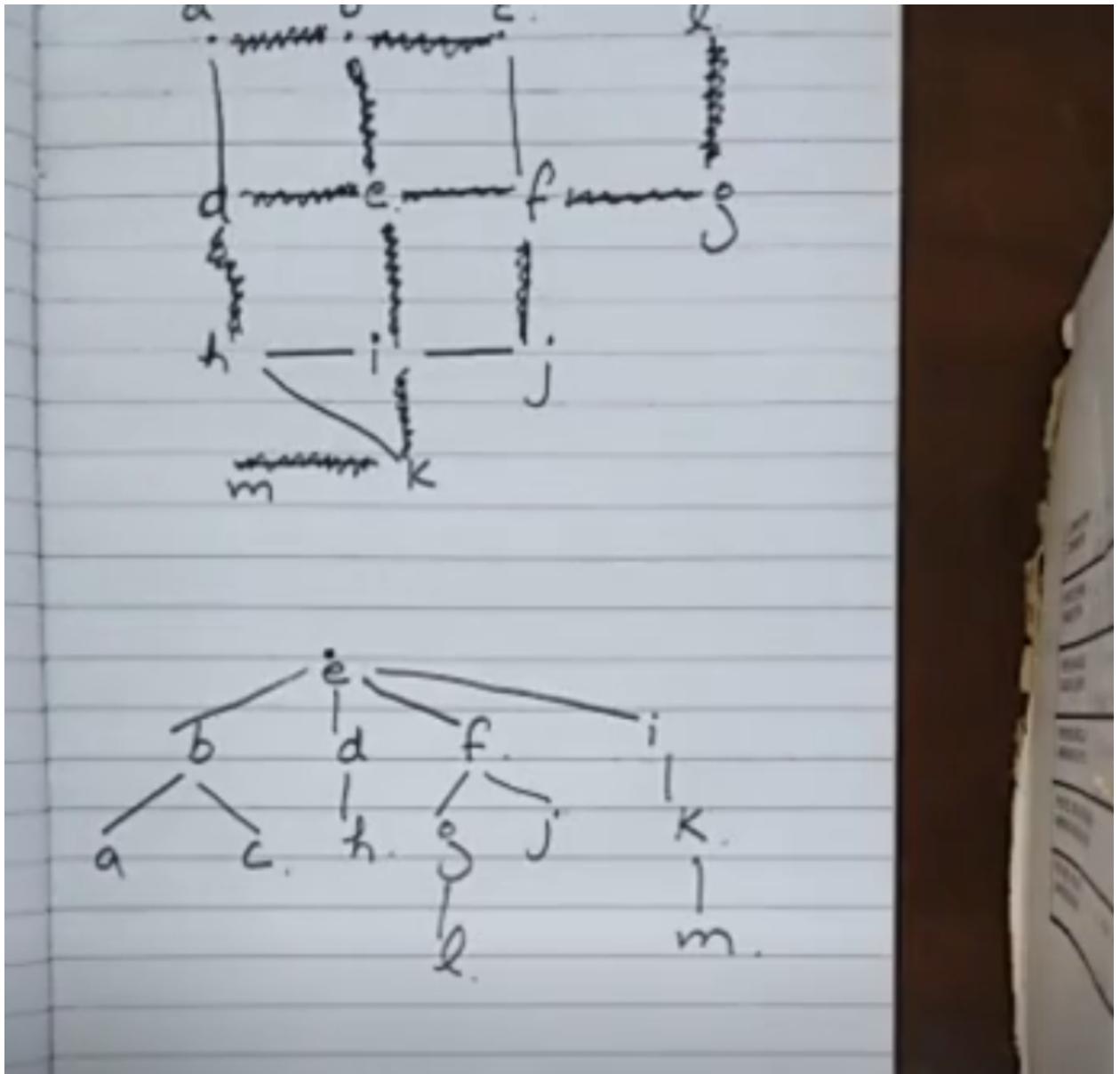
- Then we generally process from left to right



- We chose b and covered all of its adjacent vertices
- We do not include those vertices that are covered once



- 
- Now for d
- As all of d's adjacent vertices are already covered in the graph so we traverse towards h only
- Continue the process and in the end you will have a graph



- Difference between depth first and depth first
  - In depth first our tree will be depth-wise lengthy
  - In breadth first our tree will be breadth wise lengthy