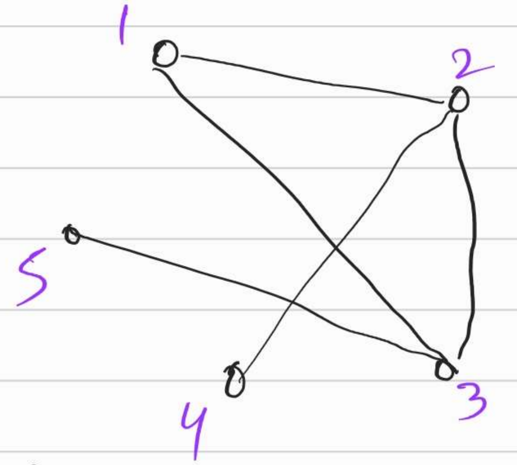
28/02/2022

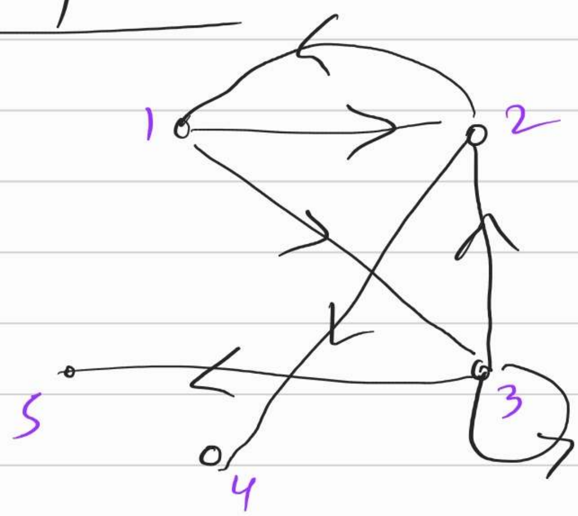
Graphs



V = Set of vertices/ nodes

E = Set of edges (pair of nodes)

Directed graphs



Walks: a sequence of vertices s.t. $\exists$ an edge between two consecutive vertices.

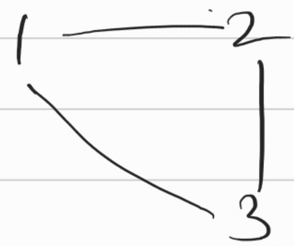
1 — 2 — 1 — 3 — 1 — 2 — 3 — 5 (undirected)

1 − 2 − 1 − 3 − 3 − 2 − 4 (directed)

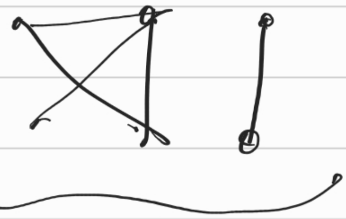
Paths: It's a walk s.t. vertices are not repeated.

1 - 2 - 3 - 5

Cycles :

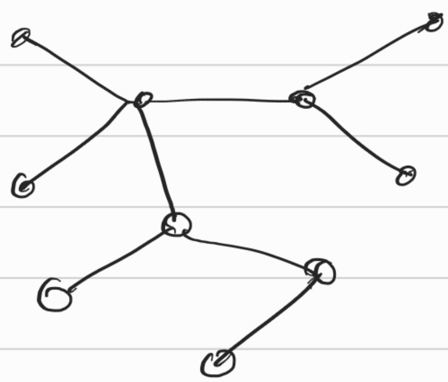


Connected graph :- It's a graph s.t. there is a path between any two vertices.

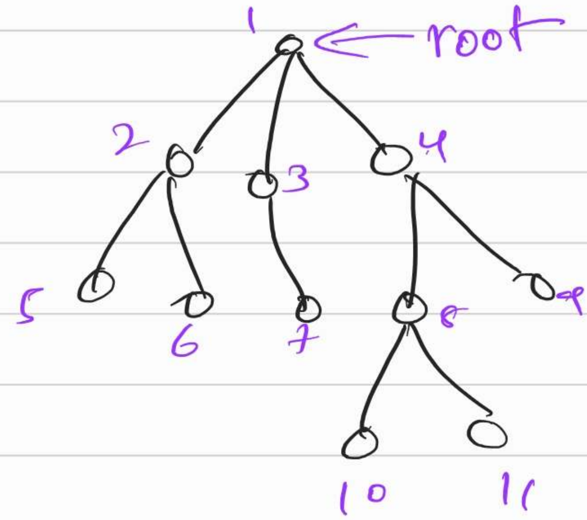


Not - connected.

Trees :- Connected graph that has no cycles in them.



Rooted trees :- There is a designated vertex that is called the root



9 is a child of 4.

4 is a parent of 9.

11 is a descendant of 4

4 is an ancestor of 11

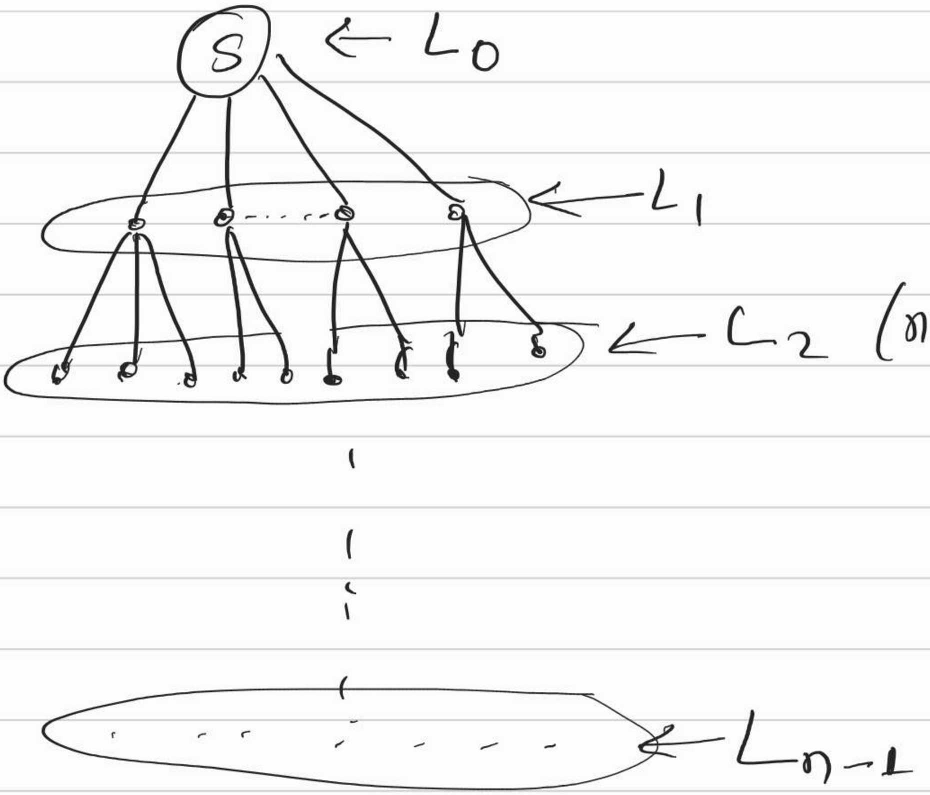
5, 6, 7, 10, 11, 9 are called leaves

Breadth-first Search (BFS).

$G = (V, E)$ given a node s.

Main Question:

Reachability: Given a graph G and two vertices s, t. Is there a path from s to t?



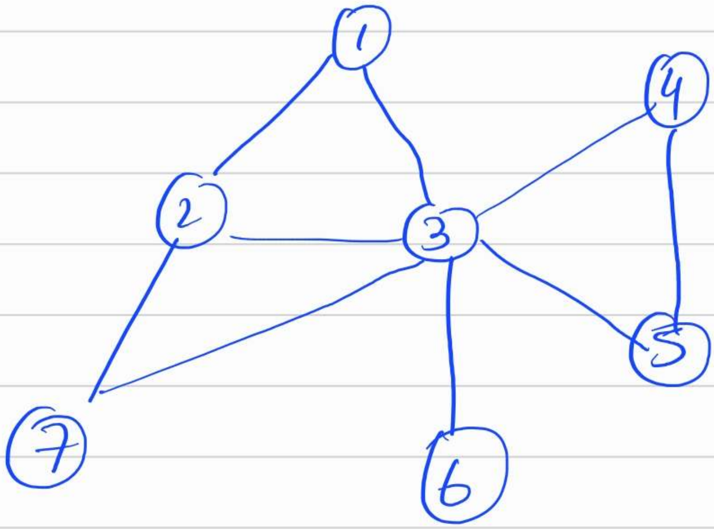
$|V| = n$

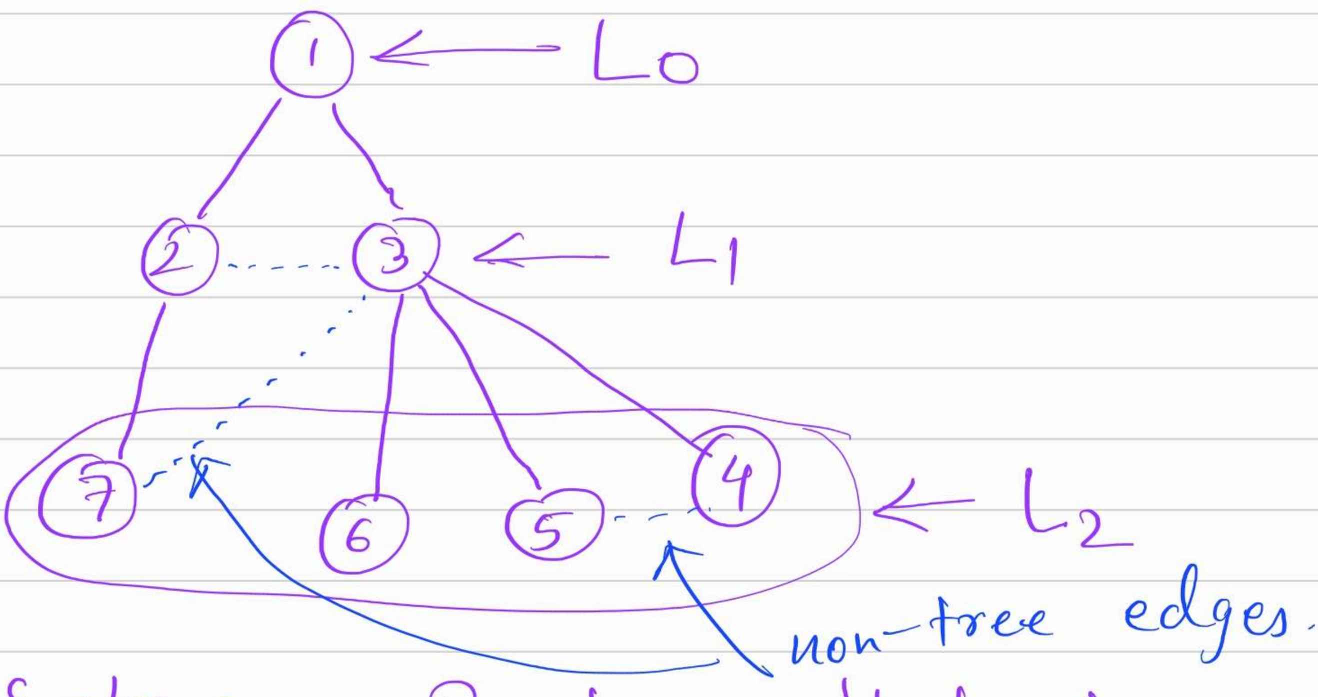
Q: Can the no. of layers be more than $n-1$? NO.

Q. How do you check t is reachable from s?
just check if t is in any of the layers.

Defn: distance between two vertices the length (# edges) of the shortest path between them.

Observation:- The shortest length path from s to vertices in layer i is of length i.





BFS-tree: The tree that is obtained by adding edges that is responsible for "discovery" of a vertex.

Observation:- Let T be a BFS-tree of G. Let (x, y) be an edge that does not belong to T.

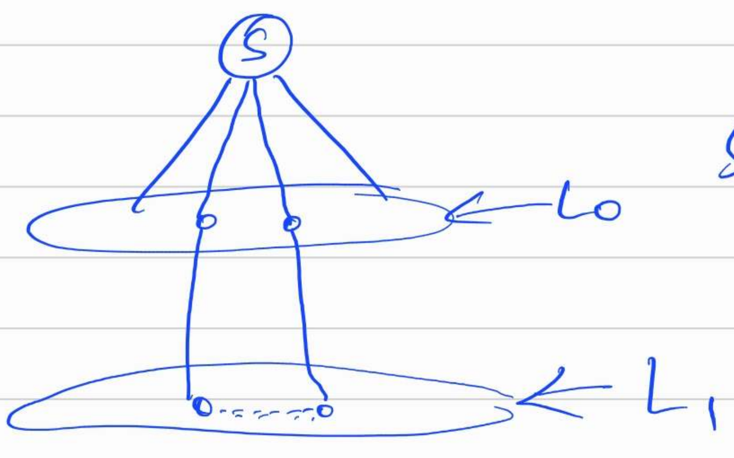
Let $x \in L_i$ and $y \in L_j$

then. $|i - j| \le 1$.

Q:- Suppose there are two vertices in the same layer and they are connected by an edge in the graph.

Is this graph a tree?

NO!



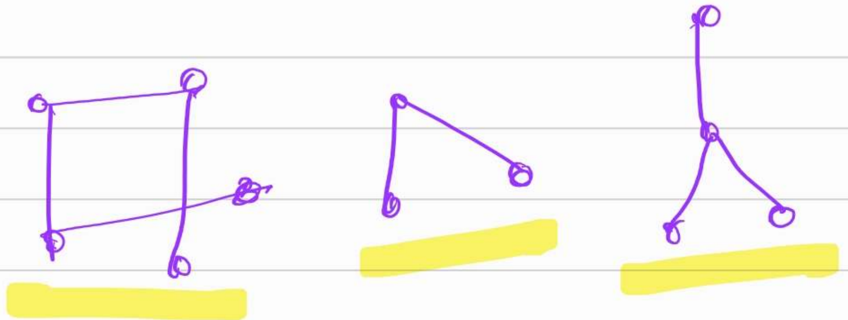
Such a graph will have a cycle.

what would be the length of such a cycle?

This will be at most $\leq 2i+1$

This length will be ODD!

Connected Components:



This graph has 3 components

Obs: The set of vertices obtained by doing a BFS starting at a vertex s is the connected component containing s.

Goal :- is to find the set of vertices reachable from the vertex s.

Initialise $R = \{s\}$

For each edge (u, v) if $u \in R$ and $v \notin R$ then

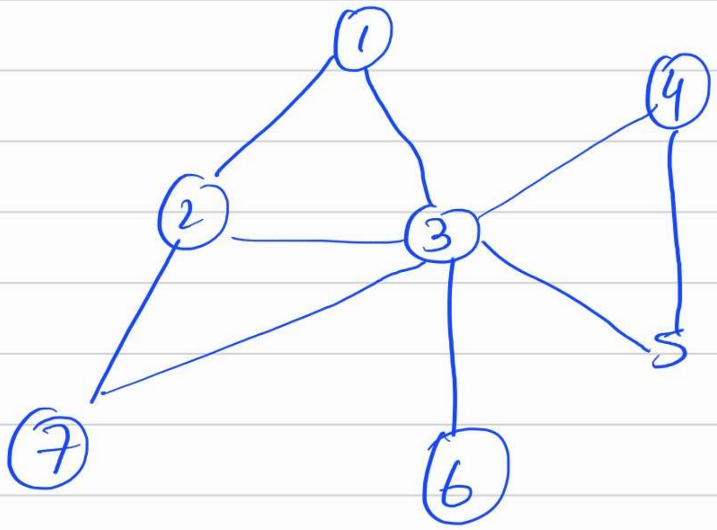
Add $v \in R$

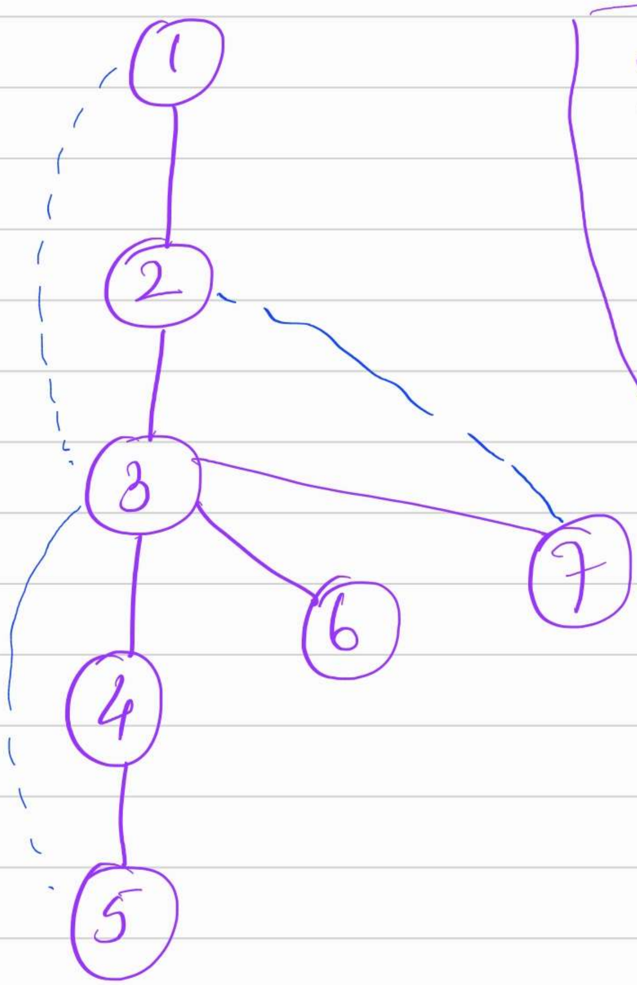
End For

Depending on the choice of choosing edges you get a different algorithm.

BFS :- our choice was to pick the edges incident on the last layer of vertices.

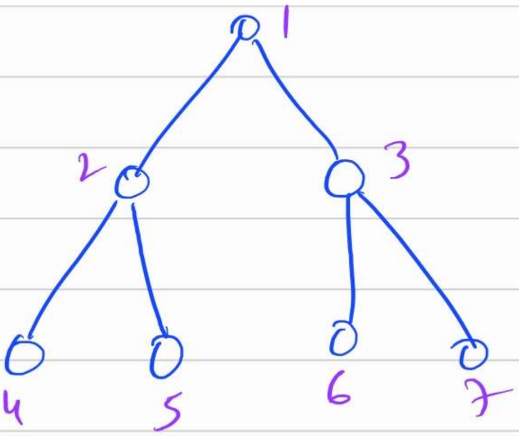
DFS: pick an edge on the last "seen" vertex.

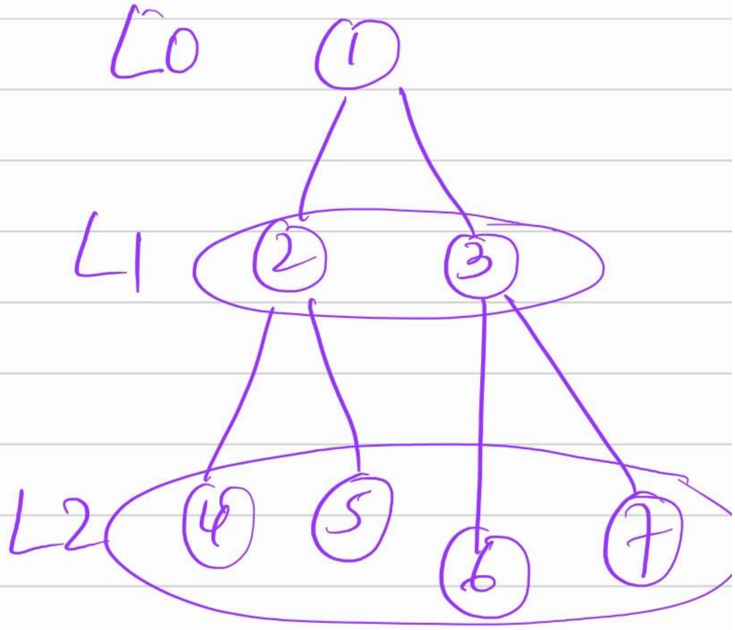




DFS(s)

Mark S as "seen"
For each edge (s, v)
  if v is not marked seen
    then mark v as "seen"
    Recursively call DFS(v)
  End if
End for.





$L(i)$ = { all nodes that are reachable from $L(i-1)$ and have not been seen before }