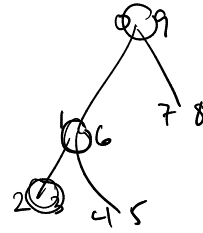


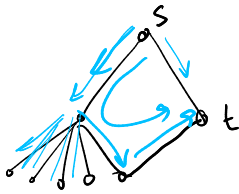
Last time: DFS

Also:

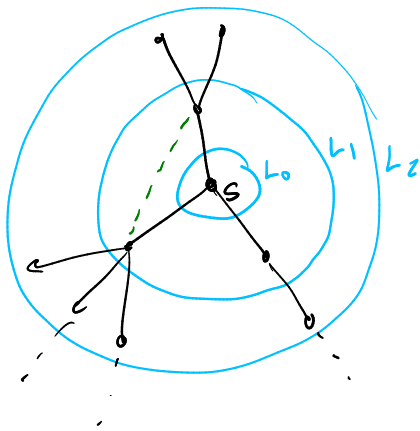
- Find articulation vertices (i.e. choke points)
- Find bridge edges
- Topological sorting
- Identify ancestors in a tree



DFS does not find shortest paths.



Breadth-First Search (BFS)



Properties:

- ① The shortest path $s \rightarrow v$ for any vertex v has length k iff $v \in L_k$.
- ② For any edge (u, v) , the layers of u and v differ by at most 1.

BFS (G, s):

Mark all vertices UNVISITED

$Q \leftarrow$ empty queue

Mark s VISITED

Add s to Q

[parent $\leftarrow$ empty dict
layer $\leftarrow$ empty dict
layer[s] $\leftarrow 0$]

While Q is not empty:

$u \leftarrow Q.$ dequeue

for each neighbor v of u :

if v is UNVISITED:

Mark v VISITED

$Q.$ enqueue(v)

[parent[v] $\leftarrow u$.

layer[v] $\leftarrow$ layer[u] + 1.

invariant: everything
in Q is VISITED -
+ has its layer filled in.

$\Theta(V + E)$ by same
argument as DFS.

Java: "double-ended
queue"
ArrayDeque
(implements Queue)

Python:
from collections
import deque.