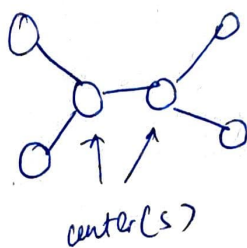
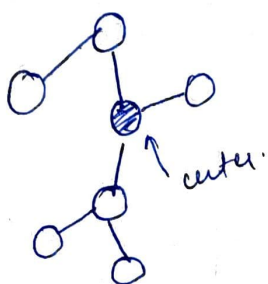


Tree center(s) 10.

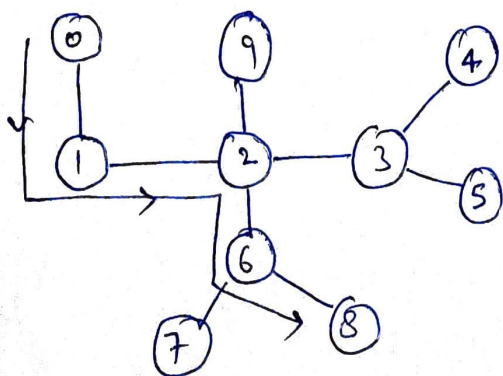
Center(s) of an undirected tree

An interesting and ~~many~~ often asked problem is finding a tree's center node(s). This could come in handy to select a good node to root our tree.

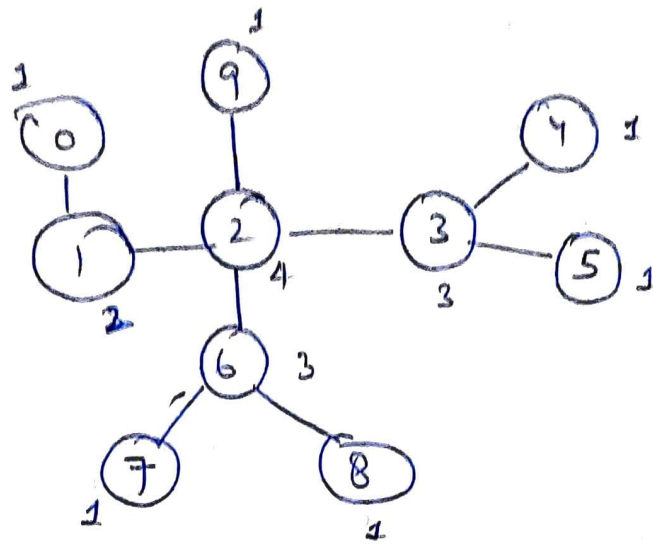


There can be more than one center but not more than 2.

Notice that this is because the center is always the middle vertex or middle two vertices in every longest path along the tree.



Another approach to find the center is to iteratively⁽²⁾ pick off each leaf node layer like we were peeling an onion.



Numbers outside represent degree of each node. Observe that each leaf node will have a degree of one.

As we prune nodes, also reduce the degree values. Repeat the process until we reach the root node.

g = tree represented as an undirected graph. (3)

function treeCenters(g):

$n = g.numberOfNodes()$

degree = [0, 0, ..., 0] # size n .

leaves = []

for ($i = 0$; $i < n$; $i++$):

degree [i] = $g[i].size()$

if degree [i] == 0 or degree [i] == 1 :

leaves.add(i)

degree [i] = 0

count = leaves.size()

while (count < n):

new_leaves = []

for (node : leaves):

for (neighbour : $g[node]$):

degree [neighbour] = degree [neighbour] - 1.

if degree [neighbour] == 1:

new_leaves.add(neighbour)

degree [node] = 0

count += new_leaves.size()

leaves = new_leaves.

return leaves # center(s)