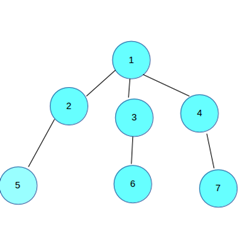
Check if two nodes are on same path in a tree

Given a tree (not necessarily a binary tree) and a number of queries such that every query takes two nodes of tree as parameters. For every query pair, find if two nodes are on the same path from root to the bottom.

For example, consider the below tree, if given queries are (1, 5), (1, 6) and (2, 6), then answers should be true, true and false respectively.  
[](http://d1gjlxt8vb0knt.cloudfront.net/wp-content/uploads/Check-if-two-nodes-are-on-same-path-in-a-tree.png)  
Note that 1 and 5 lie on same root to leaf path, so do 1 and 6, but 2 and 6 are not on same root to leaf path.

It is obvious that Depth First Search technique is to be used to solve above problem, the main problem is how to respond to multiple queries fast. Here our graph is a tree which may have any number of children. Now DFS in a tree if started from root node proceeds in a depth search manner i.e. Suppose root has three children and those children have only one child with them so if DFS is started then it first visits the first child of root node then will go deep to the child of that node. The situation with a small tree can be shown as follows:  
The order of visiting the nodes will be – 1 2 5 3 6 4 7 .

Thus other children nodes are visited later until completely one child is successfully visited till depth. To simplify this if we assume that we have a watch in our hand and we start walking from root in DFS manner.

Intime – When we visit the node for the first time

Outtime- If we again visit the node later but there is no children unvisited we call it outtime,

*Note: Any node in its sub-tree will always have intime < its children (or children of children) because it is always visited first before children (due to DFS) and will have outtime > all nodes in its sub-tree because before noting the outtime it waits for all of its children to be marked visited.*

For any two nodes u, v if they are in same path then,

Intime[v] < Intime[u] and Outtime[v] > Outtime[u]

OR

Intime[u] < Intime[v] and Outtime[u ]> Outtime[v]

* If given pair of nodes follows any of the two conditions, then they are on the same root to leaf path.
* Else not on same path (If two nodes are on different paths it means that no one is in subtree of each other).

**Pseudo Code**

We use a global variable time which will be incremented as dfs for a node begins and will also be incremented after

DFS(v)

increment timer

Intime[v] = timer

mark v as visited

for all u that are children of v

DFS(u)

increment timer

Outtime[v] = timer

end

Time Complexity – O(n) for preprocessing and O(1) per query.