Least Common Ancestor

While there are many ways to approach this problem, the easiest one is the following one.

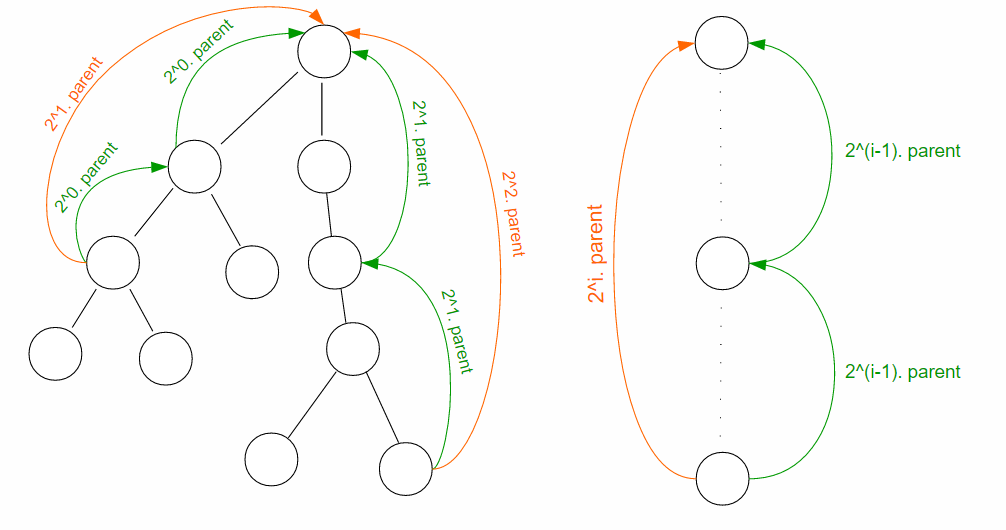
The idea is that we first preprocess the tree then query the results to get the lca in log (n) time.

1. Preprocessing

We first simply do a dfs to mark each node with its level. The following pseudocode should do so.

For the second part of the preprocessing, we store the parent for each node in powers of 2,ie, for each node, we store its 2^0, 2^1, 2^2, …. 2^k parents where k = log(n). Here, we are storing at max log(n) parents for every node. Hence the space it will take is O( n \* log(n)). For n = 1000000, the number of parents to store for each node will be at max 20.

The following diagram shows the relationship between a node and its parent nodes.



The current node’s ith parent will be its parent’s i – 1 th parent, ie,

parent[i][node] = parent[ i - 1][parent[i - 1][node]]

This defines the recurrence relation used to fill the table. It can be done in the following manner.



1. The Algo

First we need to ensure that the two nodes are at the same level. For that, we use the following code.

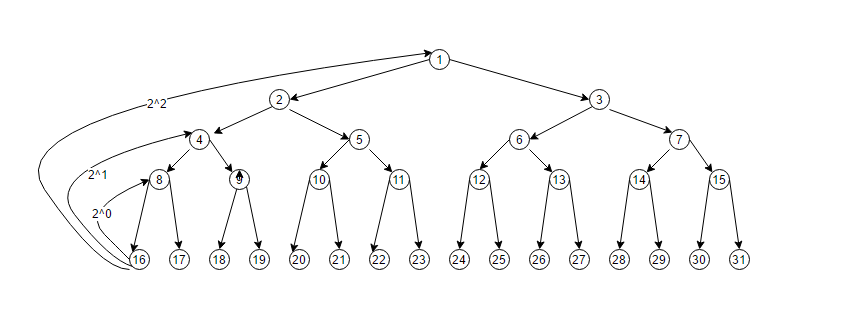
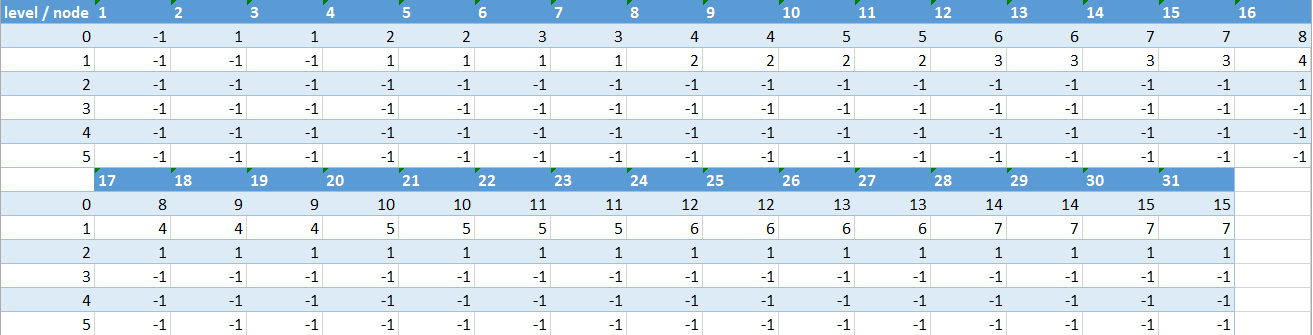


Now that u and v are at the same level, we can start jumping upwards until we find the least common ancestor. This can be done using



Lets take an example.

The figure shows respective parents for node 16. The table for all the nodes will be filled as follows.



pa[2][16] = pa[1][pa[1][16]] = pa[1][4] = 1.

That means node 16’s 2^2 parent is 1 which is shown in fig. You can further use the codes along with the table to check the working of the algorithm.