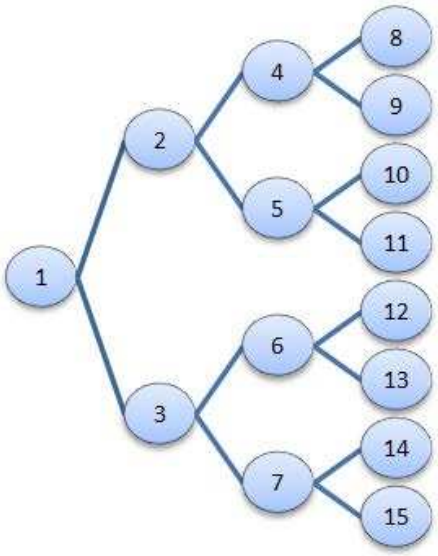
UCF "Practice" Local Contest — Aug 27, 2016

Lowest Common Ancestor

filename: `lca`
(*Difficulty Level:* `Hard`)

Perfect binary trees are one of the coolest structures that computer scientists study. They have a lot of properties that make them very nice to work with. One of the nicest properties is that they can just be described by a single integer *n* giving the depth of the tree. For instance, the perfect binary tree for $n = 3$ looks like:



In general, a perfect binary tree with depth *n* will have exactly $2^{n+1} - 1$ nodes, and can be numbered by following the pattern seen above (there are of course other ways to number the nodes of the tree, but this is the scheme we will use in this problem).

A common question that arises when dealing with trees is the query of the lowest common ancestor (commonly called LCA) of two nodes in the tree. Formally, the LCA of *x* and *y* is the node *z* of greatest depth in the tree such that *z* is an ancestor of *x* and *y*. Node *a* is an ancestor of node *c* if *c* exists in the sub-tree rooted at node *a*. Notice that 1 is trivially a common ancestor of any two nodes in the tree, but is not always the *lowest* common ancestor. For instance, the common ancestors of nodes 7 and 12 are 1 and 3, and 3 is the LCA since it is the node of greatest depth. The LCA of 2 and 13 is node 1, and the LCA of 5 and 11 is node 5. The definition of LCA guarantees that the LCA of any two nodes will always be unique.

The Problem:

Given two nodes in the tree using the numbering scheme shown above, determine the LCA of the two nodes.

The Input:

Input will begin with a positive integer, $T \leq 2 \cdot 10^6$, indicating the number of test cases. This will be followed by T test cases, each on a separate input line. Each test case will contain two space separated integers, X and Y, represented in hexadecimal. X and Y will each contain at most 1000 characters from the set {0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f}, where a-f represent 10-15, respectively. You are to determine the LCA of X and Y.
Note: The hexadecimal (base 16) number $d_n d_{n-1} \cdots d_1 d_0$ is converted to a decimal number (base 10) by the following formula: $d_0 \cdot 16^0 + d_1 \cdot 16^1 + \cdots + d_{n-1} \cdot 16^{n-1} + d_n \cdot 16^n$.

The Output:

For each case, output a single line:

`Case #x: y`

where x is the case number beginning with 1, and y is the LCA in hexadecimal with no leading 0's. Leave a blank line after the output for each test case.

Sample Input:

```
7
7 c
2 d
b 5
10 11
a020fac a030ccf
12afcdb 12afcdc
100000000 ffffffff
```

Sample Output:

```
Case #1: 3

Case #2: 1

Case #3: 5

Case #4: 8

Case #5: 501

Case #6: 255f9b

Case #7: 1
```