Topic 5: Tree Review

LCA. HLD

Tree

We already know a lot about trees

- DFS
- BP
- etc.

Sample Problem: Diameter of the tree

This lecture adds one perspective

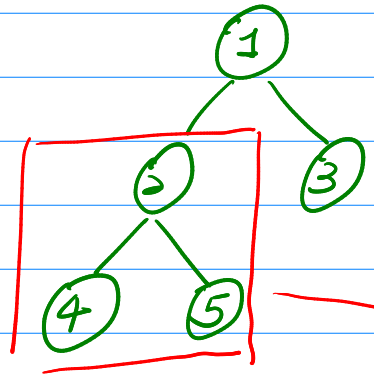
An array is simpler than a tree

So we turn a tree into an array

and do range queries ...

- subtree: DFS order
- path: LCA & HLD

1975 Order



1 2 4 5 3

Any subtree is a continous subarray

Query on a Tree

Query on a Tree

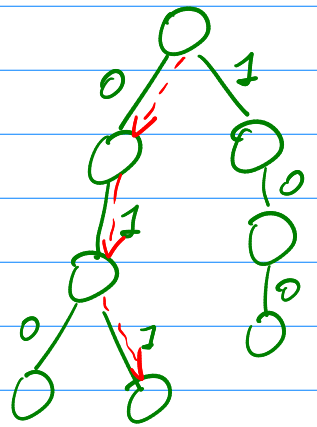
Static version:

$(a_1, a_2, \dots, a_n)$ v

find $\max_i (a_i \oplus v)$

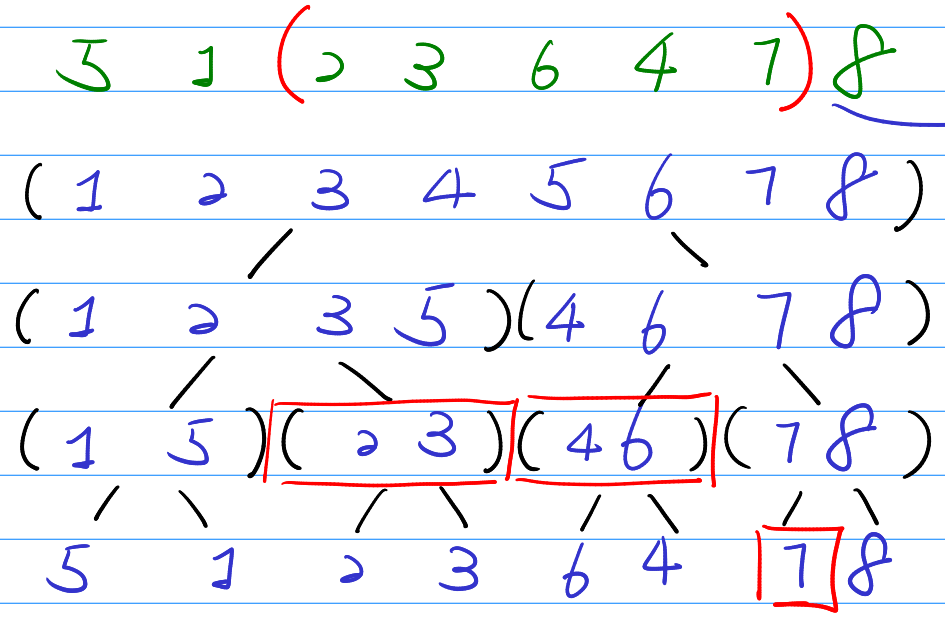
Trie

$(010)_2$ $(100)_2$ $(011)_2$



$v = (110)_2$

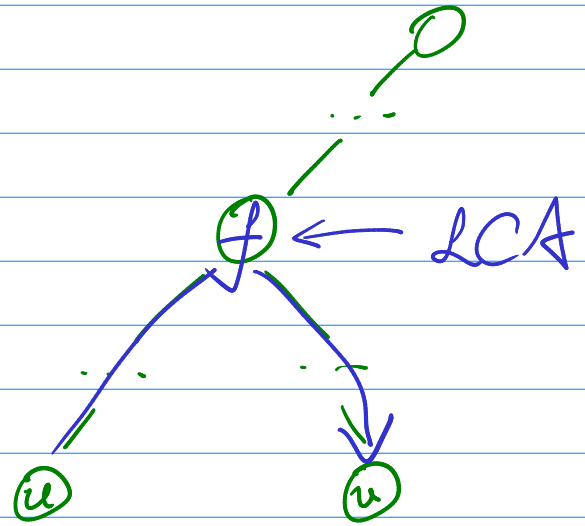
Query on a Tree



Each range is $O(\log n)$ nodes

Query on one node is $O(\log n)$

LCA (Least Common Ancestor)



Every path (u, v) on the tree is an up-path (u, f) and a down-path (f, v)

Complexity: Preprocess $O(n \log n)$
Query $O(\log n)$

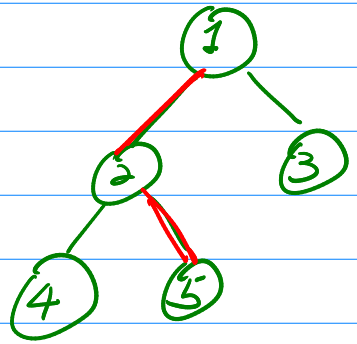
Maze Designer

HLD (Heavy-Light Decomposition)

We know DFS order turns subtrees into ranges

Can we turn paths into ranges?

We can if we DFS carefully

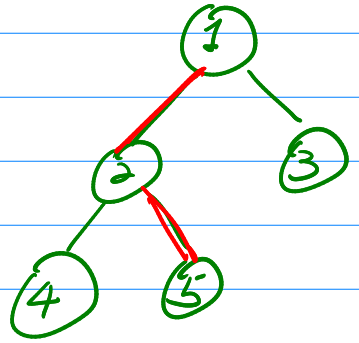


(1 2) 4 (5) 3

a couple ranges

Goal: Make it so that there are only $O(\log n)$ ranges per path.

HLD (Heavy-Light Decomposition)



(1 2) 4 (5) 3

a couple ranges

Goal: Make it so that there are only $O(\log n)$ ranges per path.

Method: When we DFS, we go down the largest subtree first.

Why? When we enter a new range, we are not going down the largest subtree which means the remaining tree size $< \frac{1}{2}$ current size.

Very Important Edge

Very Important Edge

(Code on the webpage)