Heavy-Light Decomposition

Oliver-Matis Lill

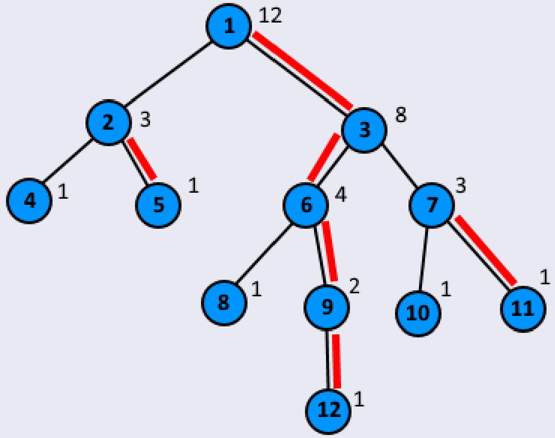
June 16, 2017

The Problem

- Suppose you have tree with $N \leq 10^5$ vertices, where each vertex i has some value v_i
- You are given $Q \leq 10^5$ queries of the form:
  1. Give vertex i a new value x
  2. Output the sum of values on the path from vertex i to j
- How would you approach this?

- For each vertex i let's denote the number of vertices in its subtree as w_i
- Next for each vertex i pick among its children a vertex j with the greatest w_j. Connect them with a "link"
- The links will form a collection of paths and this division into paths is called "Heavy-Light Decomposition"

Heavy-Light Decomposition Example



Heavy-Light Decomposition Applications

- In real problems we usually build segment trees on those paths to facilitate various queries on tree paths
- Note that if we traverse up to root or ancestor, each time we swap paths, the weight of the current vertex more than doubles
- This means we can swap paths at most $O(\log N)$ times
- With segment trees this gives us an $O(\log^2 N)$ algorithm for various complicated queries on tree paths
- This method is not very common and usually you can use sqrt-decomposition instead

Heavy-Light Decomposition Code

```
int HLDtraversal(Node* cur, Node* anc) {
    SegmentTree* pathTree = cur->pathTree;
    int result = 0;

    while(1) {
        if(anc->pathTree == pathTree) {
            result += pathTree->sum(anc->depth, cur->depth);
            return result;
        }

        result += pathTree->sum(0, cur->depth);
        cur = pathTree->top->parent;
        pathTree = cur->pathTree;
    }
}
```