Heavy-Light Decomposition Notes

Quanquan Liu

February 11, 2015

1 Heavy Light Decomposition

- Used in analysis of link-cut trees. Can make certain operations involving edge weights faster on unbalanced trees.
- One important question: what is the distance between two nodes in a tree?

2 LCA Algorithm on Trees

- Find the lowest common ancestor between two nodes in the tree by moving two pointers up the tree prioritizing the pointer at a greater distance from the root. Preprocessing time: $O(n)$ using DFS from root.
- Algorithm performs well on trees with low depth. But the runtime of this algorithm is $O(n)$ for trees that are just paths or chains.
- Can keep track of the distance as you perform the LCA Algorithm. Or $d(v, w) = d(v, root) + d(w, root) - 2d(LCA(v, w), root)$.

3 Computing Distance Between Two Nodes on Chains

- Static case: Compute the distance from root: $d(v, w) = |d(v, root) - d(w, root)|$. Preprocessing time: $O(n)$. Query time: $O(1)$.
- Dynamic case: Use a balanced binary search tree (segment tree) to keep track of distances between nodes. (Probably will not go over in detail during the lecture.)

4 Heavy-Light Decomposition

Important insight: can make a tree of disjoint chains so that we take advantage of both runtimes given above. The tree of chains will have low depth and the chains can be used efficiently.

4.1 Definitions

- Given a rooted tree, define $size(v)$ to be the number of nodes in the subtree rooted at node v (the size includes the node v).
- We define any edge as $(v, parent(v))$ where $parent(v)$ is v's parent in the tree. Therefore, $parent(v)$ refers to the endpoint that is closer to the root and v represents the endpoint that is farther from the root.
- An edge between two nodes of the tree is defined as ***heavy*** iff $size(v) > \frac{1}{2}size(parent(v))$.
- An edge is defined as ***light*** otherwise (i.e. $size(v) \leq \frac{1}{2}size(parent(v))$.

4.2 Important Characteristics

- Any node has **at most one** heavy child (i.e. child linked by a heavy edge).
- Any node may be joined by at most 2 heavy edges.
- The heavy edges **decompose the tree nodes** into disjoint paths/chains.
- Chains are connected to each other via light edges.
- We cross at most $O(\log n)$ heavy chains on any path from the root to any node in the tree.
- A ***path tree*** contains the paths formed by the heavy-light decomposition paths where all nodes in the path are contracted into one node.

5 Finding Distance Between Two Nodes in Any Rooted Tree in Polylog Time

Given a rooted (static) tree, find the minimum distance between any two nodes a and b, $d(a, b)$.

(To be discussed in lecture.)