Lecture 9: Lowest Common Ancestors

Johannes Fischer
Lowest Common Ancestors
Some Initial Thoughts

• store only tree:
  \[ O(n) \text{ w.c. query time} \]

• store all \( \Theta(n^2) \) answers:
  \[ O(1) \text{ query time} \]

• difficulty:
  - \( O(1) \text{ query time with } O(n) \text{ space} \)
  - lecture "Text Indexing" (SS'12)
  - here: \textbf{distributed} data structure
Distributed Data Structures

• no access to **global** data structures
  → minimize **communication overhead**

• labeling scheme:
  ▶ assign label $l(v)$ to each node $v$
  ▶ compute $l($LCA$(x,y))$ from $l(x)$ and $l(y)$

• goal:
  ▶ short labels
  ▶ fast query time
Simple Tree Labelings: parent(x,y)

parent(x,y) iff first lg n bits of l(x) = 2nd lg n bits of l(y)
Simple Tree Labelings: LCA(x,y)

\[ l(LCA(x,y)) \approx LCP(l(x), l(y)) \]

\[ l(v) = l(parent(v)) \cdot DFS(v) \]
Simple LCA-Labeling

• longest label length:
  ▶ between $O(\lg^2 n)$ and $O(n \lg n)$ bits
  ➞ cannot even compute LCP in $O(1)$ time

• in the following:
  ▶ label length $O(\lg n)$ bits
  ▶ $O(1)$ query time
Definitions

• node $v$:
  - $p(v)$ = parent of $v$
  - $c(v)$ = set of $v$'s children
  - $size(v)$ = #nodes in $v$'s subtree $T_v$

• heavy nodes:
  - having largest subtree among its siblings
  - $u$ heavy if $size(u) = \max\{size(w) : w \in c(p(u))\}$
  - take arbitrary child if max not unique

• all other nodes: \textbf{light} (incl. root)
Heavy Paths

• heavy nodes divide $T$ into **heavy paths**:
  ▶ from light node follow heavy nodes
  ▶ continue recursively
  ▶ **heavy path decomposition**

• $\langle v_1, v_2, ..., v_k \rangle$ heavy path
  ▶ $v_1 = a(v_i)$ is the **apex** of $v_i$ for all $i$

• **light size** of $v$:
  • $lsize(v) = size(v) - size(w)$ if $w$ is $v$'s heavy child
Labels

• **heavy label** \( hl(v) \)
  - to any node \( v \)
  - different for two nodes on one heavy path
  - can determine if \( i < j \) from \( v_i, v_j \) on \( \langle v_1, v_2, ..., v_k \rangle \)

• **light label** \( ll(v) \): 
  - only to light nodes \( v \)
  - different for nodes with same parent

• **label** \( l(v) = l(p(a(v))) \cdot ll(a(v)) \cdot hl(v) \)
Answering LCA($x, y$)

- compute LCP of $l(x)$ and $l(y)$
- 2 cases
  - depending on whether mismatch occurs in $hl$ or $ll$
  - need helper label ($0 \triangleq hl, 1 \triangleq ll$)
- see blackboard
Analysis: Idea

• $hl(v)$ repeated in all nodes below $v$ apart from those below heavy child

  $\implies hl(v)$ occurs $lsize(v)$ times

  $\sim$ use **shorter** heavy labels for **large** *lsizes*

• $ll(v)$ occurs in all nodes below $v$

  $\implies ll(v)$ occurs $size(v)$ times

  $\sim$ use **shorter** light labels for **large** *subtrees*
Precise Analysis

- see blackboard