Labeling Schemes for Small Distances in Trees

Stephen Alstrup

Philip Bille (speaker)

Theis Rauhe

The IT-University of Copenhagen

An example problem

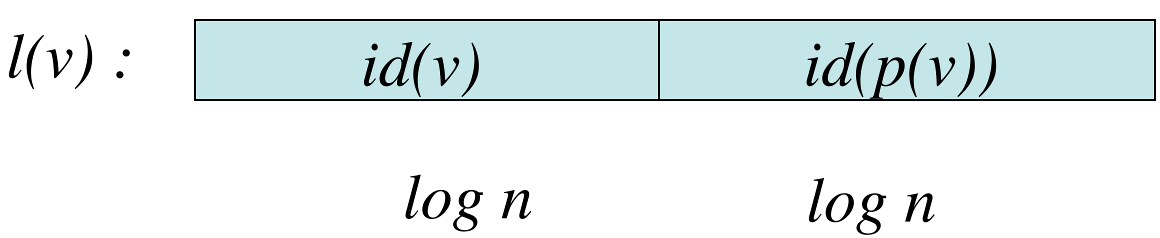
- **Given:** A rooted tree T with n nodes.
- **Task:** Assign a label, $l(v)$, to each node v such that for any pair of nodes v and w we can determine from $l(v)$ and $l(w)$ alone if:
  - w is the parent of v
  - v is the parent of w
  - v and w are siblings
- **Goal:** Minimize the maximum length of labels.

A simple $2\lceil \log n \rceil$ solution

- Assign a unique identifier, $id(v) \in \{1,\ldots,n\}$, to each node v.

A simple $2\lceil \log n \rceil$ solution

- $w = p(v)$ iff $id(w) = id(p(v))$.
- $v = p(w)$ iff $id(v) = id(p(w))$.
- $p(v) = p(w)$ iff $id(p(v)) = id(p(w))$.



Can we do better?

Previous upper bound: $\log n + O(\sqrt{\log n})$ [Kaplan and Milo '01]

Theorem There is a labeling scheme for trees supporting parent and sibling queries with labels of maximum length $\log n + O(\log\log n)$.

Theorem Any labeling scheme for trees supporting parent and sibling queries must use labels of length at least $\log n + \Omega(\log\log n)$.

Applications

- XML search engines
- Routing schemes

  see e.g. [Abiteboul, Kaplan and Milo '01], [Thorup and Zwick '01]

Related work

- Adjacency labeling schemes and implicit graph representation.

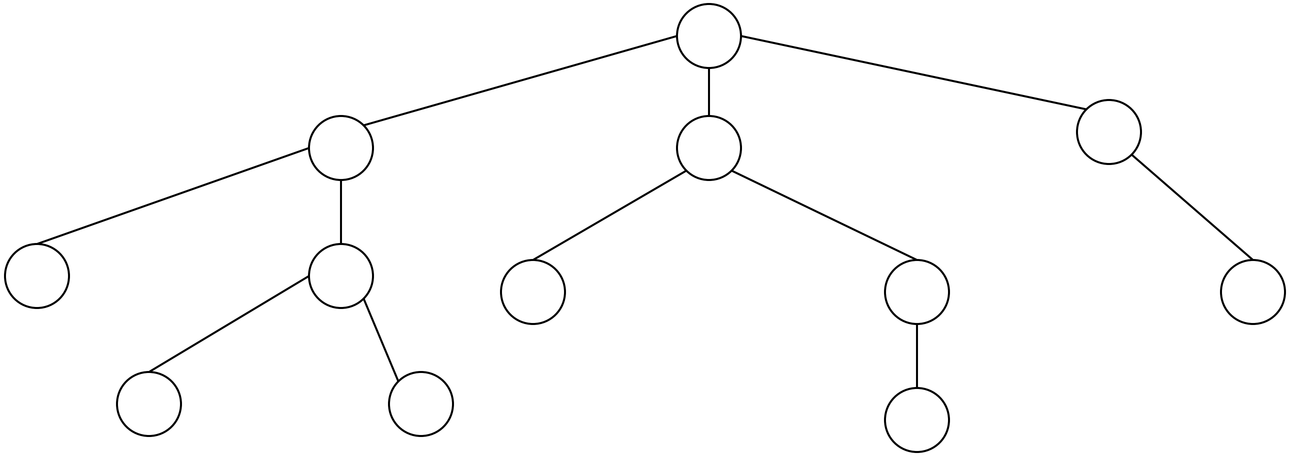
  e.g. [Kannan, Naor and Rudich '88]
- Distance labeling schemes.

  e.g. [Gavoille, Peleg, Perennes, Raz '01]

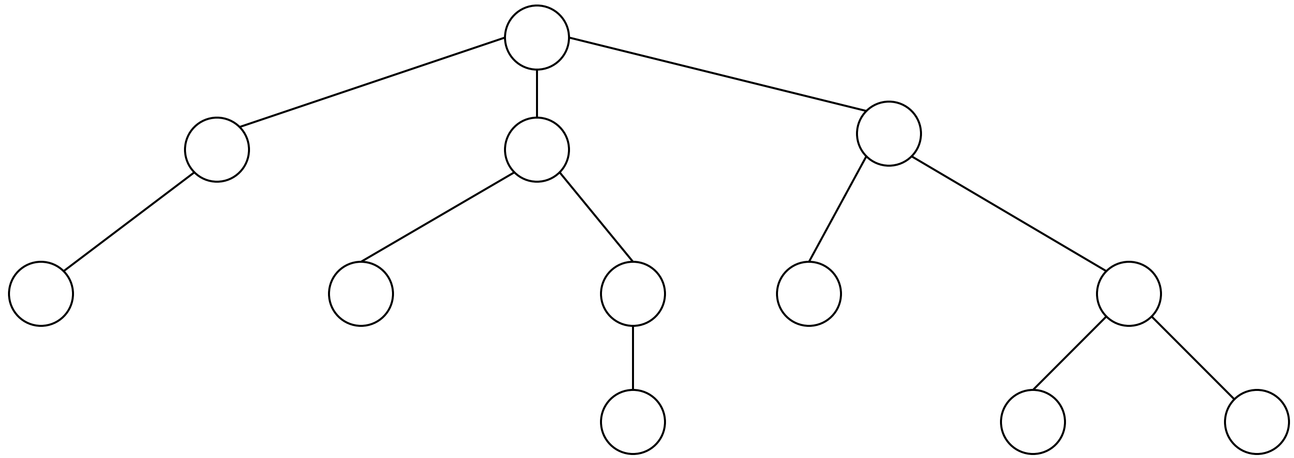
Related work

- Ancestor labeling schemes.
  e.g. [Abiteboul, Kaplan and Milo '01]
- Flow and connectivity labeling schemes.
  [Katz, Katz, Korman and Peleg '02]

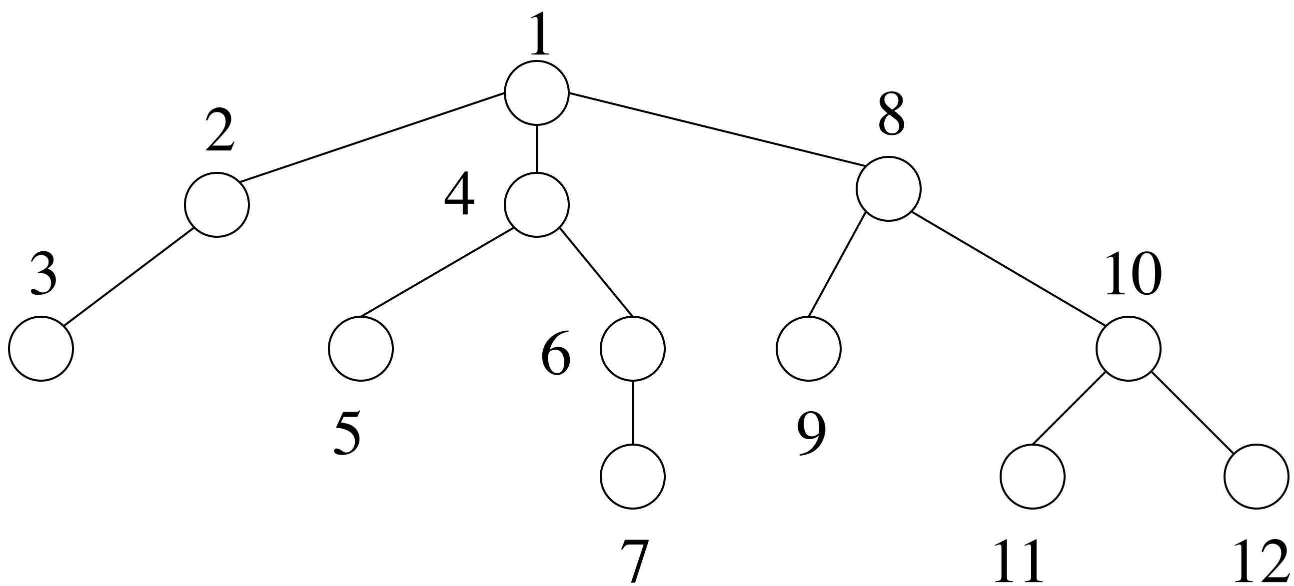
A tree



Order according to subtrees size



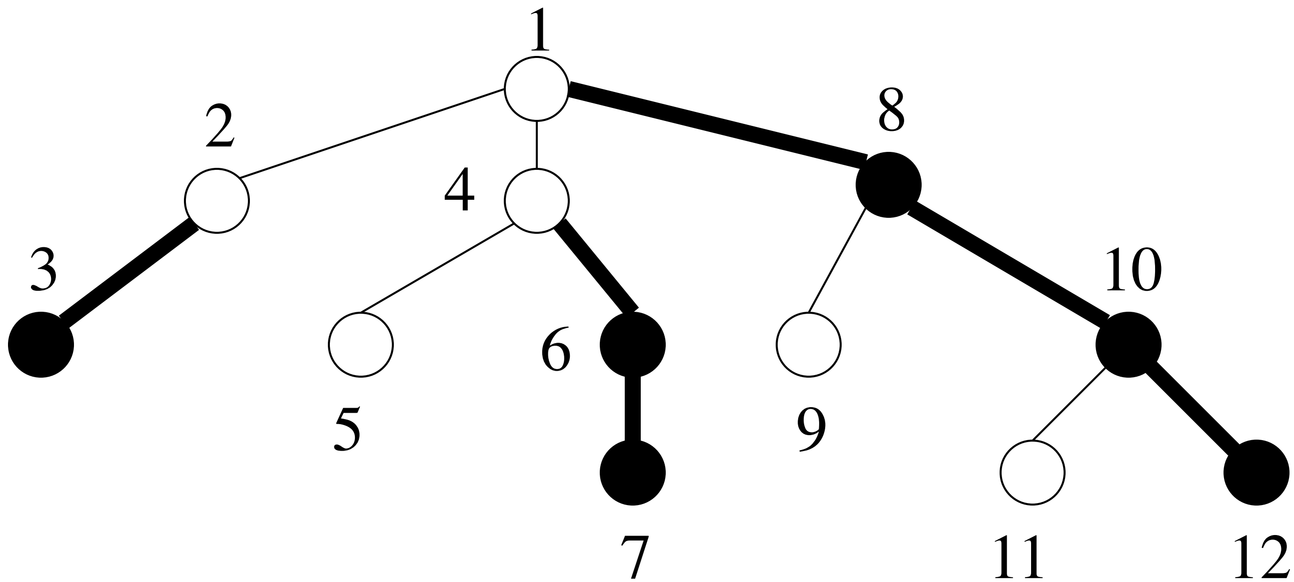
Preorder numbers



pre(v)

log n

Heavy-path decomposition



pre(v)

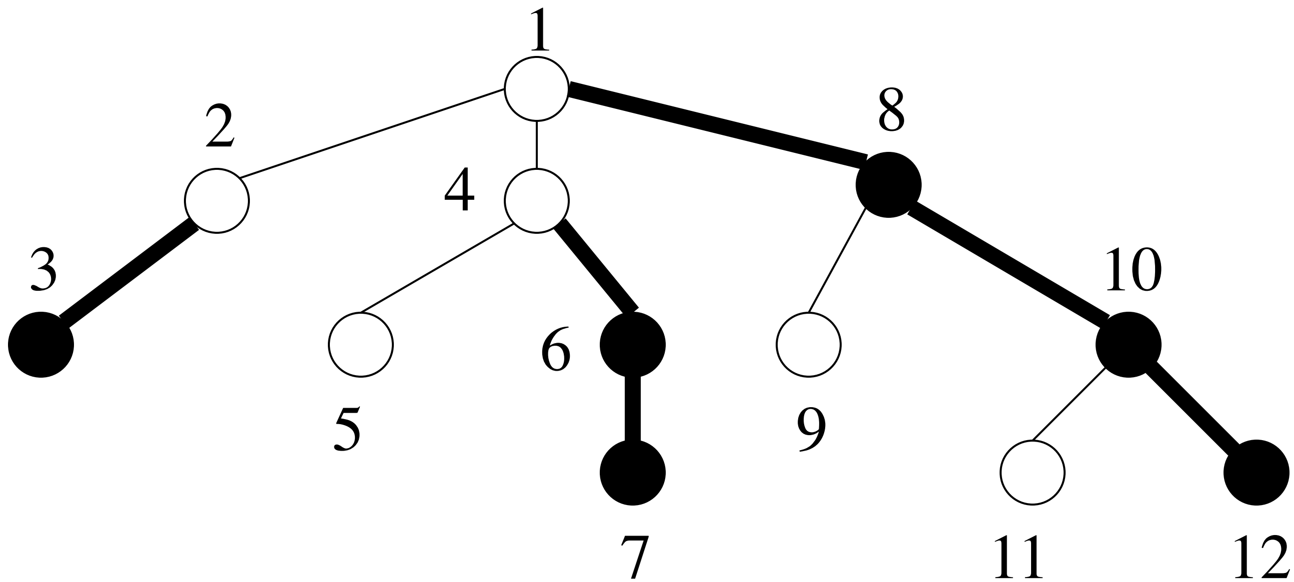
log n

Light depth

- *ldepth(v)* = number of light edges on path from *v* to the root.

Lemma [Harel and Tarjan '84] For any node v, $ldepth(v) \leq \log n + O(1)$

Heavy-path decomposition



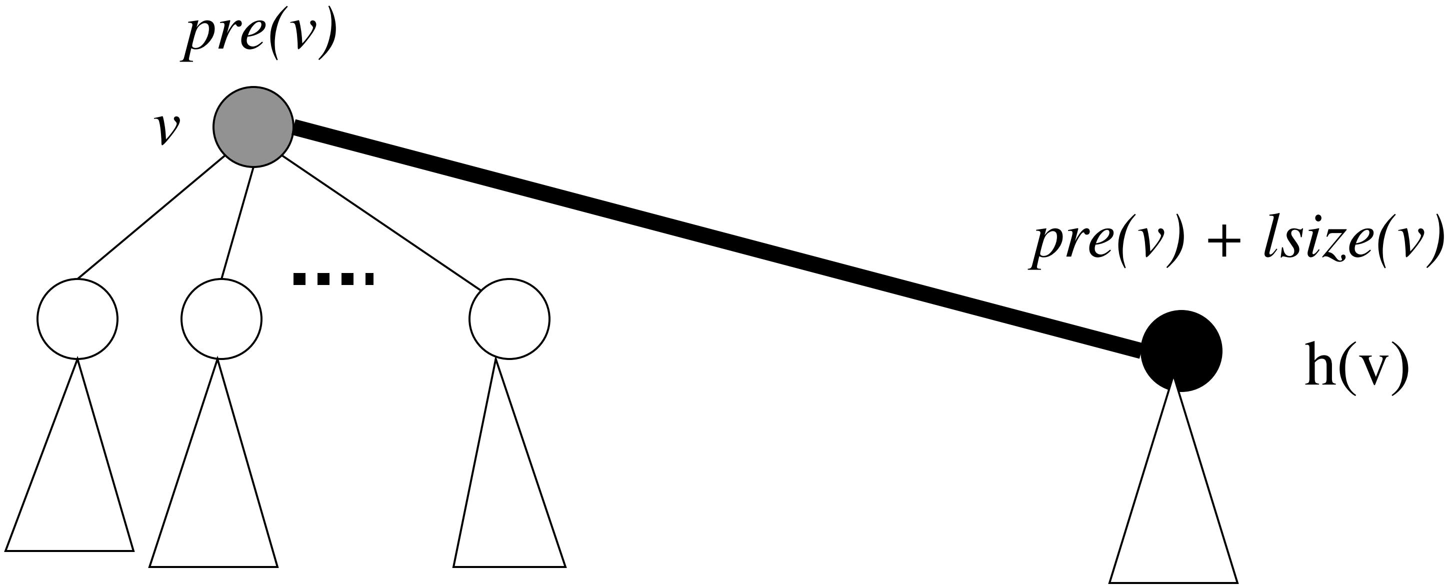
pre(v)	*ldepth(v)*

log n   *log log n*

Significant preorder numbers

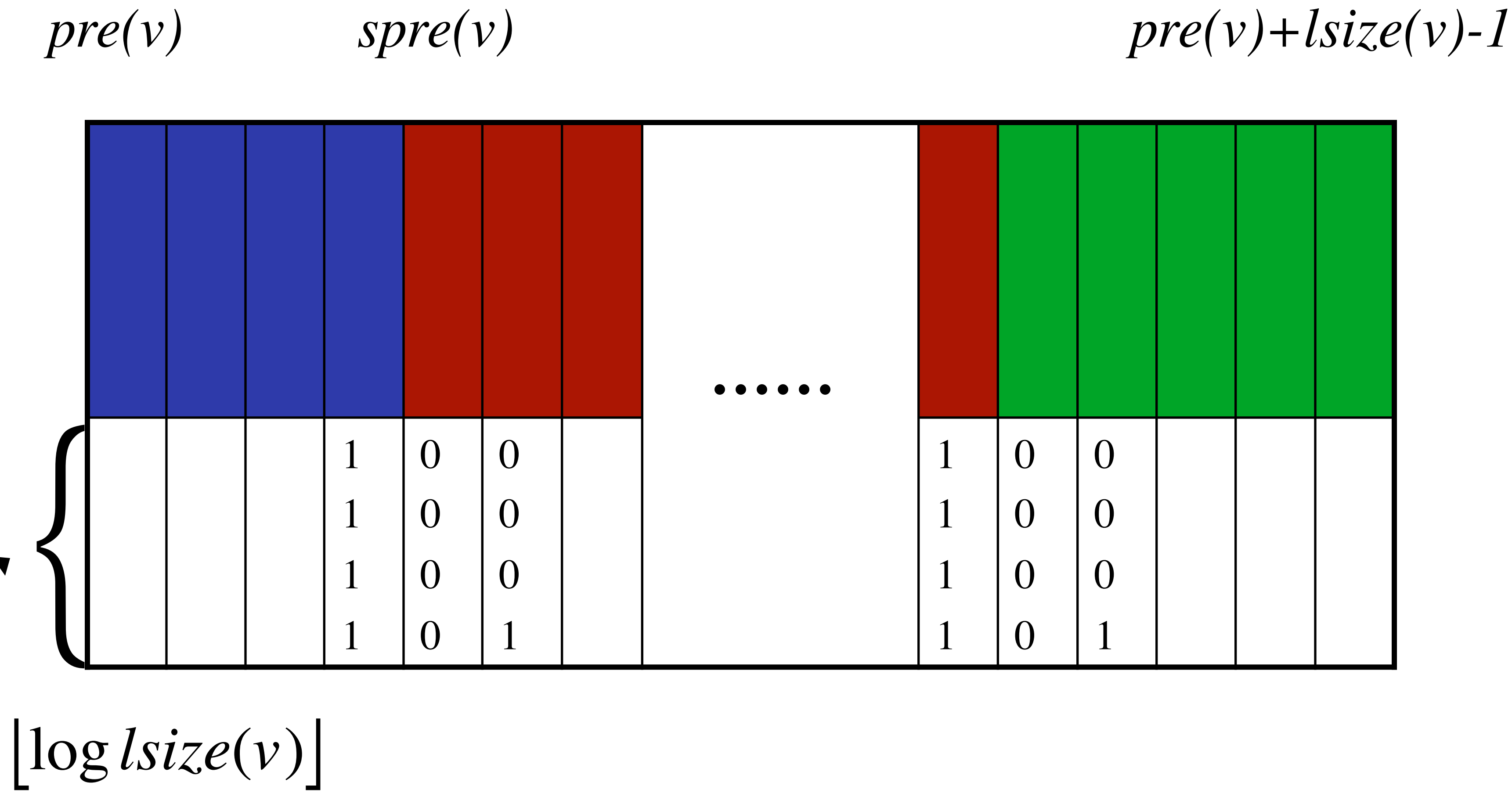
- Assign a number *spre(v)* to each node *v* such that
  - *spre(v)* can be used to uniquely identify *v*.
  - *spre(v)* can be efficiently represented in the label of *v* and all light children of *v*.

Light size



lsize(v) = size(v) - size(h(v))

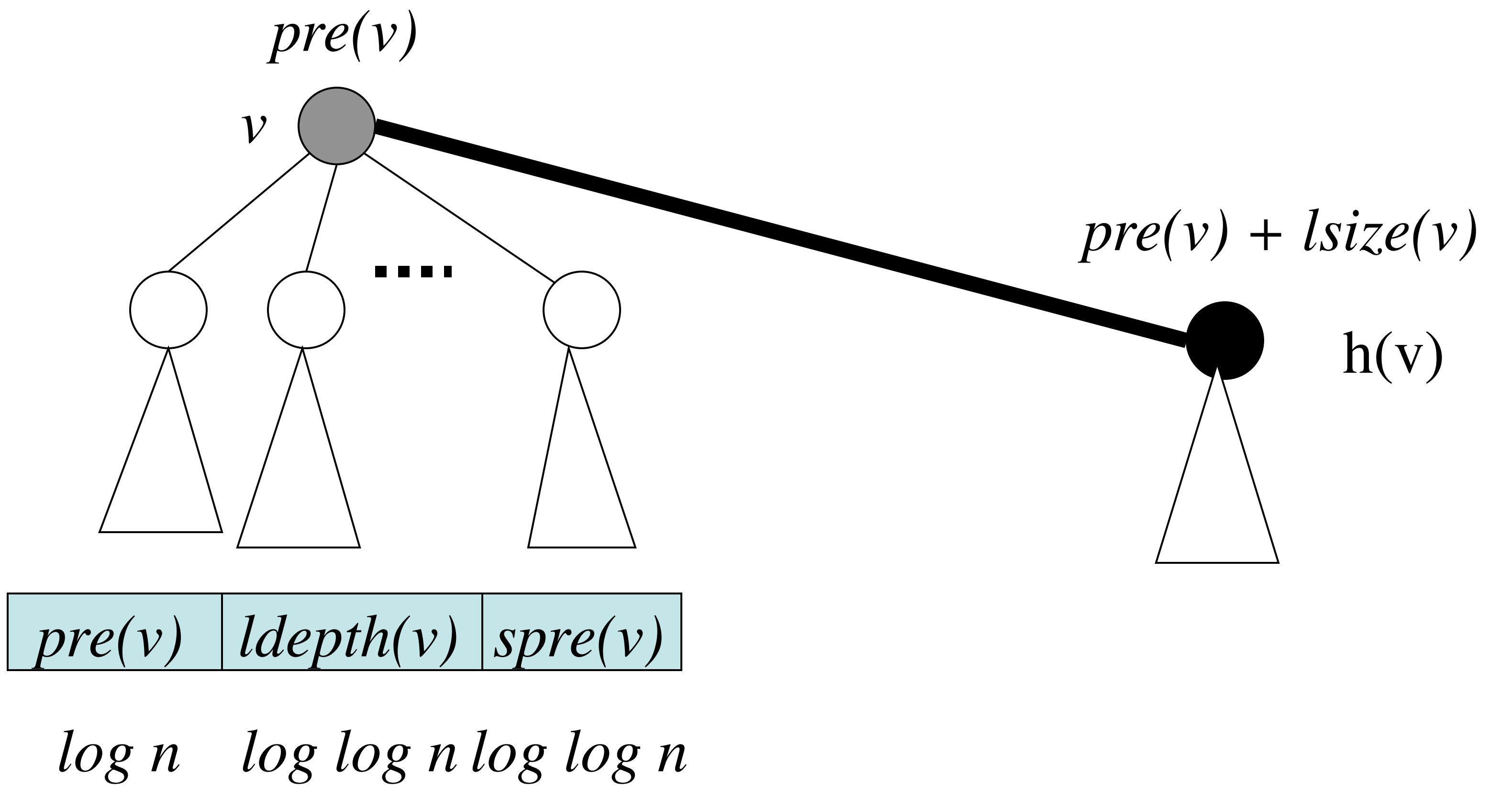
Significant preorder numbers



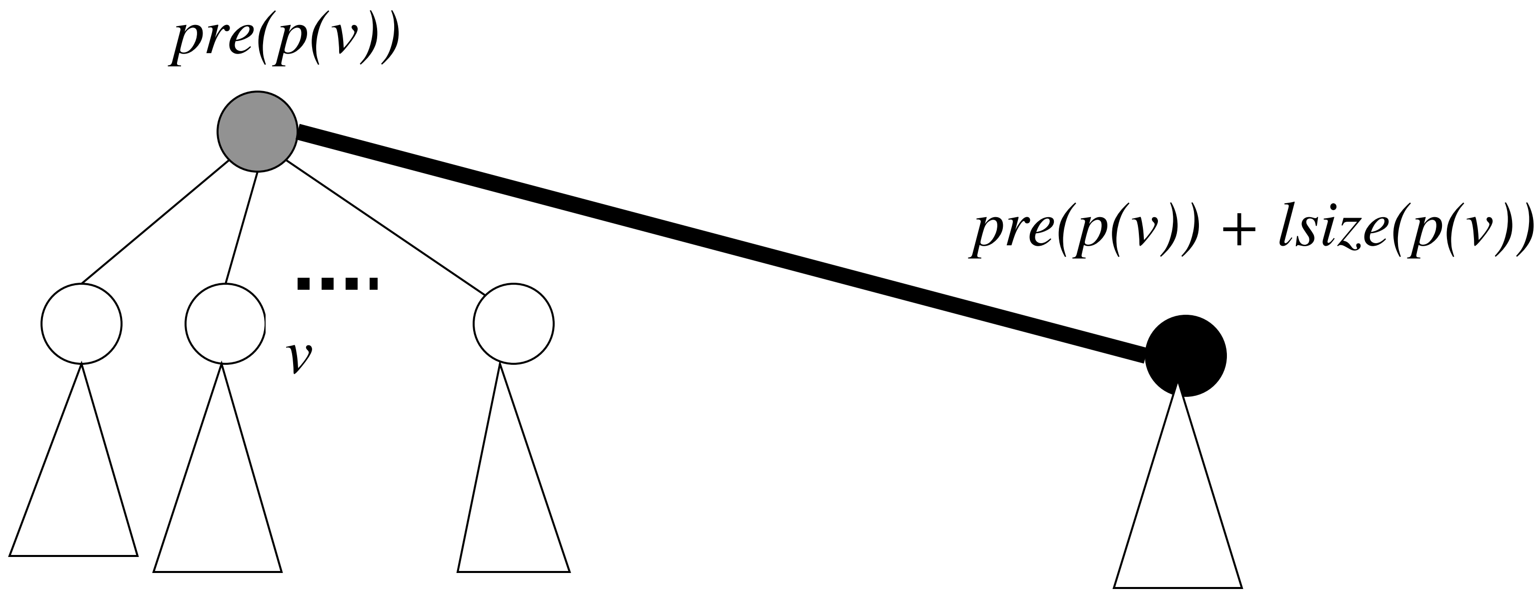
Significant preorder numbers

- $v = w$ iff $spre(v) = spre(w)$ and $ldepth(v) = ldepth(w)$.
- Given $pre(v)$ and $\lfloor \log lsize(v) \rfloor$ we can compute $spre(v)$.

Significant preorder numbers



Case 1: *v* is light



pre(v)	*ldepth(v)*	*spre(v)*	*spre(p(v))*
log n	*log log n*	*log log n*	*log log n*

Answering queries

Lemma For two light nodes v and w, v and w are siblings iff $ldepth(v) = ldepth(w)$ and $spre(p(v)) = spre(p(w))$.

Lemma For a light node v and node w, w is the parent of v iff $ldepth(v) = ldepth(w) + 1$ and $spre(p(v)) = spre(w)$.

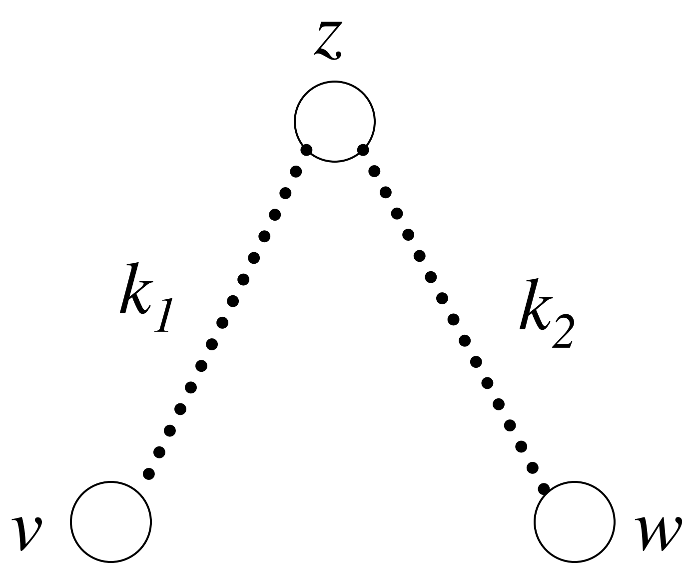
Case 2: v is heavy

- In the paper!

Theorem There is a labeling scheme for trees supporting parent and sibling queries with labels of maximum length $\log n + O(\log\log n)$.

General relationships

- Two nodes v,w with z = nca(v,w) are (k_1,k_2)-related if $dist(v,z) = k_1$ and $dist(w,z) = k_2$.



Example

If *v* and *w* are:

- (0,1)-related => *v* is the parent *w*
- (1,0)-related => *w* is the parent of *v*
- (1,1)-related => *v* and *w* are siblings.

k-relationship labeling scheme

- A k-relationship labeling scheme supports tests for whether two nodes are (k_1,k_2)-related for $k_1,k_2 \leq k$
- Ex.: A 1-relationship labeling scheme supports parent and sibling queries.

Results

Theorem There is a k-relationship labeling scheme using labels of length at most $\log n + O(k^2(\log\log n + \log k))$

Theorem For constant $k \geq 1$, any k-relationship labeling scheme must use labels of length $\log n + \Omega(\log\log n)$.

k-restricted distance labeling scheme

- With a *k*-restricted distance labeling scheme we can decide if two nodes are at distance at most *k* and if so compute the distance.
- Same bounds as before.

k-relationship (const. k)	trees	$\log n + \Theta(\log\log n)$
k-rest. dist. (const. k)	trees	$\log n + \Theta(\log\log n)$
biconnectivity	graphs	$\log n + \Theta(\log\log n)$
sibling (non-unique)	trees	$\lceil \log n \rceil$
sibling (unique)	trees	$\log n + \Theta(\log\log \Delta)$
connectivity (non-unique)	forest	$\lceil \log n \rceil$
connectivity (unique)	forest	$\log n + \Theta(\log\log n)$
ancestor	trees	$\log n + \Omega(\log\log n)$

Biconnectivity

- Our result: $\log n + \Theta(\log\log n)$
- Previous upper bound: $3\log n$

  [Katz, Katz, Korman and Peleg '02]

Ancestor

- Our result: $\log n + \Omega(\log\log n)$
- Previous upper bound: $\log n + O(\sqrt{\log n})$

  [Alstrup and Rauhe '02]