List Order Maintenance

![Diagram of a linked list with nodes E, B, H, D, I, C, F, G, A, and edge between D and I.]

- Insert(D, I)
- Build data structure

- **Insert(x, y)**: Insert y after x
- **Order(x, y)**: Returns if x is to the left of y

Monotonic List Labeling

![Diagram of a linked list with labeled nodes 10, 12, 14, 17, 18, 19, 20, 21, 24, and edge between 17 and 18.]

- **Insert(x, y)**: Insert y after x
- Each node an integer label
- **Relabel** nodes on insertion

Density Maintenance

![Diagram of a file with nodes A, J, C, G, D, B, F, H, E, and a gap between J and C.]

- **Insert(i, x)**: Insert x at position i
- **Shift** elements on insertion

- **Insert(i, x)**: Insert y after x
- **Order(x, y)**: Returns if x is to the left of y

- **Insert(x, y)**: Insert y after x
- Each node an integer label
- **Relabel** nodes on insertion

- **Insert(i, x)**: Insert x at position i
- **Shift** elements on insertion
List Order Maintenance


Query and Insert $O(1)$

Monotonic List Labeling


Max label $O(n^k)$  \(\Theta(\log n)\) relabelings

[D. Willard, Maintaining Dense Sequential Files in a Dynamic Environment, ACM Conference on Theory of Computing, 114-121, 1982]

Max label $O(n)$  \(\Theta(\log^2 n)\) relabelings

Applications

Amortized $O(\log^2 n)$ Density Maintenance

- **Threshold** $\tau = 1/(2\log n)$
- **Level $i$ node overflows if density $> 1-i\cdot\tau$**
- **Insert** redistribute lowest non-overflowing ancestor

  $\Rightarrow$ a child requires $\tau$ fraction insertions before next overflow
  $\Rightarrow$ amortized insertion cost $= \#\text{levels} \cdot 1/\tau = O(\log^2 n)$

---

**Diagram**

- **Level**
- **Redistribute**
- **Redistribution threshold**
- **Density 5/8**
- **Insert(6,K)**

**Before**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>J</th>
<th>C</th>
<th>G</th>
<th>D</th>
<th>B</th>
<th>F</th>
<th>H</th>
<th>E</th>
</tr>
</thead>
</table>

**After**

|   | A | J | K | C | G | D | B | F | H | E |
Amortized $O(\log^2 n)$ Density Maintenance

- Threshold $\tau = \frac{1}{2 \log n}$
- Level $i$ node overflows if density $> 1 - i \cdot \tau$
- Inserted redistribute/level $i$ node overflowing ancestor
  - a child requires $\tau$ fraction insertions before next overflow
  - amortized insertion cost = $\text{#levels} \cdot \frac{1}{\tau} = O(\log^2 n)$

⇒ List Order Maintenance

Max label $O(n)$

Amortized $O(\log^2 n)$ relabelings / insertion

Before: AJ C G D BF H E
After: AJ K CGD BF H E
Amortized $O(\log n)$ List Relabelings

- **Level** $i$ node overflows if density $> (2/3)^i$
- **Insert** redistribute lowest non-overflowing ancestor
  
  $\Rightarrow \leq \log_{4/3} n$ levels $\Rightarrow \max$ label $2^{\log_{4/3} n} \leq n^{2.41}$
  $\Rightarrow$ a child requires $1/2$ fraction insertions before next overflow
  $\Rightarrow$ amortized insertion cost = $\#\text{levels} \cdot 3 = O(\log n)$

- $2/3 \rightarrow 1/2 + \varepsilon$ implies max label $n^{1+O(\varepsilon)}$
Amortized $O(\log n)$ List Relabelings


Only relabels to the right
- Relabeling area $k$: $w_k = \Omega(k^2)$
- Requires labels $\text{mod } O(n^2)$

$$i = 1$$

**while** $w_{2i} \leq 4 \cdot w_i$ **do**

$$i = i + 1$$

Relabel uniformly "2i area"
Monotonic List Labeling
$O(\log N)$ easy insertions

$\text{Insert}(x,y)$ Label $y = (\text{left} + \text{right})/2$

$\Rightarrow$ Can perform $\log N$ insertions without relabeling
Amortized O(1) List Order Maintenance

top-tree of size $\leq n/\log^2 n$

Amortized $O(\log^2 n)$
Density Maintenance

the list

two-level bucket of degree $[\log n..2\log n]$ and keys $[0..n^2]$

Insertion

– create and label new leaf
– split nodes of degree $> 2\log n$ and relabel with gap $n$
– insert in top tree
Amortized $O(1)$ List Order Maintenance

Worst-case top-tree of size $\leq n/\log^2 n$

Amortized $O(\log^2 n)$ Density Maintenance

Worst-case [Willard 1982]

Two-level bucket of degree $[\log n..2\log n]$ and keys $[0..n^2]

Insertion

- incremental insertion into top tree
- incremental splitting of bucket nodes
- split nodes of degree $> 2\log n$ and relabel with gap $n$
- insert in top tree

$\Rightarrow$ largest bucket size $O(\log^3 n)$
Theorem 5 Let $x_1, \ldots, x_n$ be $n$ real valued variables, all initially zero. Repeatedly perform the following procedure:

1. Find an $i$, $1 \leq i \leq n$, such that $x_i = \max_j \{x_j\}$. Set $x_i$ to zero.

2. Pick $n$ nonnegative reals $a_1, \ldots, a_n$ such that $\sum_{i=1}^{n} a_i = 1$.

3. For $i = 1, \ldots, n$, set $x_i$ to $x_i + a_i$.

No $x_i$ will ever exceed $H_{n-1} + 1$, where $H_k = \sum_{i=1}^{k} i^{-1}$, the $k$th harmonic number.