**List Order Maintenance**

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

**Monotonic List Labeling**

- **Insert** $(i,j)$: Insert $j$ after $i$
- Each node an integer label
- **Relabel** nodes on insertion

**Density Maintenance**

- **Insert** $(i,x)$: Insert $x$ at position $i$
- Shift elements on insertion
- Gap $O(1)$
List Order Maintenance


Query and Insert $O(1)$

Monotonic List Labeling


Max label $O(n^k), k>1+\varepsilon \Theta(\log n)$ relabelings


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 showing density and redistribution thresholds](image)

<table>
<thead>
<tr>
<th>Level</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>before</td>
<td>A</td>
<td>J</td>
<td>C</td>
<td>G</td>
<td>D</td>
</tr>
<tr>
<td>after</td>
<td>A</td>
<td>J</td>
<td>K</td>
<td>C</td>
<td>G</td>
</tr>
</tbody>
</table>

Threshold $\tau = 1/(2\log n)$

Redistribution threshold:
- 4/8
- 5/8
- 6/8
- 7/8
- 1
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
  
  ⇒ a child requires $\tau$ fraction insertions before next overflow
  ⇒ amortized insertion cost $= \#\text{levels} \cdot 1 / \tau = O(\log^2 n)$

![Diagram showing density maintenance with redistribution thresholds and level counts.](image-url)
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

$\Rightarrow$ List Order Maintenance

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

Max label $O(n)$

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

Threshold $\tau = 1/(2\log n)$

1. Level $i$ node overflows if density $> 1-i \cdot \tau$
2. Insert redistribute lowest non-overflowing ancestor

$\Rightarrow$ List Order Maintenance

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

Max label $O(n)$

Amortized $O(\log^2 n)$ relabelings / insertion
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 $n^{1+O(\varepsilon)}$

```
before
A C

after
A C K
```
Amortized $O(\log n)$ List Relabelings


1 2 4 5 7 8 9 12 15 17 18 19 21 23

$i$  \hspace{2cm} $w_i = 12 - 8$

$2i$  \hspace{2cm} $w_{2i} = 18 - 8$

---

$i = 1$

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

$i = i +1$

Relabel uniformly "$2i$ area"

- Only relabels to the **right**
- Max label $M=4n^2$
- Requires labels $\text{mod } M+1$
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

Insertion

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

Worst-case [Willard 1982]

1. top-tree of size $\leq n/\log^2 n$
2. Amortized $O(\log^2 n)$ Density Maintenance

- the list

- three

- 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 rebalance largest bucket
- 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.