Iterators

Suppose we want to look through every element in our data structure:

```
8 → 2 → 5 → ø
```

```
- / 
  a  b  c
```
Iterators encapsulated access to our data:

<table>
<thead>
<tr>
<th>Cur. Location</th>
<th>Cur. Data</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListNode *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>size_t</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stack&lt;Node *&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Disjoint Sets

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8</td>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
```
Disjoint Sets – Smart Union

We will show the height of the tree is: \( \log(n) \).

<table>
<thead>
<tr>
<th>Union by height</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>-4</td>
<td>10</td>
<td>7</td>
<td>-3</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Union by size</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>-8</td>
<td>10</td>
<td>7</td>
<td>-4</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

**Idea:** Keep the height of the tree as small as possible.

**Idea:** Minimize the number of nodes that increase in height.
Union by Size

To show that every tree in a disjoint set data structure using union by size has a height of at most $O(\log n)$ we will show that the inverse.

Base Case

Inductive Hypothesis
Union by Size

Case 1
Union by Size

Case 2
Union by Height - Rank

Base

New UpTrees have Rank =

When you join two UpTrees
Union by Rank

1. For all non-root nodes $x$, $\text{rank}(x) < \text{rank}(\text{parent}(x))$

2. Rank only changes for roots and only up
Union by Rank

Much like before we will show the min(nodes) in a tree with a root of rank $k \geq 2^k$

Base Case

IH
Union by Rank

For any integer $r \geq 0$, there are $\leq n/2^r$ nodes with rank $r$.

Why?
Path Compression