Fibonacci Heaps

- Finite but may become infinite
- Lazy Binomial trees (Heaps)
- Amortised analysis
-lazy

<table>
<thead>
<tr>
<th>Operation</th>
<th>Time Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make Heap</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Insert</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Min</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Extract Min</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>Dequeue Key</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Delete</td>
<td>$O(\log n)$</td>
</tr>
</tbody>
</table>

**Amortised Cost**

$$\text{Amortised Cost} = \text{Actual Cost} + \Delta \phi$$

$$\phi(H) = \sum \phi(t) - \phi(H)$$

**Theorem**

There is a function $D(n) = \Theta(\log n)$ which is the maximum possible degree of any node in a Fibonacci Heap of $n$ items.
Amortized cost =

Actual cost = \[ O(c) - c + O(1) \]

- Potential before = \[ \left( c(v) + c(l) \right) \]

+ Potential after = \[ \left( c(v) + c(l) \right) - (c-1) + 1 \]

\[ \begin{align*}
&+ c-1 \\
&2c + 2 \\
&-2c + 4 \\
&-c + 3
\end{align*} \]

To get degree \( k \) need at least \( F_k \) nodes

\[ A \in \text{final node possible to get degree } k \text{ at most} \]

\[ A \geq 1 + 2 \]

\[ a_2 \geq 1 + 3 \]

\[ a_2 \geq 1 + 3 = 6 \]

\[ a_2 \geq 1 + 3 = 6 \Rightarrow 2^{a_2} \geq 1 + 1 + 1 + \cdots + 2^{a_2-2} \]

\[ \Rightarrow 2^{k} \geq F_{k-1} \text{ by induction} \]