Binomial Heaps. The heap-ordered trees in F-heaps are very regular since they are generated either from scratch or by joining two trees of equal degrees at the root.

Binomial heaps

number of nodes  1  2  4  8  16
degree of root   0  1  2  3  4

Lemma. (i) Every tree in an F-heap is a binomial heap
(ii) A tree with the root of degree \( k \) has \( 2^k \) nodes
(iii) Each node in a tree with \( n \) nodes has degree \( \leq \log_2 n \)
The problems with Decrease-key and Delete operations are that they destroy the binomial-heap property.

**Cascading cuts.** Let \( x \) be a node that becomes a child of another node at time \( t \).

1. The first time \( x \) loses a child after \( t \) is marked.

2. The second time \( x \) loses a child after \( t \), we unlink \( x \) from its parent and add it to the root cycle.

Note: operation 2 may cause another operation 2 and hence the “Cascading cuts”
Bounding the maximum degree.

Lemma 2. Let \( x \) be any node and \( \text{degree}(x) = k \). Let \( y_1, y_2, \ldots, y_k \) denote the children of \( x \) in the order in which they were linked to \( x \), from the earliest to the latest. Then, \( \text{degree}(y_i) \geq 0 \) and \( \text{degree}(y_i) \geq i-2 \) for \( i = 2, 3, \ldots, k \).

Proof. Before \( y_i \) became a child of \( x \), \( x \) had at least \( i-1 \) children and so had \( y_i \). Later, \( y_i \) lost at most one child and therefore its degree is at least \( i-2 \).

Lemma 3. Let \( x \) be any node with \( \text{degree}(x) = k \). Then, no. of descendants of \( x \) including itself is at least \( F_{k+2} \geq \phi^k \) where \( \phi = \frac{1 + \sqrt{5}}{2} \).

Proof. \( S_k = \) # of descendants if degree \( k \)

\[
S_0 = 1, \quad S_1 = 2 \\
S_k \geq 2 + \sum_{i=2}^{k} S_{i-2} \\
\text{By induction on } k \\
\Delta_k \geq F_{k+2}
\]

\[
S_k \geq 2 + \sum_{i=2}^{k} S_{i-2} \\
\geq 2 + \sum_{i=1}^{k} F_i \\
= 1 + \sum_{i=1}^{k} F_i \\
= F_{k+2}
\]

\[
F_0 = 0, \quad F_1 = 1, \quad F_k = F_{k-1} + F_{k-2}
\]

Induction basis \( k = 0, 1 \).
Corollary. The maximum degree \( D(n) \) of any node in an \( n \)-node F-heap is \( O(\log n) \).

**Proof.** Let \( x \) be any node with \( \text{degree}(x) = k \). We have \( n \geq \#\text{descendants of } x \geq \phi^k \),

Thus, \( k \leq \lfloor \log_\phi n \rfloor \)

\( D(n) = O(\log n) \)

---

**Time analysis of Decrease-key, delete**

**Actual cost** = \( O(c) \) \( c \) calls of (cascading) cuts.

Increase in Pot. = \( c \) more roots

Decrease in Pot. = \( c-1 \) nodes unmarked

(last cut might have marked a node)

Change in Pot. = \( c - 2(c-1) \)

\( = -c + 2 \)

**Actual cost** = \( O(c) - c + 2 = O(1) \)

*Amortized* with scaling