1. Define size(x) as #(nodes) in the subtree rooted at node x (including x itself).

- x can be any node in heap.

Lemma 19.1 Suppose x.degree $= k$, let $y_1 \sim y_k$ be the children of x in the order in which they were linked to x, from the earliest to the latest. Then $\begin{cases} y_1.\text{degree} \geq 0 \\ y_i.\text{degree} \geq i-2, \ i = 2 \sim k \end{cases}$

$\rightarrow$ y_1.degree ≥ 0 obviously

$\rightarrow$ For $i \geq 2$, when y_i is inserted, y_{i-1} were children of $x \rightarrow x.\text{degree} \geq i-1$.

$\rightarrow$ Since CONSOLIDATE() moves y_i a child of $x \leftrightarrow y_i.\text{degree} = x.\text{degree} \rightarrow y_i.\text{degree} \geq i-1$

$\rightarrow$ Since then, y_i can lost at most one 1 before it is cut from x (CASCADING_CUT())

$\rightarrow$ $y_i.\text{degree} \geq (i-1) - 1 = i-2$

Lemma 19.2 $\forall k \geq 0, k \in \mathbb{N}$, $F_{k+2} = 1 + \sum_{i=0}^{k} F_i$, where F_i is the i-th Fibonacci number, with $\begin{cases} F_0 = 0 \\ F_1 = 1 \end{cases}$

Prove by induction on k. Base case $k=0$, $F_2 = 1 + F_0 = 1 + 0 = 1$

Assume for some $(k-1) \geq 0$, we have $F_{k+1} = 1 + \sum_{i=0}^{k-1} F_i$.

And we have $F_{k+2} = F_{k+1} + F_k = \left(1 + \sum_{i=0}^{k-1} F_i\right) + F_k = 1 + \sum_{i=0}^{k} F_i$ #

演算法 HW4
109511028
黄威凱

Lemma 19.3 $\forall k \geq 0, k \in \mathbb{N}$, the $(k+2)$nd Fibonacci number satisfies $F_{k+2} \geq \phi^k$, $\phi = \frac{1+\sqrt{5}}{2}$.

Prove by induction on k.

Base case $\begin{cases} k=0, & F_2 = 1 + F_0 = 1+0 = 1 \geq \phi^0 \\ k=1, & F_3 = 2 \geq 1.619 > \phi^1 \end{cases}$

Assume for all $0 \leq i < k$, we have $F_{i+2} \geq \phi^i$.

We have $F_{k+2} = F_{k+1} + F_k \geq \phi^{k-1} + \phi^{k-2} = \phi^{k-2}(\phi+1)$. Note that ϕ is a solution to $x^2 = x+1$

$\rightarrow \phi^{k-2}(\phi+1) = \phi^{k-2} \cdot \phi^2 = \phi^k \rightarrow F_{k+2} \geq \phi^k \quad \forall k \geq 0$.

Lemma 19.4 Let x be any node in a Fibonacci heap, and $k = x.degree$. Then $size(x) \geq F_{k+2} \geq \phi^k$

Let s_k be the minimum possible size of any node of degree k in any Fibonacci heap.

Clearly, $s_0 = 1$, $s_1 = 2$, and s_k is at most $size(x)$.

Since adding a child cannot decrease a node's size, s_k increases monotonically with k.

Let z be a node in any Fibonacci heap, and $z.degree = k$, and $size(z) = s_k$.

Here we try to compute a lower bound on s_k and hence lower bound on $size(x)$.

$size(x) \geq s_k \geq 1$ (for z itself) $+ 1$ (y_1: first child of z) $+ \sum_{i=2}^{k} s_{y_i.degree} \geq 2 + \sum_{i=2}^{k} s_{i-2}$. (by Lemma 19.1) and monotonicity

Then we show by induction on k that $s_k \geq F_{k+2}$ $\forall$ $k \geq 0$, $k \in \mathbb{N}$.

Base case $\begin{cases} k=0, & s_0 = 1 \geq F_2 = 1 \\ k=1, & s_1 = 2 \geq F_3 = 2 \end{cases}$.

Assume for all $0 \leq i < k$, $s_i \geq F_{i+2}$.

(Lemma 19.2) (Lemma 19.3)

We have $s_k \geq 2 + \sum_{i=2}^{k} s_{i-2} \geq 2 + \sum_{i=2}^{k} F_i = 1 + \sum_{i=0}^{k} F_i = F_{k+2} \geq \phi^k$

$\longrightarrow$ $\text{size}(x) \geq s_k \geq F_{k+2} \geq \phi^k$.

<u>Corollary 19.5</u> The maximum degree $D(n)$ of any node in an n-node Fibonacci Heap is $O(\lg n)$

Let x be any node in an n-node Fibonacci heap, and $k = x.\text{degree}$.

By <u>Lemma 19.4</u>, $n \geq \text{size}(x) \geq \phi^k$. Take $\log_\phi(x)$ to each term and we get:

$\log_\phi n \geq k$. (or $k \leq \lfloor \log_\phi n \rfloor$ since $k \in \mathbb{N}$)

The maximum degree $D(n)$ of any node is thus $O(\lg n)$ #