Advanced Data Structures

Lecture by Shay Mozes
Typeset by Steven Karas

2016-11-17

1 Fibonacci Heaps

Design List of heap roots. Keep a reference to the smallest heap.

Heaps are implemented as a pointer to the largest child, and cyclical linked list between siblings.

Notation

$$t(H) = \#\text{ of trees in heap } H$$
$$m(H) = \#\text{ of marked nodes in heap } H$$
$$\phi(H) = t(H) + 2m(H)$$

1.1 Delete-Min

Promote the children of the removed root. Drop the marks from any immediate children. Merge heaps in order of root traversal until at most one heap of each degree is left.

In the slides, some roots are marked. They should not be.

1.2 Decrease-Key

Decrease the key. If the heap condition is violated, disconnect the subheap and merge it into the root list.

Unmarked Parent Mark the parent.

Marked Parent Disconnect the marked parent as a subheap and merge it into the root list. Mark the grandparent and repeat (this is sometimes referred to as cascade).

Complexity Let c be the number of cascading grafts that were performed. $O(c)$.

Amortized Cost $O(1)$.

$$t(H') = t(H) + c$$
$$m(H') \leq m(H) - c + 2$$
$$\Delta$$

1.3 Delete

Decrease the key value to $-\infty$ and Delete-Min.

1.4 Bounding Maximum Degree

$D(N) =$ max degree in Fibonacci heap with N nodes.

Key Lemma $D(N) \leq$

Key lemma. D(N) log N, where = (1 + 5) / 2. Corollary. Delete and Delete-min take O(log N) amortized time. Lemma. Let x be a node with degree k, and let y1, . . . , yk denote the children of x in the order in which they were linked to x. Then: degree(yi) 0 if i 1 i 2 if i 1 Proof. When yi is linked to x, y1, . . . , yi-1 already linked to x, degree(x) =i-1 degree(yi) = i - 1 since we only link nodes of equal degree Since then, yi has lost at most one child – otherwise it would have been cut from x Thus,degree(yi)=i-1 or i-2

$$D(N) =$$

1.4.1 Lemma

In a Fibonacci heap with N nodes, the maximum degree of any node is at most $\log_\phi N$, where $\phi = \frac{1+\sqrt{5}}{2}$

Proof Proof in video around 703pm, slide 60

Let $size(x) = \#$ nodes in the subtree rooted at x. For any node x, we want to show that $size(x) \geq \phi^{\text{degree}(x)}$. Let s_k be the minimum size of the tree rooted at

Fibonacci's Sequence

$$F_k = \begin{cases} 1 & \text{if } k = 0 \\ 2 & \text{if } k = 1 \\ F_{k-1} + F_{k-2} & \text{if } k \geq 2 \end{cases}$$

$$F_k \geq \phi^k, \text{ where } \phi = \frac{1+\sqrt{5}}{2}$$

For $k \geq 2$, $F_k = 2 + \sum_{i=0}^{k-2} F_i$

Formal proofs on slide 64.

2 Hashing

The purpose of Hashing is to manage a set S of elements within some universe U.

Query Is the element $x \in U$ inside S?

Insertion/Deletion $S' = S \cup x$ and $S' = S \setminus x$

Formally Let $h(x) : S \to [m]$, where $[m] = 0, ..., m - 1$.

2.1 Chaining

Colliding elements are stored in a list. Worst case behavior of $O(n)$.

2.2 Perfect Hashing

This is the point where I left the lecture for Thanksgiving dinner.