

Binomial Heaps

Alex Chen

March 5, 2012

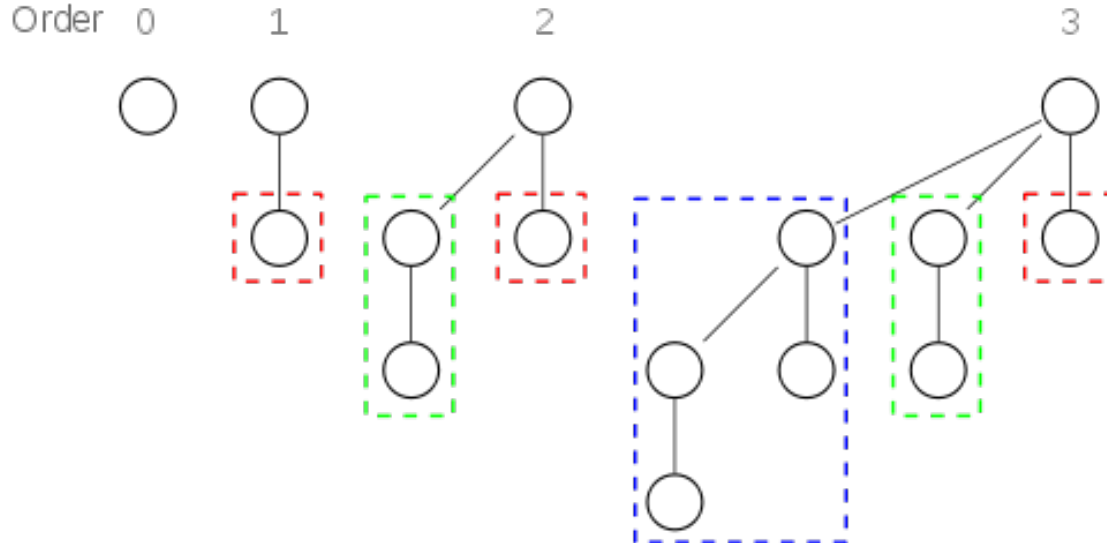
1 Introduction

Binomial heaps are like other heaps and can be used to extract minimum or maximum elements of sets efficiently while maintaining the ability to be easily modified. However, a major advantage of binomial heaps over other heaps is that they can be merged in $O(\log N)$ time.

This handout is only intended to be a guide to the many illustrations in the lecture.

2 Depiction

Image from Wikipedia. A binomial heap consists of individual binomial trees, each of which satisfies the heap property and contains a number of nodes that is equal to a power of 2. A binomial tree of order $n + 1$ can be created by attaching a binomial tree of order n as a child of the root of another binomial tree of order n .



Binomial heaps are named the way they are because the number of nodes at each depth in a binomial tree correspond to elements of Pascal's triangle. Also be sure to note the way these binomial trees are like binary numbers. Merging two binomial heaps is like adding two binary numbers. Adding a single element to the binomial heap is like merging with a heap of size 1.

3 Efficiency

Operation	ArrayList	Binary Heap	Binomial Heap	Fibonacci Heap
insert	$O(1)$	$O(\log N)$	$O(\log N)$	$O(1)$
deletemin	$O(N)$	$O(\log N)$	$O(\log N)$	$O(\log N)^*$
merge two heaps	$O(M)$	$O(N + M)$	$O(\log N)$	$O(1)$

An asterisk means amortized. N refers to the size of the larger heap and M refers to the size of the smaller one.

4 Problem

You are given a rooted tree with $N \leq 100,000$ nodes. Each leaf contains 10 numbers. For each node, find the sum of the 10 largest numbers in its subtree.