CSC263 - Binomial Heaps

Cristyn Howard

Monday, January 29, 2018

Mergeable heaps support the following operations:

- **MAKE-HEAP() :** creates and returns a new heap with no elements.
- **INSERT(H, x) :** inserts node x into heap H.
- **MINIMUM(H) :** returns a pointer to smallest-keyed node in heap H.
- **EXTRACT-MIN(H) :** deletes the node from heap H whose key is minimum, returns a pointer to it.
- **UNION(H_1, H_2) :** creates and returns a new heap that contains all the nodes of heaps H1 and H2. Heaps H1 and H2 are destroyed by this operation.
- ***DECREASE-KEY(H, x, k) :** assigns to node x within heap H the new key value k, requires $k \leq x.key$.
- ***DELETE(H,x) :** deletes node x from heap H.

When heaps are implemented using binary trees stored in arrays, all of the operations listed except for UNION are in $O(\log n)$. For this data structure, UNION is performed by concatenating two arrays and running MIN-HEAPIFY over the result, so it is $O(n)$.

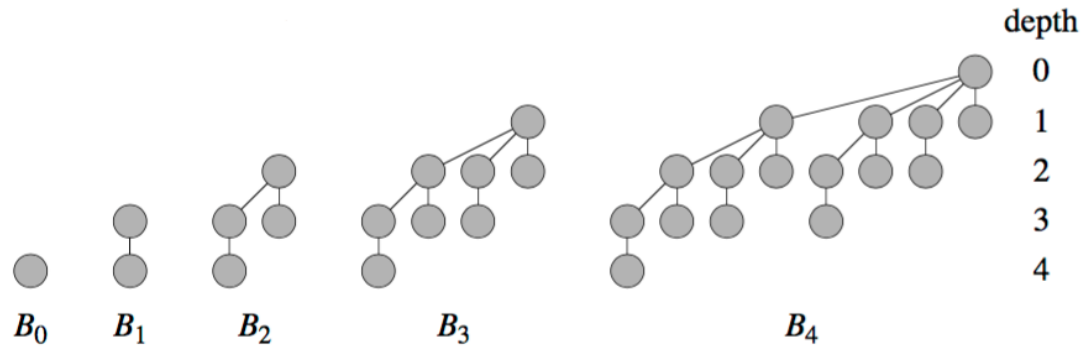
Procedure	**Binary heap**	**Binomial heap**	**Fibonacci heap**
MAKE-HEAP	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
INSERT	$\Theta(\log n)$	$O(\log n)$	$\Theta(1)$
MINIMUM	$\Theta(1)$	$O(\log n)$	$\Theta(1)$
EXTRACT-MIN	$\Theta(\log n)$	$\Theta(\log n)$	$O(\log n)$
UNION	$\Theta(n)$	$O(\log n)$	$\Theta(1)$
DECREASE-KEY	$\Theta(\log n)$	$\Theta(\log n)$	$\Theta(1)$
DELETE	$\Theta(\log n)$	$\Theta(\log n)$	$O(\log n)$

*Note: binary, binomial, and fibonacci heaps are all inefficient at performing SEARCH - finding a node with a given key. Thus the procedures DECREASE-KEY and DELETE require a pointer to the node to be deleted to be input.

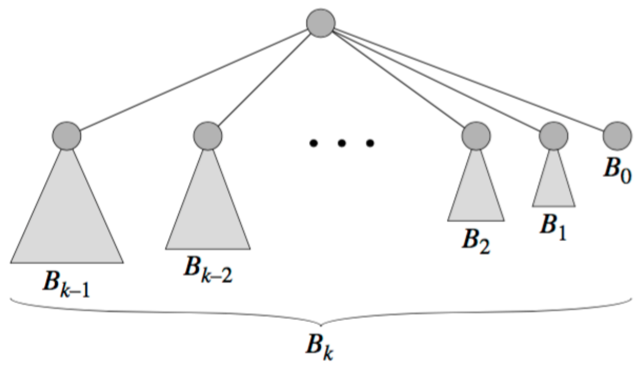
A binomial heap is a collection of **binomial trees**.

Binomial tree B_k : an ordered tree where:

- B_0 is a single node
- B_k consists of two B_{k-1} trees linked together such that the root of one is the leftmost child of the roof of the other.



You can also think of a B_k tree as a collection of $\{B_{k-1}, B_{k-2}, ..., B_1, B_1\}$ attached to a common root node.



Properties of Binary Trees: binomial tree B_k has...

1. 2^k nodes
2. height k
3. exactly $\binom{k}{i}$ nodes at depth $i \in \{0, 1, ..., k\}$
4. root has degree k, which is greater than that of any other node in B
5. maximum degree of any node in an n-node binomial tree is $\log n$
   - *Consider that the number of nodes in the subtree rooted at x decreases by half every time we jump to the leftmost child of x.*

Note: the **degree** of node x the number of children it has. It is equivalent to the number of edges on the longest path between it and a leaf.

A **binomial heap** is a binomial forest, a collection of binomial trees, that satisfies the binomial heap properties:

A) Each binomial tree in H obeys the min-heap property: the key of a node is greater than or equal to the key of its parent.

B) For any nonnegative integer k, there is at most one binomial tree in H whose root has degree k.