CSC263 - Week 2, Lecture 1

Cristyn Howard

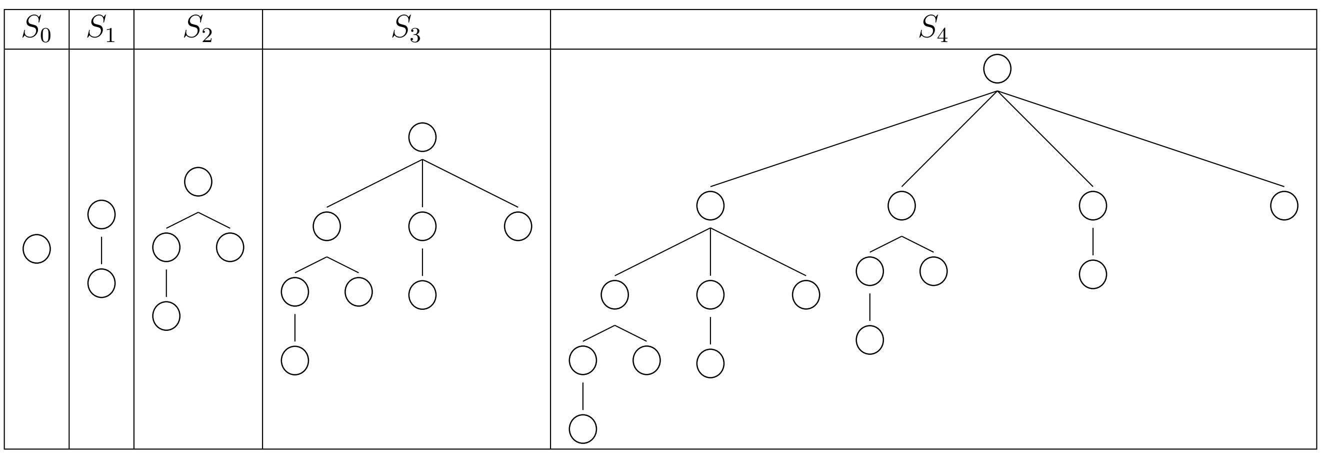
Monday, January 15, 2018

ADT	Data Structure	INSERT	MIN	EXTRACTMIN	MERGE
Priority Queues	Heaps	yes	yes	yes	NO
Mergeable Priority Queues	Binomial Queues	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$

If a CPU has two cores, each with a Priority Queue of tasks to perform, it might need to merge them. This is not easy to accomplish with Heaps, however it is possible with...

<u>Binomial Heaps</u>

S_k *tree* : $S_0 = \bigcirc$
S_k = take two S_{k-1} trees, make the roof of one the parent of the root of the other.



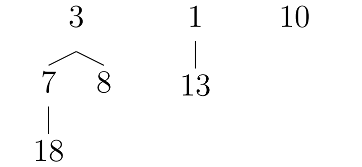
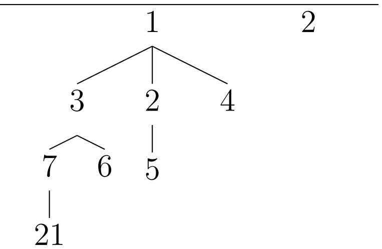
- Each S_k tree is attached to $\{S_0, S_1, ..., S_{k-1}\}$, has 2^k total nodes, and $\binom{k}{d}$ nodes at depth d.

A <u>binomial forest</u> of size m, denoted F_m, is a sequence of S_k trees with increasing k, and a total of m nodes in the entire forest.

- Represent the number m in binary, each 1 digit in the resulting number represents an Sk tree in the forest. $\boxed{\text{Ex: } m = 7 =< 1, 1, 1 >_2 = 2^2 + 2^1 + 2^0 \quad \therefore \quad F_m = \{S_2, S_1, S_0\}}$
- $\alpha(m)$ = number of 1's in the binary representation of m.
- F_m has $\alpha(m)$ trees, and $m - \alpha(m)$ edges.
- Note: a 'forest' is just the structure, it becomes a heap when keys are added.

<u>Min heap</u> has property that the key of the parent is SMALLER than the key of the children.

Some examples of binary forests with keys that conform to the min heap property:

$m = 7 =< 1, 1, 1 >_2$ $F_m = \{S_2, S_1, S_0\}$ $S = \{10, 13, 1, 3, 8, 18, 7\}$	
$m = 0 =< 1, 0, 0, 1 >_2$ $F_m = \{S_3, S_0\}$ $S = \{6, 1, 7, 3, 4, 5, 2, 2, 21\}$	

These forests must have pointers to allow for navigation between trees and interaction with the data structure. These pointers are NOT edges in the trees.

Each node stores pointers to: parent, left child, right sibling.