COL 702 Lecture 7 Aug 13

Priority queues.

Ops : min, delete min, insert, delete

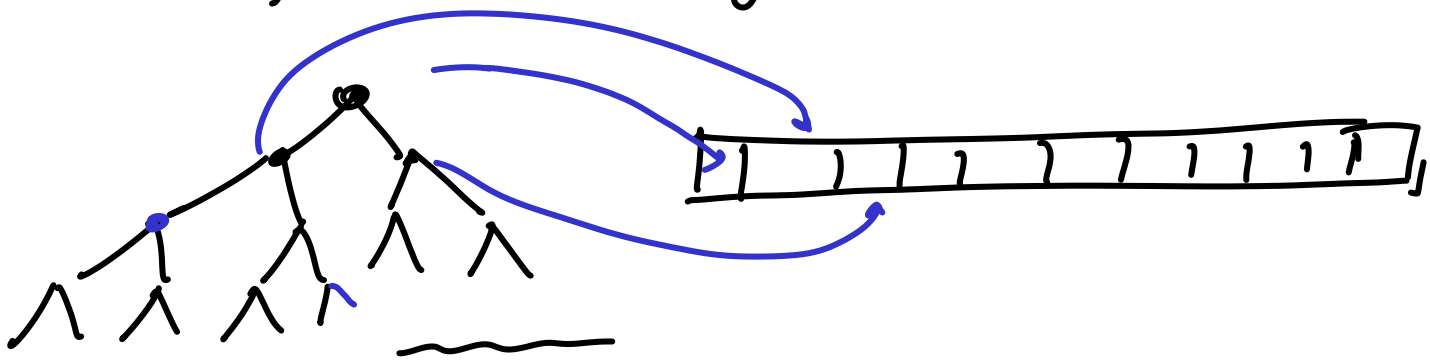
Binary Heaps

min : $O(1)$
delete min : $O(\log n)$
insert / delete : $O(\log n)$

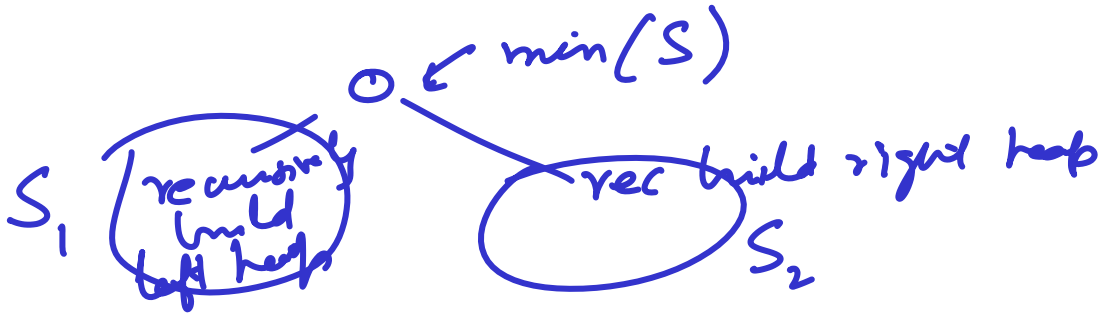
Given a set S, $|S| = n$, build heap (S)

Start fm an empty set $\emptyset$ and keep inserting elements of S

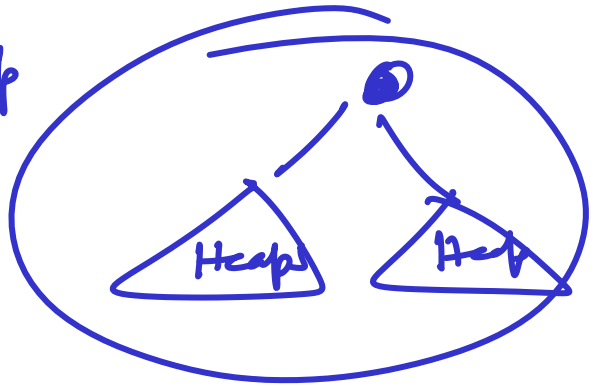
$\rightarrow$ $O(n \log n)$



Top down / Bottom-up



Bottom-up



Is BST also a priority queue?

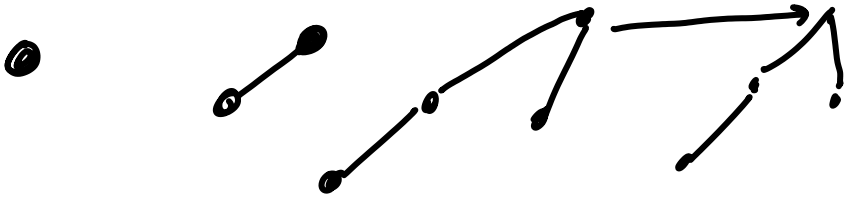
Given priority queues P_1, P_2

we want to merge P_1, P_2

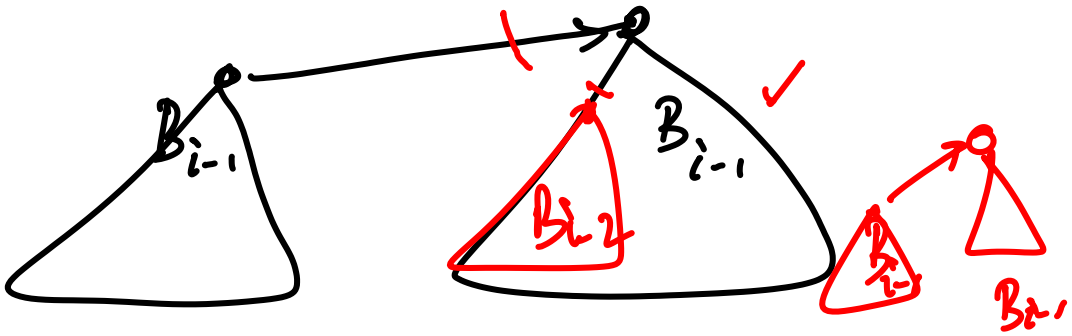
Goal: support merging/melding in $O(\log n)$ time

Binomial tree

B_0 B_1 B_2 .. B_3 .. B_i



B_i



Properties of binomial trees

(1) B_i has 2^i nodes

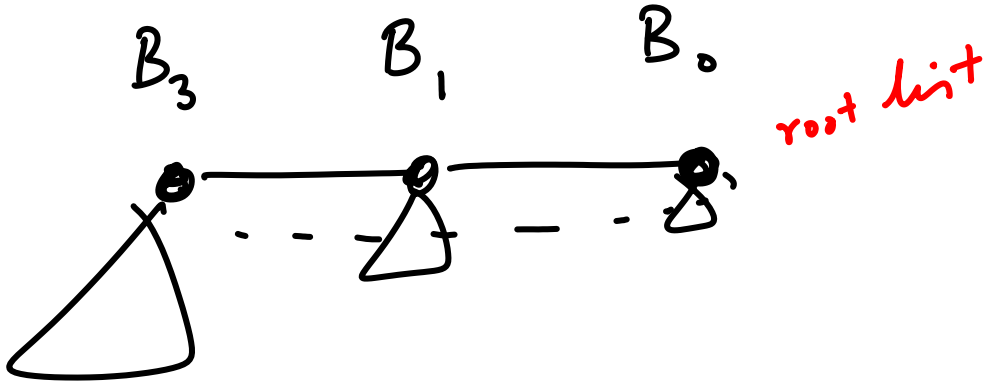
(2) Height of B_i i

(3) degrees of nodes · root has degree i

(4) What is # nodes on level k of B_i

For an arbitrary n, we will use more than one Binomial tree to represent a heap on n values

$n = 11$



At most $\lceil \log_2 n \rceil$ Binomial trees are needed

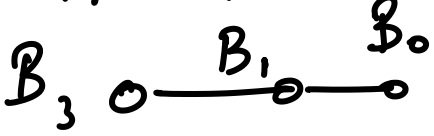
An extra min pointer in the root list

There is at most one B_i for any $i \leq \log_2 n$

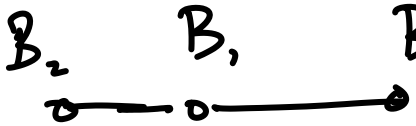
How to merge two Binomial Heaps

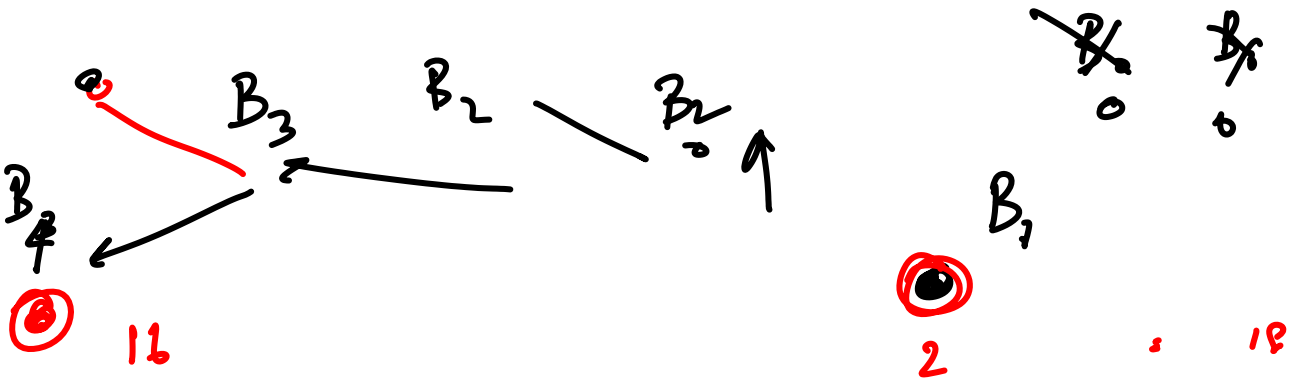
H_1 H_2

$n = 11$



$n = 7$

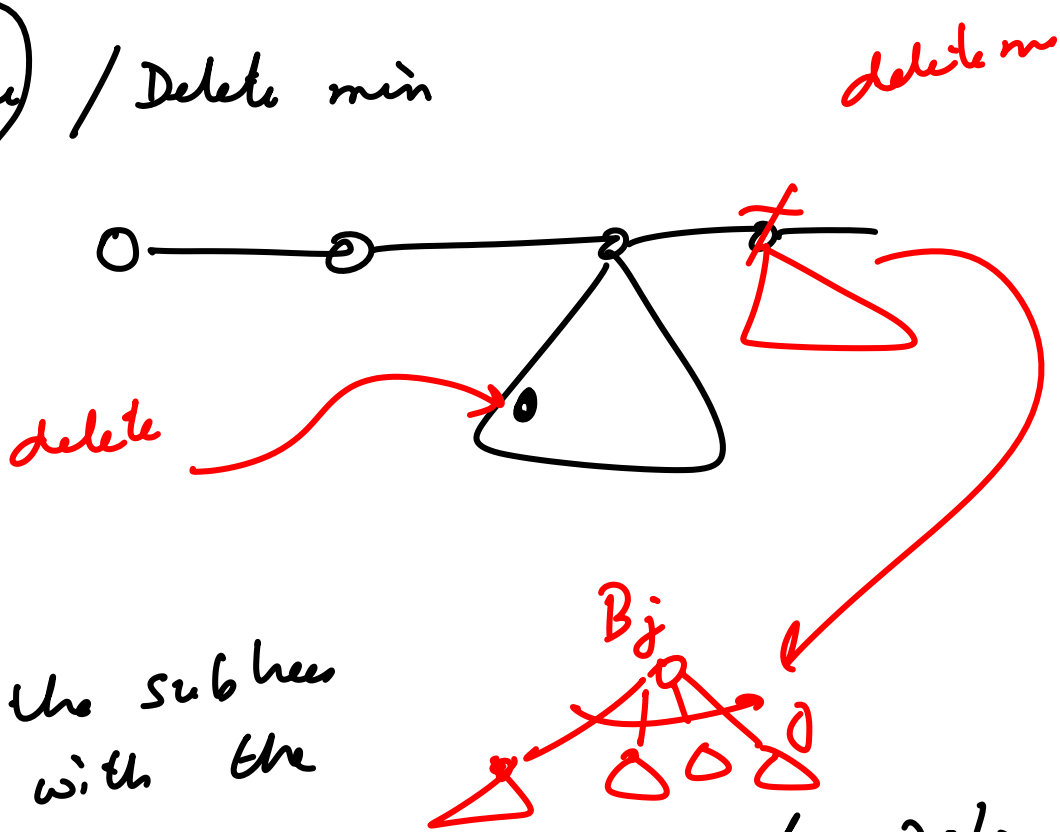




Like binary addition, total time is $O(\log n)$ $n = n_1 + n_2$

Insertion: Is merging one element heap with the rest

Delete / Delete min



Merge the subtrees of B_j with the original root list in $O(\log n)$-time