Priority queues:

Operations: min, delete min, insert, delete

Binary Heaps

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

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

Start from an empty set \(\emptyset\)
and keep inserting elements of \(S\)

\(\Rightarrow \quad O(n \log n)\)

Top-down / Bottom-up
Is BST also a priority queue?

Given two priority queues $P_1$, $P_2$, we want to merge $P_1$, $P_2$

**Goal**: support merging/welding in $O(\log n)$ time
Binomial tree

\[ B_0 \to B_1 \to B_2 \to \ldots \to B_i \]

Properties of binomial trees

1. \( B_i \) has \( 2^i \) nodes
2. Height of \( B_i \) is \( i \)
3. Degrees of nodes, root has degree \( i \)
4. \( \text{What is } # \text{ nodes on level } k \text{ of } B_i \)
For an arbitrary $n$, we will use more than one Binomial tree to represent a heap on $n$ values.

$$n = 11$$

$$B_3, B_1, B_0$$

At most $\left\lceil \log_2 n \right\rceil$ 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 $H_1, H_2$.

$$n = 11$$

$$n = 7$$

$$B_3, B_1, B_0$$

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

**Insertion**: Insert one element heap with the rest

**Delete / Delete min**: Delete min

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