**Insert:** $O(\log n)$ (up a path)
**Delete-min:** $O(\log n)$ (down a path)
**Increase-key, Decrease-key:** $O(\log n)$ (up a path)
**Preprocess:** $O(n)$

Lower bounds:
- $n$ inserts/deletes-mins require $\Omega(n \log n)$ time in comparison model by reduction from sorting.

**Q:** Faster `insert`? `decrease-key`?

**Binomial Heaps (Vuillemin ’78):**
- `find-min`: $O(1)$
- `insert`: $O(1)$ (amortized)
- `delete-min`: $O(\log n)$

**Fibonacci Heaps (Fredman-Tarjan ’85):**
- `decrease-key`: $O(1)$ (amortized)

[applies to Dijkstra’s algo $n$ delete-mins, $m$ decrease-keys $\Rightarrow O(n \log n + m)$]

Other alternatives:
- Takaoka ’03: “2-3 heaps”
- C’09: “quake heaps”
- Brodal ’96: worst case
- Pairing Heaps: simplified Fibonacci heaps but poorer guarantees

---

**Tournament Tree Approach:**
allow multiple tournament trees (forest)

insert(x): // be lazy!
    just create a new tree for \{x\}

delete-min U:
    \( X = \text{min} \) of all the roots
    remove path of \( X \)'s nodes
    whenever \( \exists 2 \) trees of same height
    link them

at end, \( \leq \log n \) trees

Amortized Analysis:

define potential \( \Phi = \# \text{trees} \)

for each insert:
    change in \( \Phi \) = +1
    runtime = \( O(1) \)
    \( \leq O(2 - \text{change in } \Phi) \)

for each delete-min:
    let \( t = \# \text{trees before} \)
    change in \( \Phi \) \( \leq (\log n) - t \)
    runtime = \( O(t + \log n) \)
    \( = O(2 \log n - \text{change in } \Phi) \)

total time for \( n \) inserts \& \( n \) delete-mins
total time for \( n_I \) inserts & \( n_D \) delete-mins
\[
\leq O(2n_I + (2\log n)n_D - \text{total change in } \Phi)
\]
(find \( \Phi \) - initial \( \Phi \))
\[
\leq O( n_I + (\log n)n_D )
\]
\[
\Rightarrow \text{ insert } O(1) \text{ amortized}
\]
\[
\Rightarrow \text{ delete-min } O(\log n)
\]

What about decrease-key\((x)\)?

basic idea - cut subtree at \( x \)
\( x \) is now a root
- can decrease key trivially!

but tree may have deg-1 nodes
& unbalanced.

Solution 1 - 2-3 Heaps (Takaoka)

\( \text{Invariants} \quad \text{all nodes have deg } 2 \text{ or } 3 \)
\( \text{All leaves in same tree have same depth} \)

Whenever 3 deg-1 node \( v \)
- fuse \( v \) with its sibling
- split if deg \( \leq 4 \)
Amortized Analysis:

New potential \( \Phi = \# \text{trees} + \# \text{nodes} \)

insert: \( O(3 - \Delta \Phi) \)

delete-min: \( O(2 \log n - \Delta \Phi) \)

decrease-key: 
  \[ \Delta \Phi = -(f - 1) + 1 \]
  \( f = \frac{\# \text{fuses}}{\# \text{nodes}} \)
  \( (\# \text{trees}) \)

\[ \text{time} = O(1 + f) \leq O(3 - \Delta \Phi) \]

Total time for \( n_I \) insert, \( n_D \) delete-min, \( n_{DK} \) decrease-key

\[ \leq O(3n_I + (2 \log n) n_D + 3n_{DK} \) - total \( \Delta \Phi \)
  \( \frac{\Phi_{\text{final}} - \Phi_{\text{init}}}{O} \)

\[ \leq O(n_I + (\log n) n_D + n_{DK}) \]

insert \( O(1) \)

delete-min \( O(\log n) \) 

decrease-key \( O(1) \) \} amort.

Solution 2 - Fibonacci heaps \( \Rightarrow \) cascading cuts

skipped

Solution 3 - Quake heaps

Idea - be lazy + clean up once in a while
(Invariant:)
- All nodes have deg 1 or 2
- All leaves in same tree have same depth

\[ n_{i+1} \leq \alpha n_i \quad \implies \quad \text{global balance constraint} \]

\[ \frac{\text{# nodes at height } i+1}{\text{# nodes at height } i} \leq \alpha \]

\[ \Rightarrow \quad \text{height } O\left(\log_{1/\alpha} n\right) \]

No fuse etc.

Sesmic event whenever \( n_{i+1} > \alpha n_i \) \( \implies \) just remove all nodes at height > i

Amortized Analysis:

New potential \( \Phi = \text{#trees} + \text{#nodes} + \frac{1}{2\alpha-1} \text{(deg-1 nodes)} \)

Insert: \( O(3 - \Delta \Phi) \)
Delete-min: \( O\left( O\left(\log n\right) - \Delta \Phi \right) \) ignoring clean-up

Decrease key: Say \( n_{i+1} > \alpha n_i \).

\[ \Delta (\text{#trees}) \leq + n_i \]
\[ \Delta (\text{#nodes}) \leq - n_{i+1} - n_{i+2} - \cdots \]
\[ \Delta (\text{deg-1 nodes}) \leq + 1 - (2n_{i+1} - n_i) \]
\[ \leq + 1 - (2\alpha - 1) n_i \]

\[ \Rightarrow \quad \Delta \Phi \leq + 1 - n_{i+1} - n_{i+2} - \cdots \]
runtime $O(1 + n_{i1} + n_{i2} + \ldots)$

$\leq O\left(2 - \frac{\Delta \Phi}{\gamma}\right)$

$\Rightarrow$ decrease-key $O(1)$ amortized.

delete-min $O(\log n)$

insert $O(1)$