

HW 7 extended to next week 2x credit

(Lots of details to get right!)

PLEASE look at graded solutions on Gradescope!

Analgin OH: Mon 6-7

next Sat back to normal
next Fri back to normal

Self OH:

Mergeable heaps / PQs

- Insert
- Merge
- Extract Min
- Decrease Key

Leftist heap: $O(\log n)$
worst case

Binomial heap: $O(\log n)$
worst case

At least one of these
must take $\Omega(\log n)$ time.

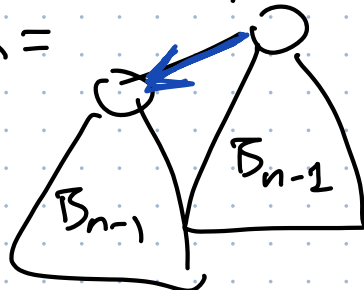
Fibonacci heap ^{Dec Key}
 Insert/Merge $O(1)$ amortized
~~Decrease~~/Ex Min: $O(\log n)$

Pairing heap Insert Merge $O(1)$ $- O(2^{\sqrt{2 \lg \lg n}})$
 Dec Key $o(\log n)$ $- O(2^{\sqrt{2 \lg \lg n}})$
 Ex Min $O(\log n)$

Binomial heap = list of binomial trees
heap-ordered

$B_0 = \circ$

$B_n =$



Insert = increment
 Merge = add

| $O(\log n)$ time

LAZY Insert + Merge: $O(1)$ worst case + amort
 ↳ add B_0 ↳ concat lists

Extract Min:

delete + remember min - for node

repeatedly ^{link} merge same ^{rank} size trees $\leftarrow O(\#roots)$ time

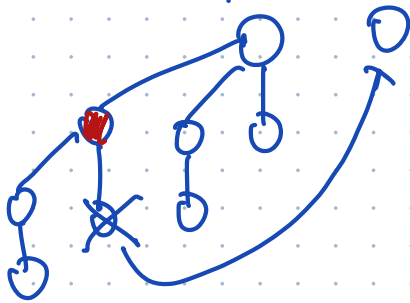
Find new min $\rightarrow O(\log n)$??

↑
charge each root to Insert

$O(\log n)$ amortized

Decrease Key:

Bubble or shatter $\Rightarrow O(\log n)$ time



Promote(v):

move v to main list
if parent(v) is marked

$O(\log n)$
worst case

unmark P
Promote P
else
mark P

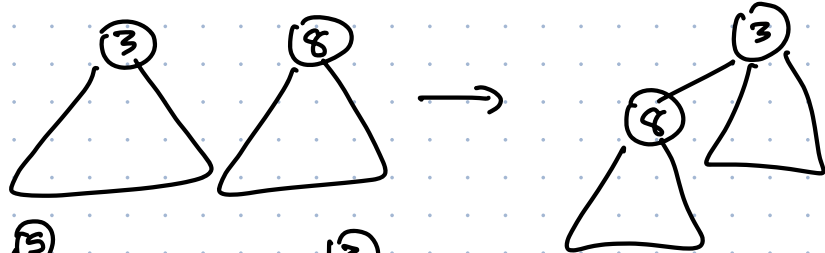
Charge to
past prom.
marked P.

$O(1)$ am. time

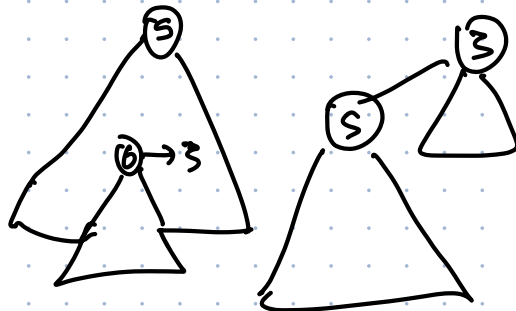
Key: $F_k \leq \#nodes\ in\ tree\ with\ rank\ k \leq 2^k$

Pairing heap: Single heap-ordered tree.

Merge = link
 $O(1)$

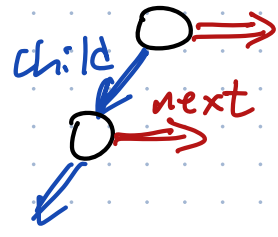
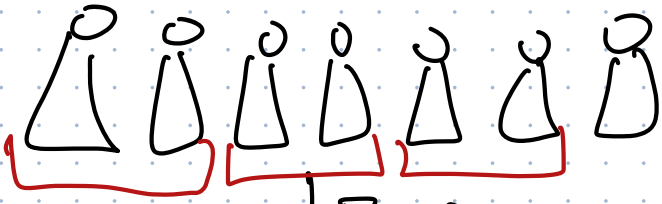


Decrease Key



Extract Min

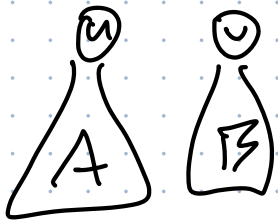
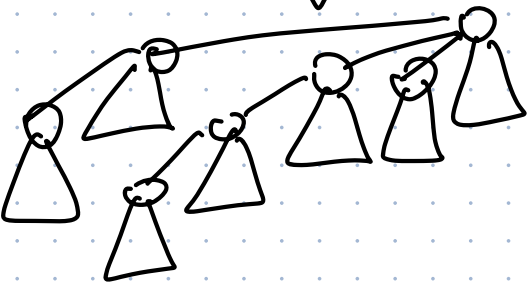
$\Theta(n)$ worst case



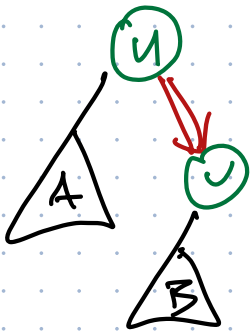
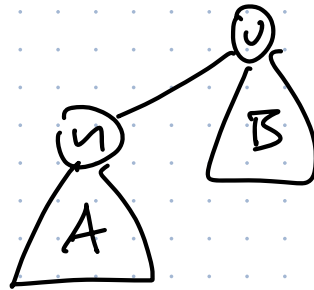
Pairing — pairs



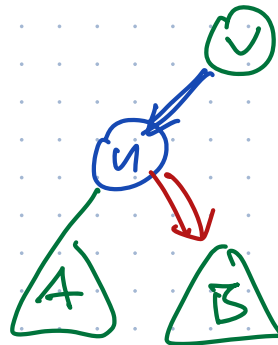
Cleanup — link last two

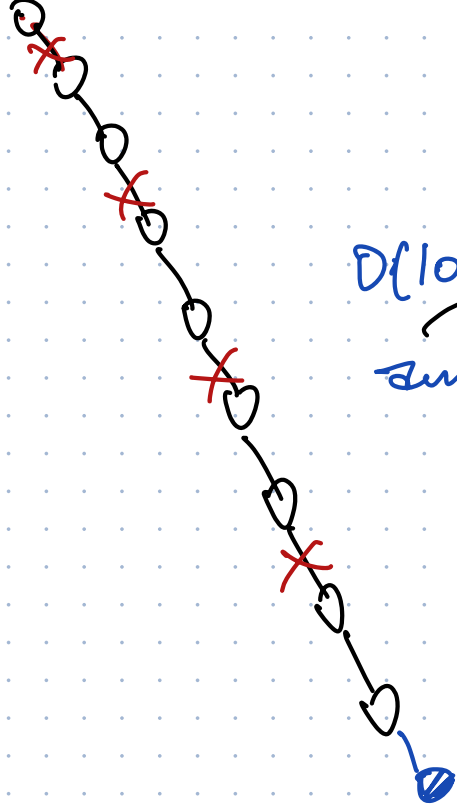


link →



rotation →





$O(\log n)$
sum. time

