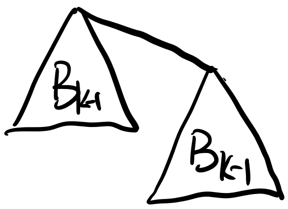
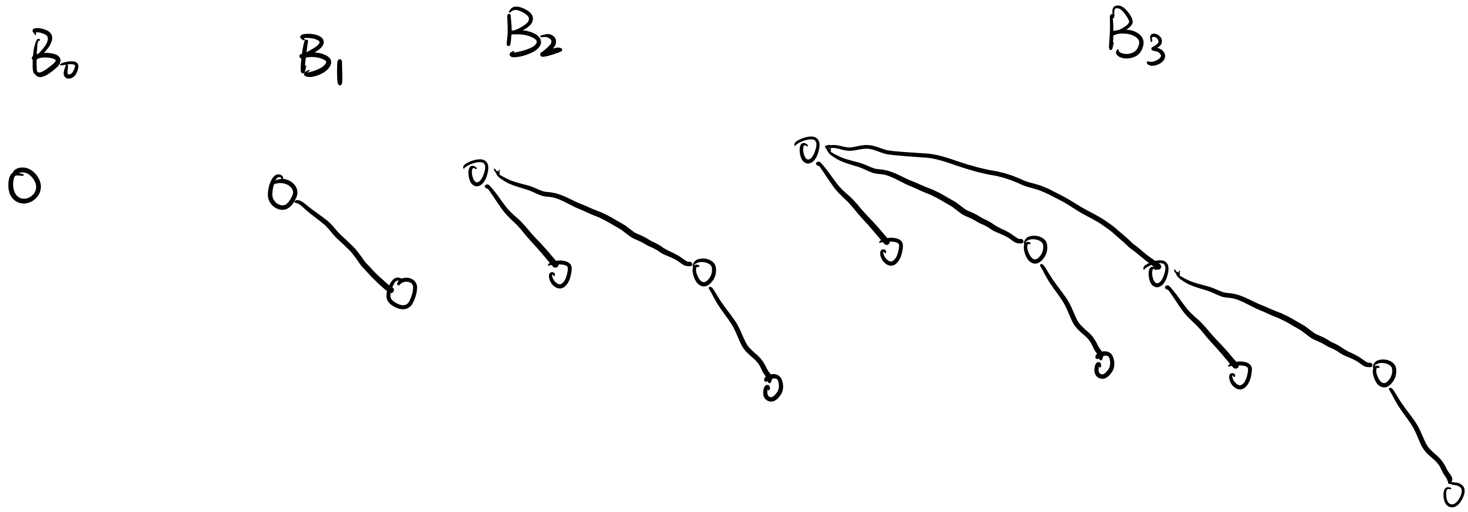
binomial tree

height = 0 $\quad B_0$

height = k $\quad B_k$

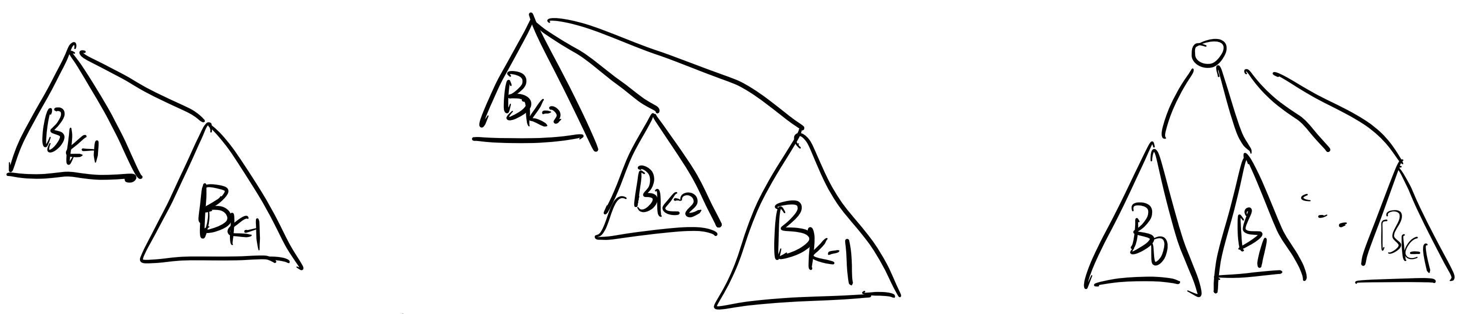




height of B_k is k

children of the root of B_k is k

the subtrees rooted at them are $B_0, B_1, \ldots, B_{k-1}$



nodes in $B_k = 2^k$

nodes at depth d of $B_k = \binom{k}{d} = \frac{k!}{(k-d)!\,d!}$

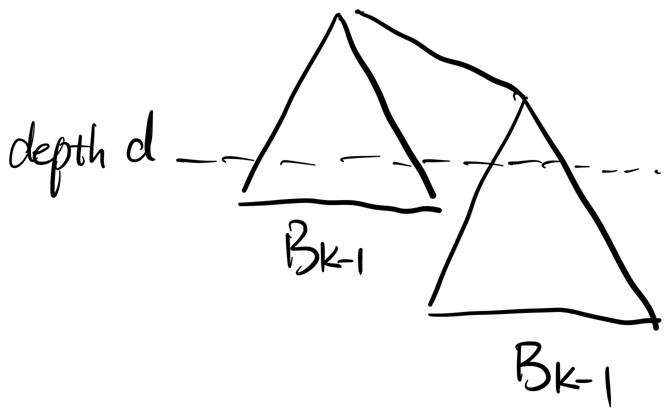
binomial coefficient

by induction on k

base case B_0 $\binom{0}{0}=1$ true

inductive hypothesis B_{k-1} true $\binom{k-1}{d}$ for any d

inductive step B_k



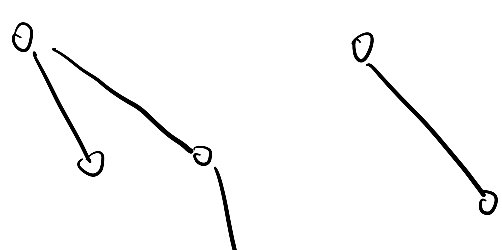
$$\binom{k-1}{d} + \binom{k-1}{d-1} = \binom{k}{d} \qquad \square$$

A binomial heap is a forest of

1. binomial trees of distinct height
2. each of which is in heap order.

A binomial heap with 6 nodes

$$6 = \underbrace{2^2}_{B_2} + \underbrace{2^1}_{B_1} + 0 \times 2^0 = 110_2$$



$13 = 1101_2$

B_3 B_2 B_0

8 4 1

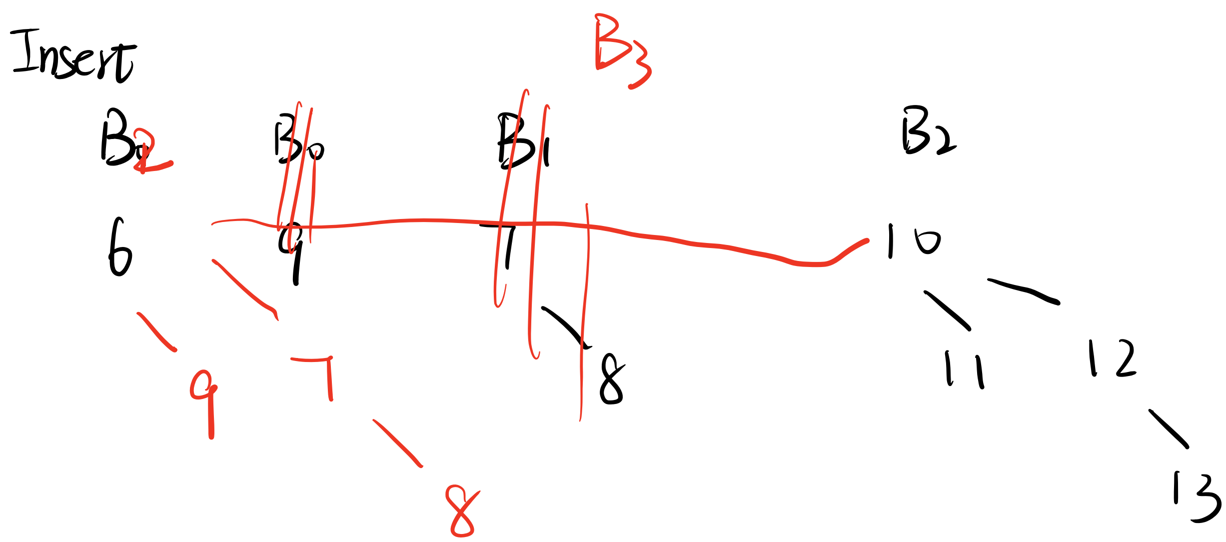
\# trees in a binomial heap with n nodes $\le \lg_2 n$

findmin

return minimum root of the trees $O(\lg n)$

maintain a pointer to the min root $O(1)$

Insert



Ins(9) Ins(6)

110_2 111_2 $+1$ 1000_2

Insert (H, x)

1. create a B_0 with key x and add it to H
2. i=0
3. while H has two B_i's
   combine them into B_{i+1}
   i=i+1

\# trees

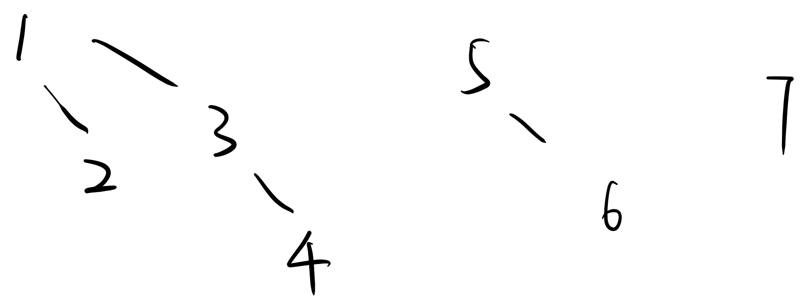
+1

\# combines = $O(\lg n)$

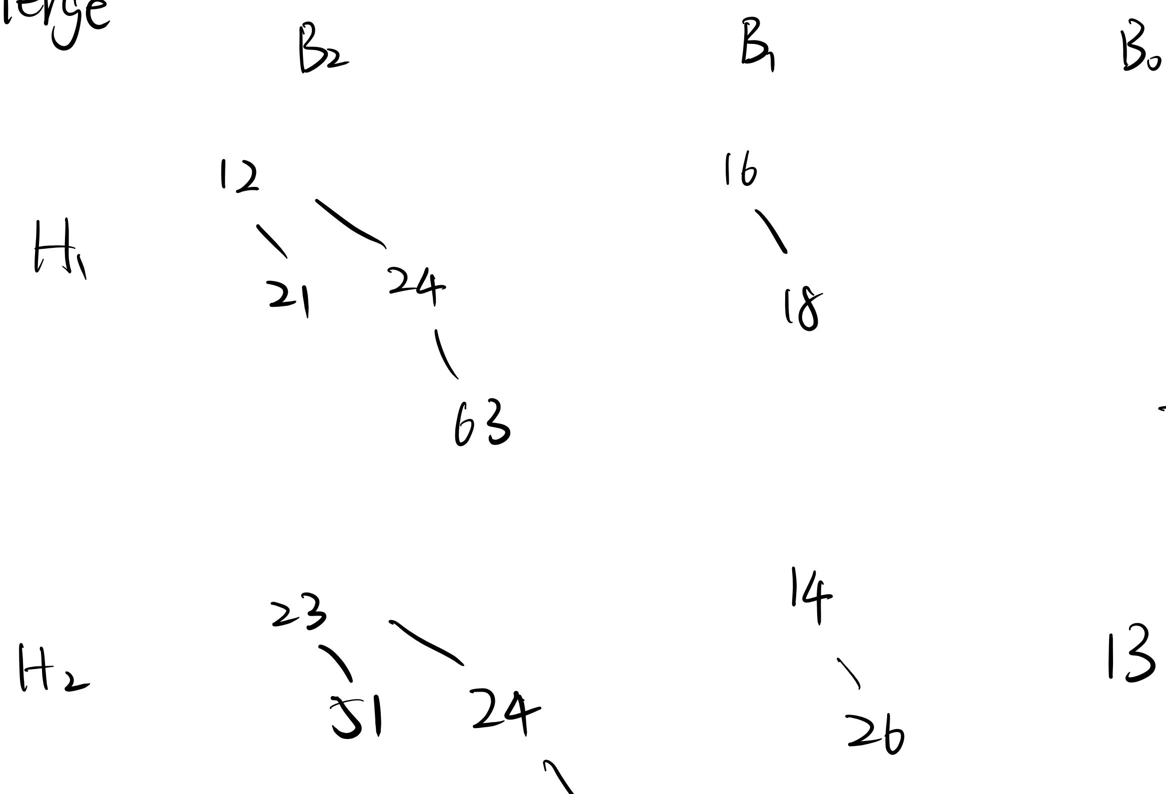
− # combines

$\Rightarrow$ Δ#tree = 1 − #combines

Insert 1~7 into an empty binomial heap

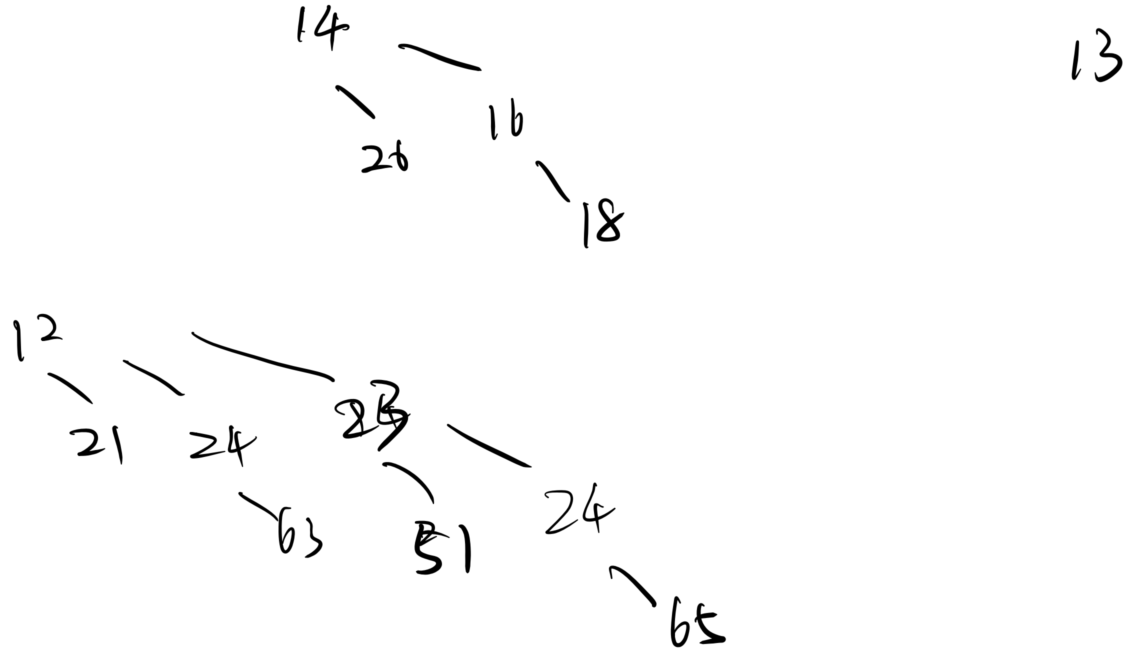


Merge



$$\begin{array}{r} 110_2 \\ +\ 111_2 \\ \hline 1101_2 \end{array}$$

65



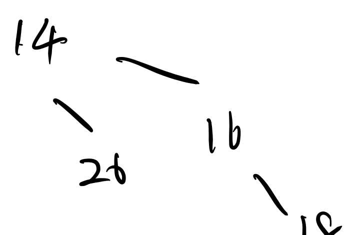
Merge(H_1, H_2)

1. let k_1 and k_2 be # trees in H_1 and H_2 respectively
2. for $i = 0$ to $\max(k_1, k_2)$
3. if there are more than one B_i's
4. combine any of them into a B_{i+1}

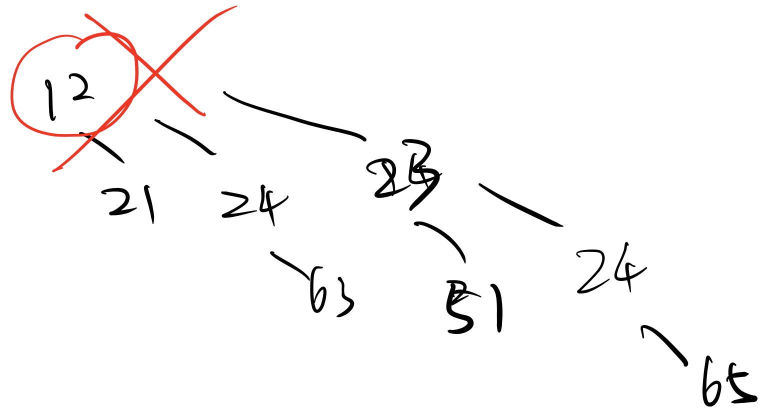
$$O(k_1 + k_2) = O(\lg n_1 + \lg n_2) = O(\lg n) \quad \underline{n_1 + n_2}$$

deletemin(H)

H



H'

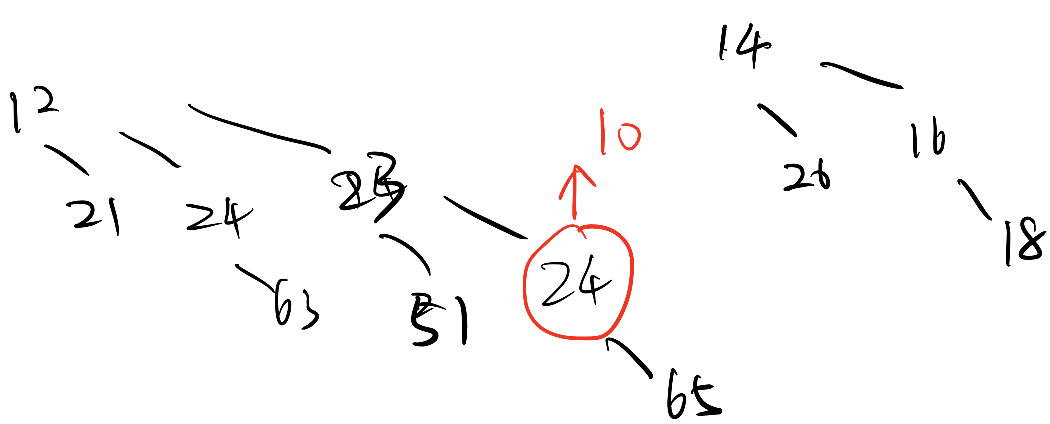


Merge(H, H')

Deletemin(H) $\longrightarrow O(\lg n)$

1. find the B_k that contains the min $O(1)$
2. $H = H - B_k$ (remove B_k from H) $O(1)$
3. $H' = B_k - \text{root}$ (delete the root of B_k) $O(k) = O(\lg n)$
4. Merge(H, H') $O(\lg n)$

Decreasekey(x, k)



$O(k) = O(\lg n)$

Delete(x)

decreasekey(x, $-\infty$) + deletemin $O(\lg n)$

Claim n insertions on an empty binomial heap needs $O(n)$ time in total.

Proof 1:

$$O(n) + \#\text{total combines}$$
$$= O(n) + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \cdots + \frac{n}{2^{\log_2 n}}$$
$$= O(n) + n$$
$$= O(n)$$

Proof 2:

$\Phi(D) = \#$ trees in D.

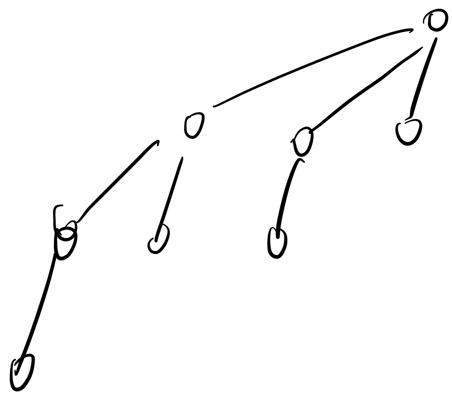
$$\text{actual cost} = O(1) + \#\text{combines}$$
$$= O(1) + 1 - \Delta\#\text{trees}$$

$$\text{amortized cost} = O(1) + 1 - \Delta\#\cancel{\text{trees}} + \cancel{\Delta\Phi}$$

$= a1)$

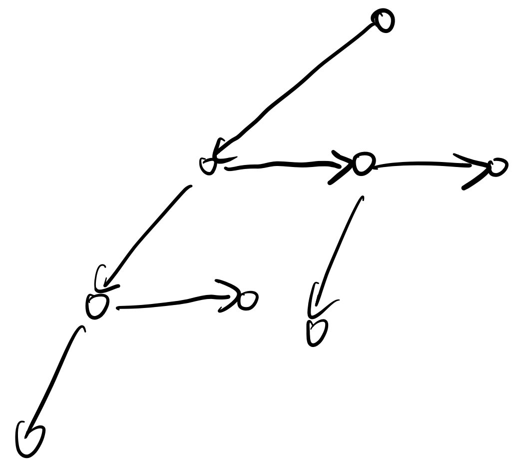
theory

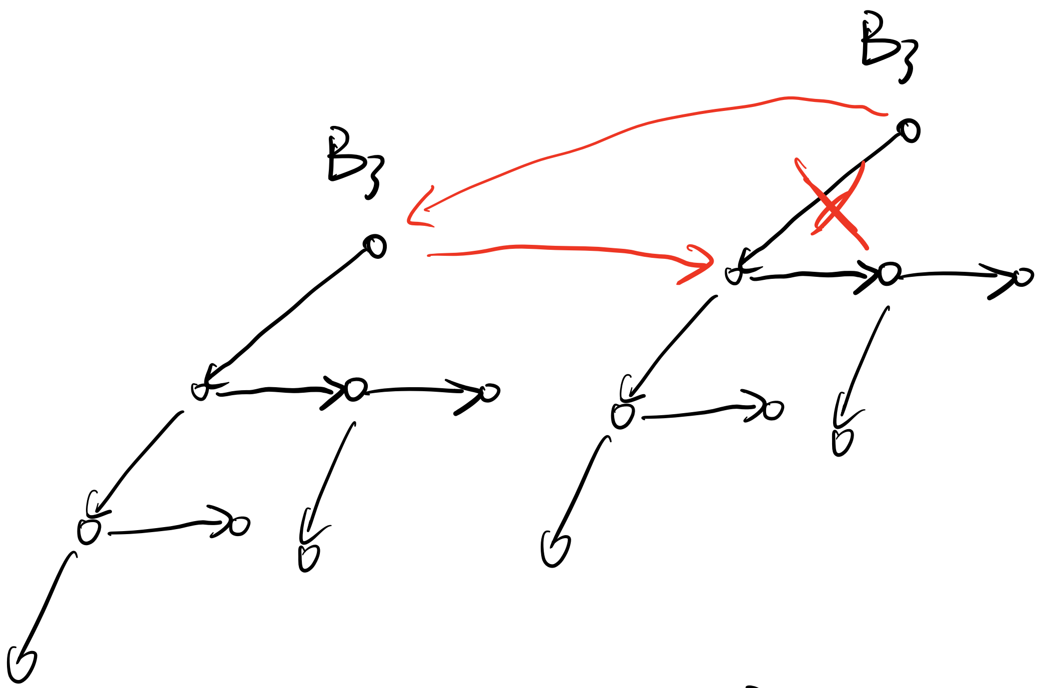
B_3

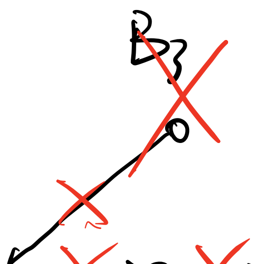


left-child-next-sibling.

B_3







$O(k)$