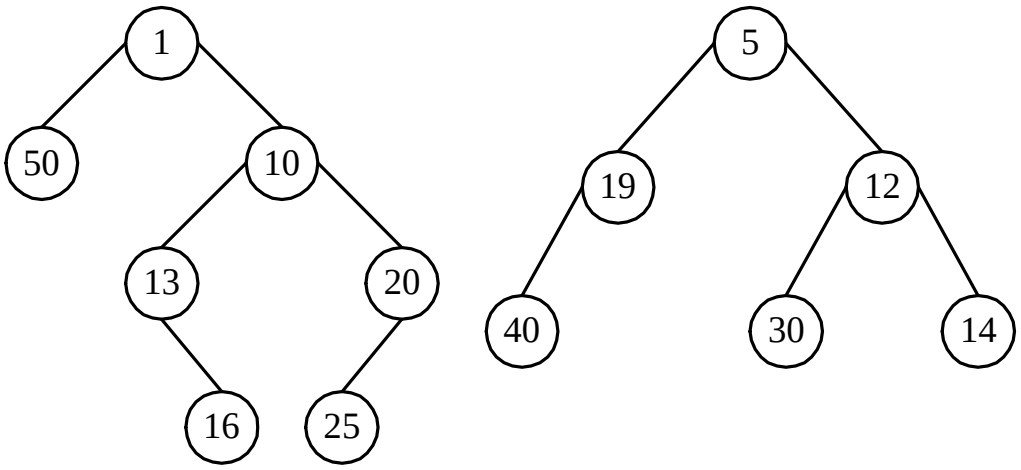
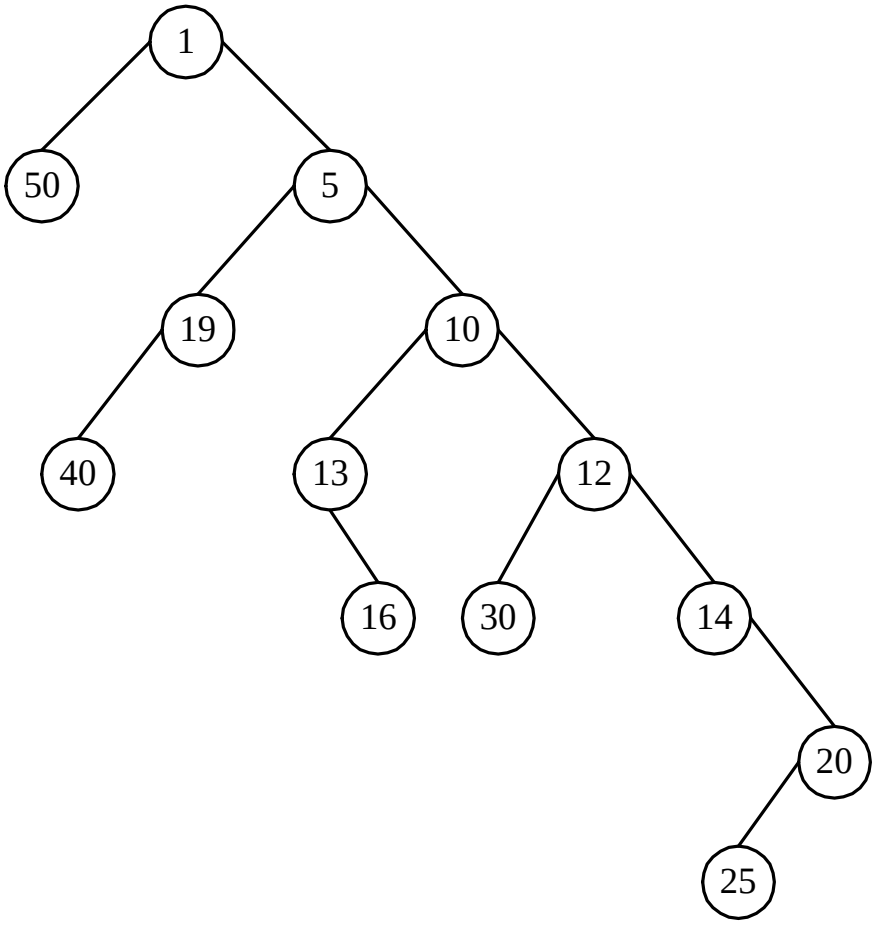
- An amortized analysis of skew heaps
  two skew heaps:

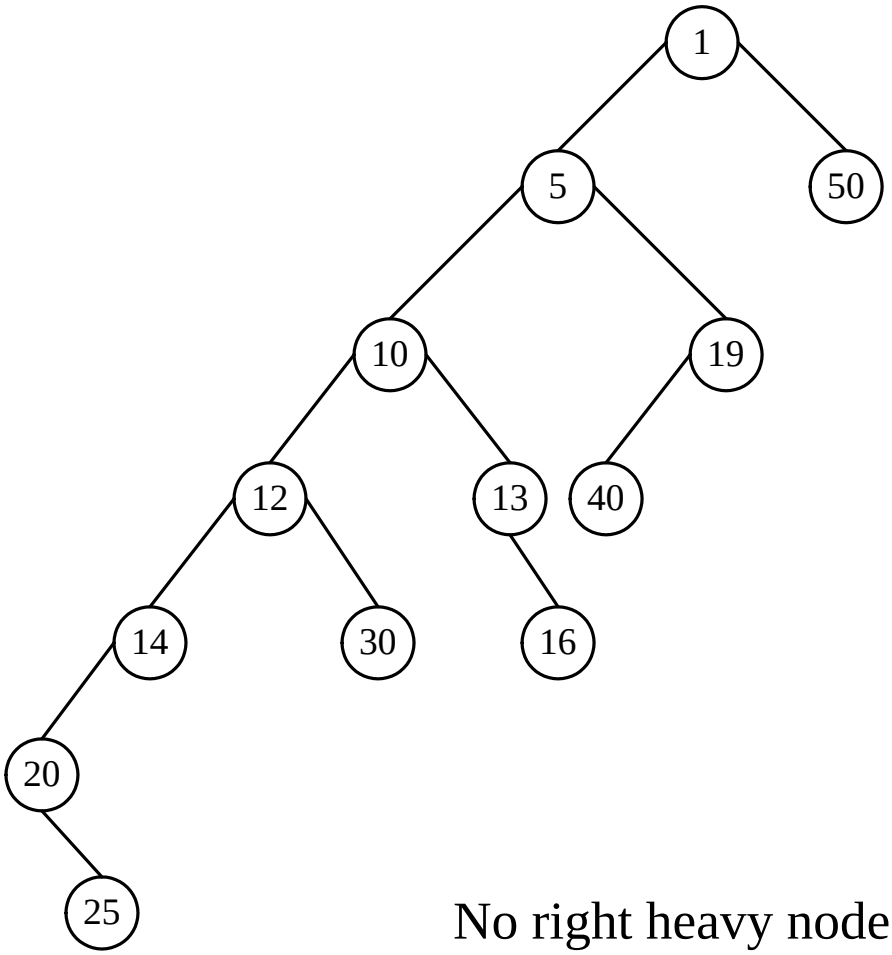


⇓ merge of the right paths



5 right heavy nodes.

⇓ swapping of children along the path formed by the merge.



No right heavy node.

meld: merge + swapping
operations on a skew heap:

1. find-min(h): find the min of a skew heap h.
2. insert(x, h): insert x into a skew heap h.
3. delete-min(h): delete the min from a skew heap h.
4. meld(h_1, h_2): meld two skew heaps h_1 and h_2.

The first three operations can be implemented by melding.

potential function

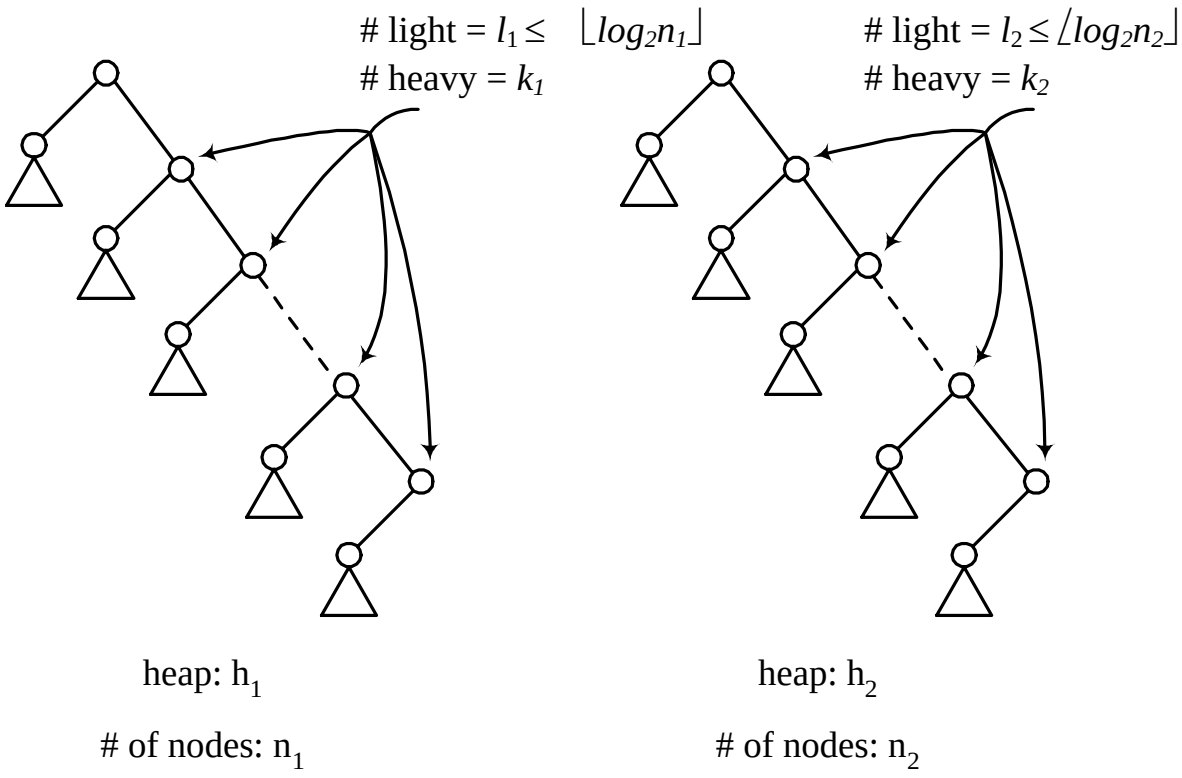
wt(x): # of descendants of node x, including x.

heavy node x: wt(x) > wt(p(x))/2, where p(x) is the parent node of x.

light node: not a heavy node

potential function Φ_i: # of right heavy nodes of the skew heap.

- amortized time



NOTE: heavy → light light → heavy/light

$a_i = t_i + \Phi_i - \Phi_{i-1}$

t_i : time spent by OP_i

$t_i = l_1 + l_2 + k_1 + k_2$

$\Phi_i - \Phi_{i-1} \le l_1 + l_2 - (k_1 + k_2)$

$a_i = t_i + \Phi_i - \Phi_{i-1}$

$\le 2 (l_1 + l_2)$

$\Rightarrow a_i = O(\log_2 n)$, where $n = n_1 + n_2$