lecture 12 - Sep 30

Theorem 7 A leftist tree with q nodes on the rightmost path has at least $2^q - 1$ nodes.

Corollary 4. The rightmost path in a leftist tree of N nodes has at most

$$\lfloor \log(N+1) \rfloor$$

nodes.

$N = 2^q - 1$

q $\lfloor \log(N+1) \rfloor$

⇝ operations on rightmost path of a leftist heap are "cheap"!

but: need to restore leftist heap property once violated.

Implementation: store npl with a node!

Leftist heap operations.

merge (heap 1, heap 2) (combines two leftist heaps into a leftist heap)

if heap 1 empty, return heap2;

if heap2 empty, return heap1;

if root(heap1) < root(heap2) return merge1 (heap1, heap2)

else return merge1 (heap2, heap1)

merge 1 (heap1, heap2)

[note: the root of heap1 is not larger than the root of heap 2]

if heap1.left empty, append heap2 to the left of heap1.

else

{

append merge (heap1.right, heap2) to the right of heap1; (a)

update npl's

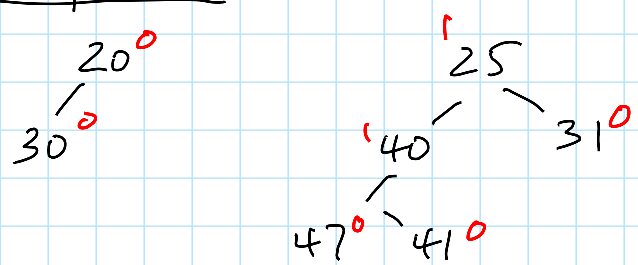
if npl(heap1.left) < npl(heap1.right)

(b) {swap children of heap 1}

npl(heap1) = 1 + npl(heap1.right)

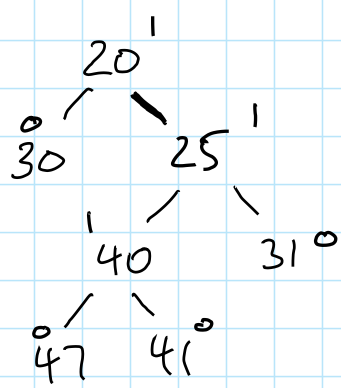
}

Example 23.

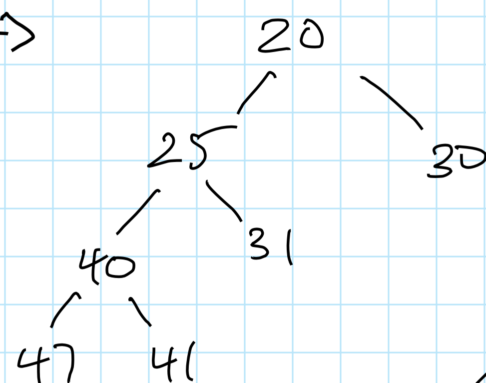


merge?

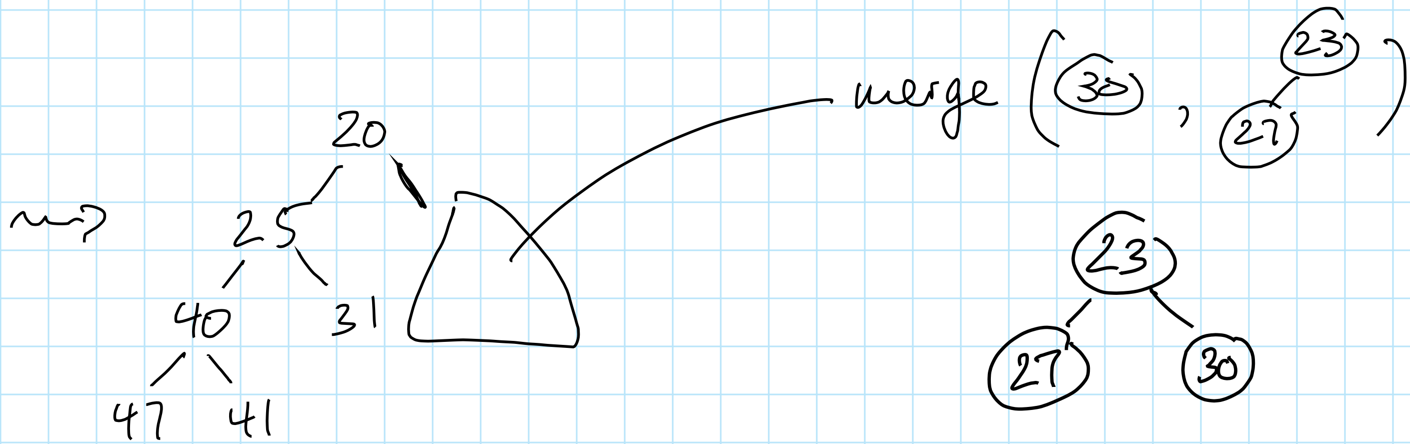
(a) →



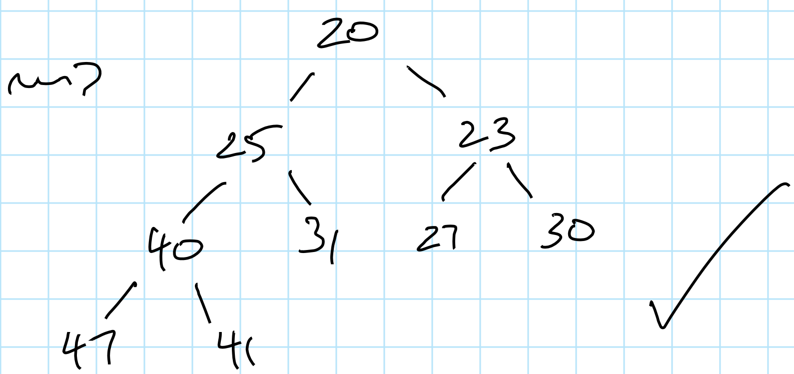
(b) →



J



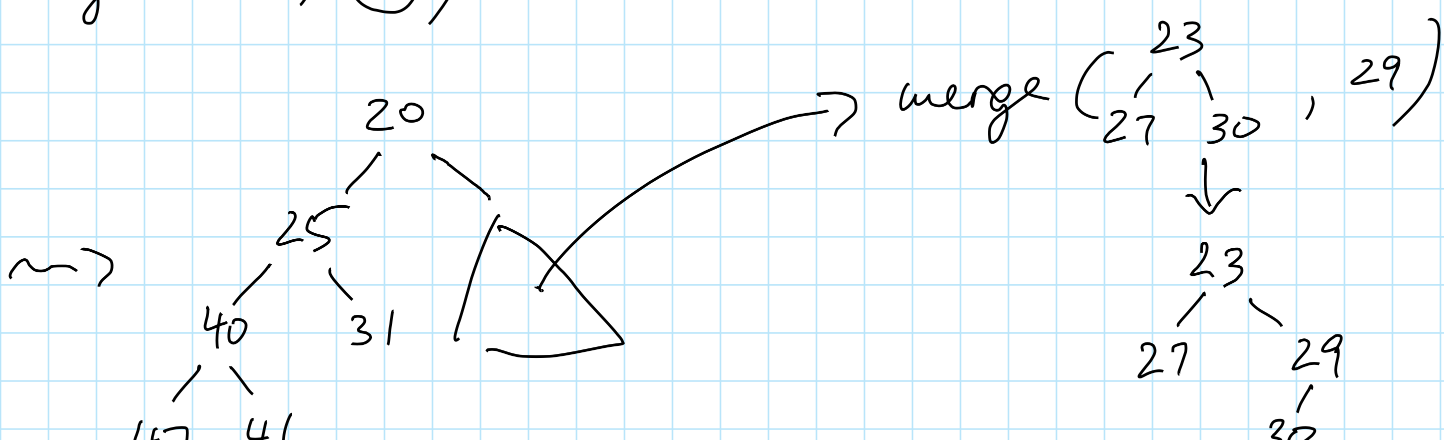
merge
20
25
40
31
47
41
30
23
27



20
25
23
40
31
27
30
47
41

J'

merge (J', 29)



20
25
40
31
47
41
merge
23
27
30
29

analysis

- at each node visited in rec. calls, constant time needed
- visit only nodes on right path of leftist heap ($O(\log(N))$ many)

$\rightsquigarrow T_{worst}(N) = O(\log(N))$.

insertion : merge of a leftist heap with another leftist heap, of size 1

$\rightsquigarrow T_{worst}(N) = O(\log(N))$

deleteMin : destroy root ($O(1)$) and merge resulting leftist heaps (subtrees of the original root) ($O(\log(N))$)

$\rightsquigarrow T_{worst}(N) = O(\log(N))$

2.5 SKEW HEAPS

version of leftist heap that requires no npl information

$T_{worst}(N) = \Theta(N)$ for all ops, but amortized time $O(\log(N))$ per op.

(cf. AVL tree $\iff$ splay tree
to
leftist heap $\iff$ skew heap

merge : as for leftist heaps, but

(b) **always** swap children of heap1

except for the case that root is the maximal node in rightmost path (empty right child)

<u>NOTE</u> · for a single op., a right path might be long!

· no space for npl information needed

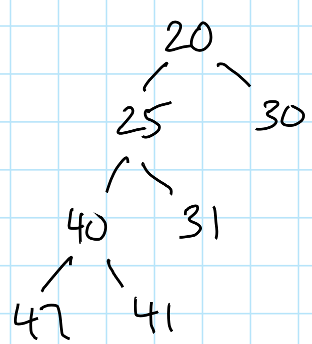
· no tests for npl needed for swaps

Open problem T_{avg} for ops on both leftist and skew heaps.

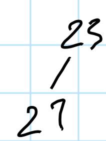
Example for skew heaps
(will be explained next class):

<u>Example 24.</u>

merge



with



⇝