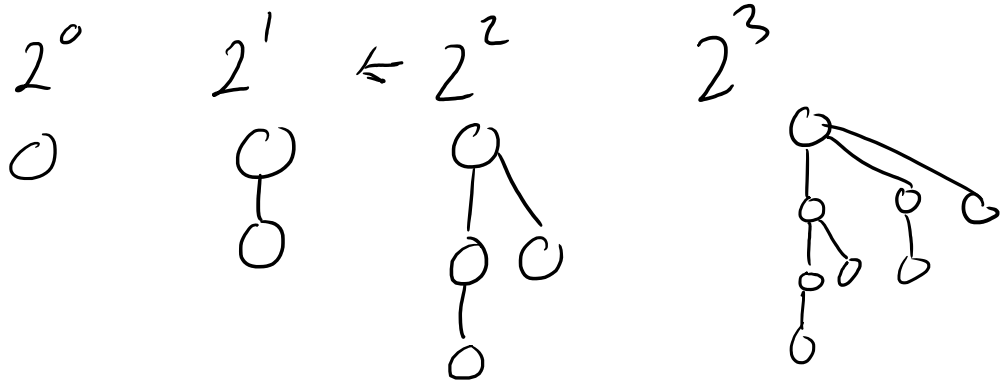
2018-10-04 Binomial & Skew Heaps

Thursday, October 4, 2018 3:01 PM

Binomial Heaps

- Are "forests" of mini heap trees
- Each heap tree is a power of 2 in size and is recursively constructed from prior heap sizes

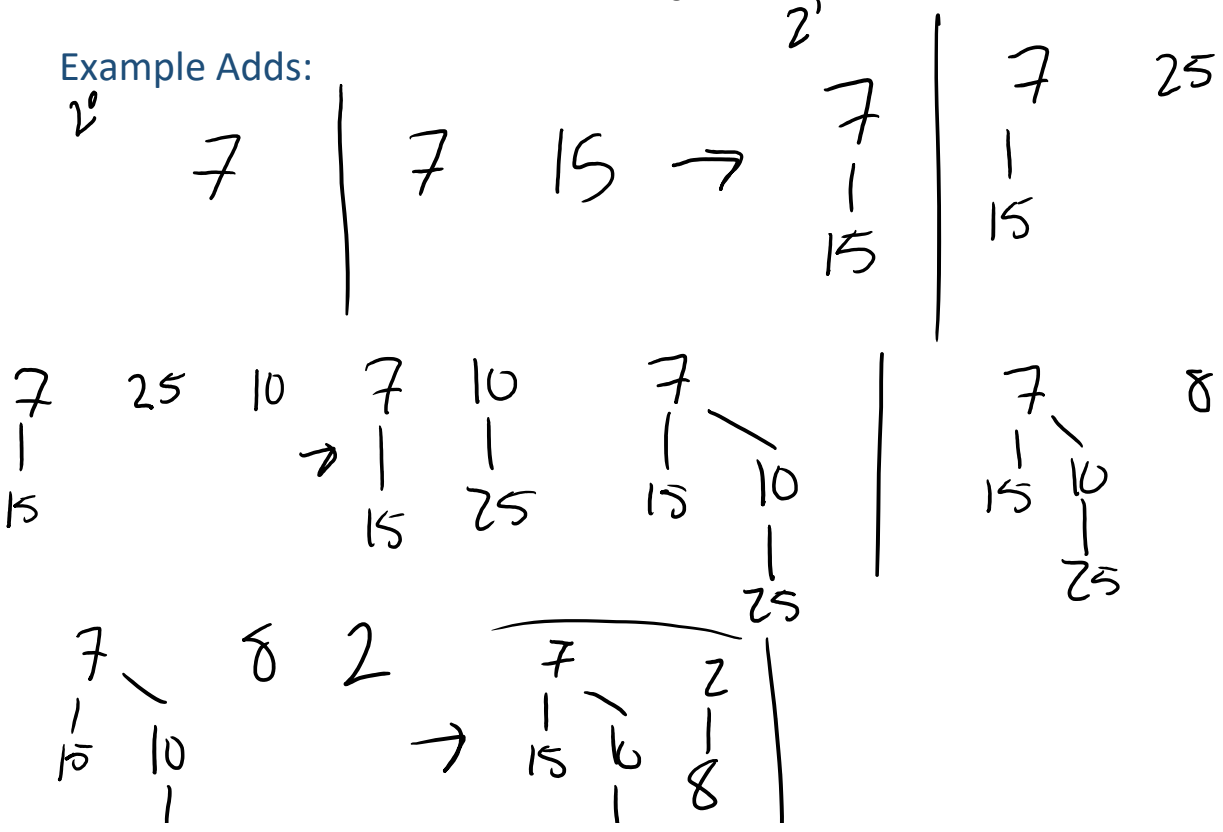


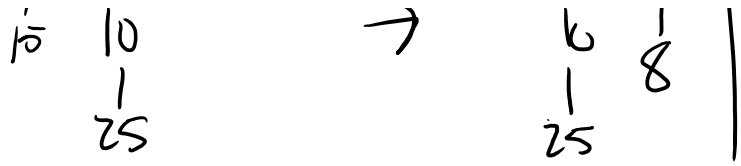
- Thus, a binomial heap has multiple "root" values.

Binomial Heap Rules

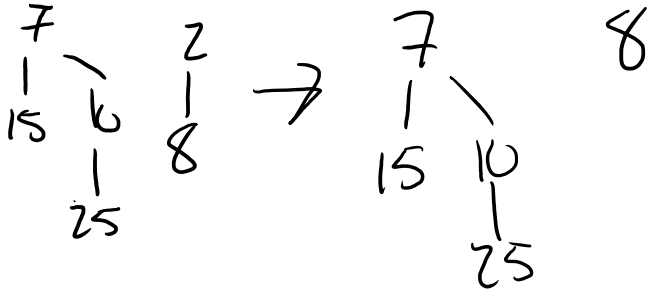
- Each tree in a binomial forest must have a unique size
- All nodes below a given node must be less important than that node
- New values are added to the heap as a singleton (tree size 1)
  - If this causes a violation of rule #1, merge trees until rule is satisfied
- Dequeue removes the most important node in a forest.
  - If this causes a violation of rule #1, merge trees until rule is satisfied

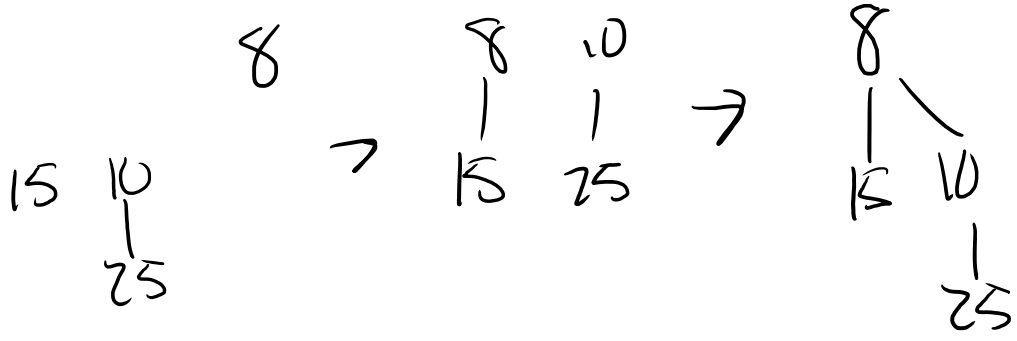
Example Adds:

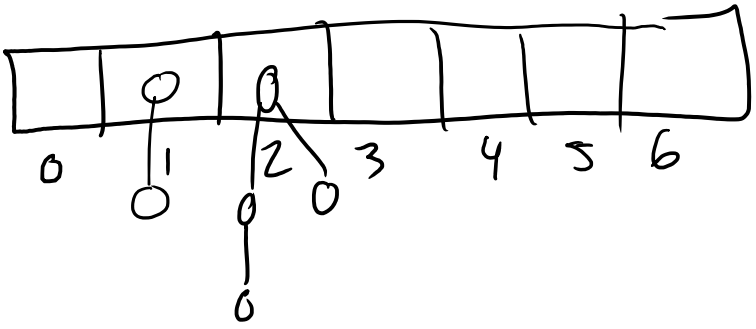


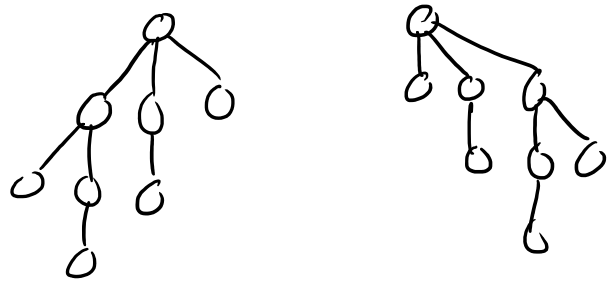


Dequeue



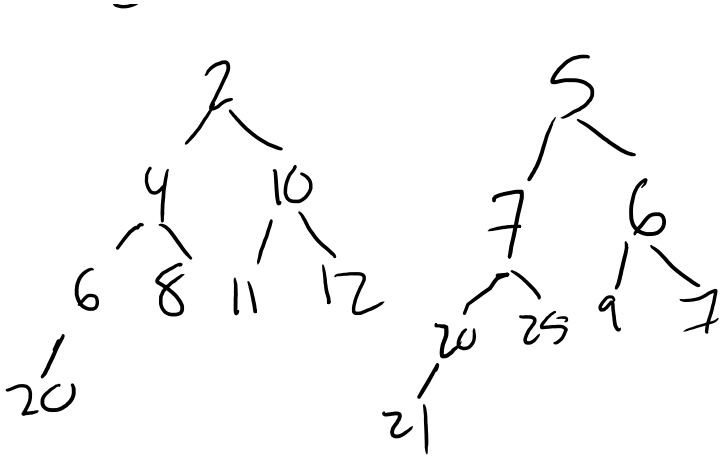






2

5



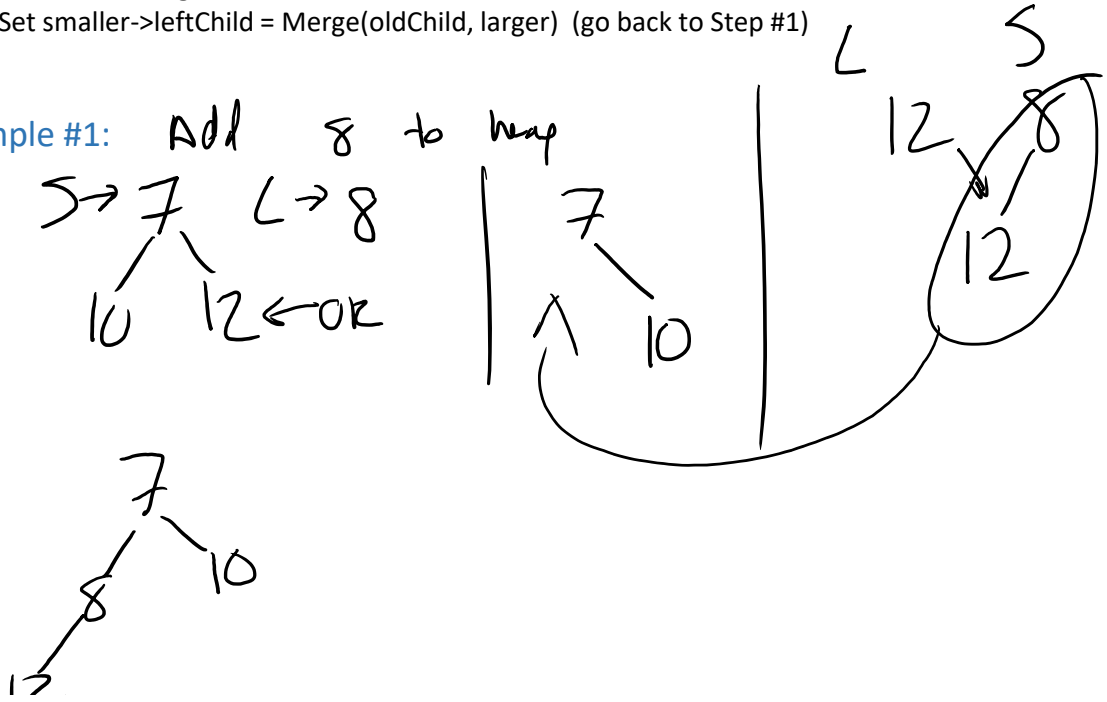
Skew Heaps

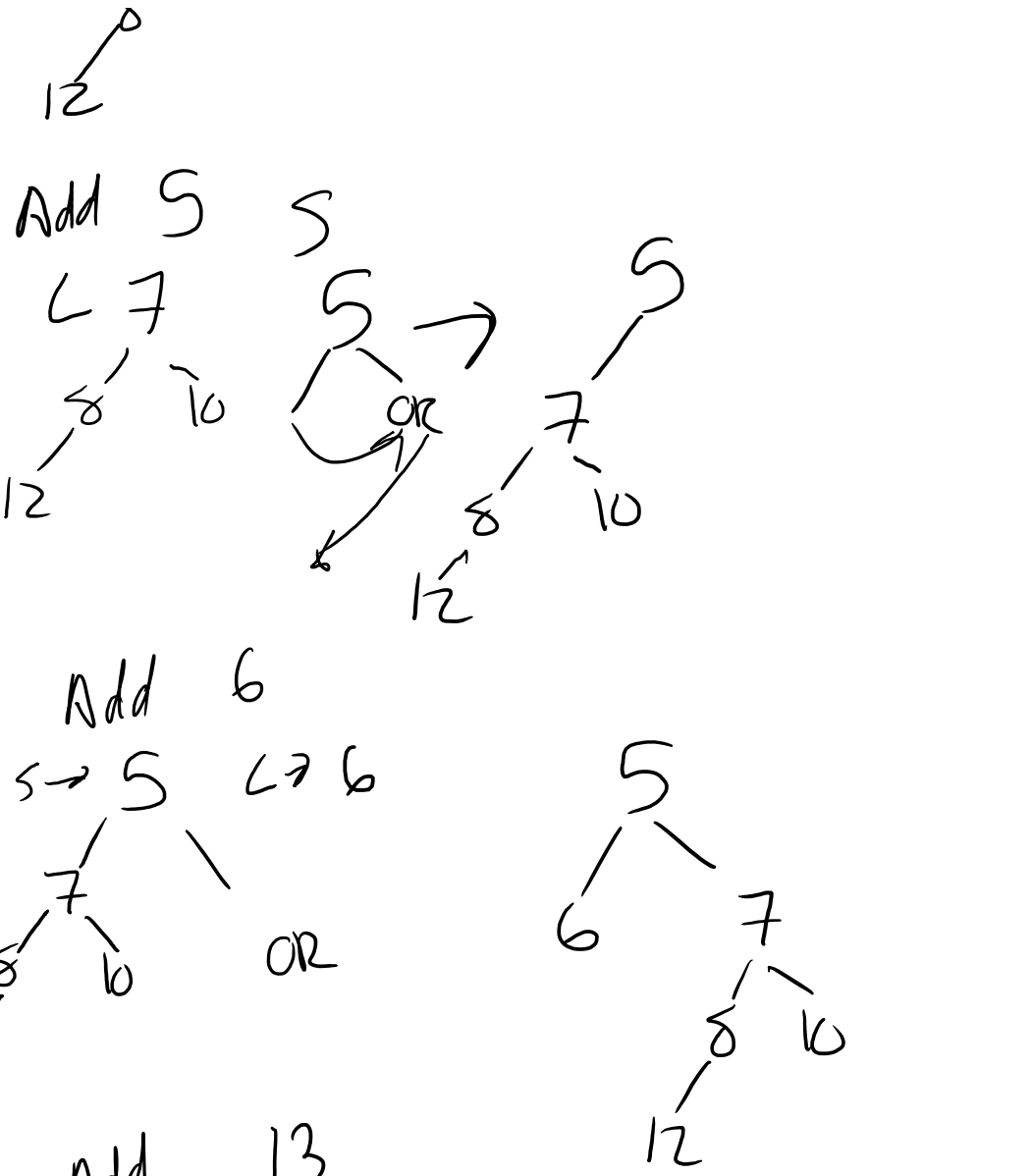
- Recall that binary heaps use vectors. This is required because:
  - We need to get the parent data
  - We need to quickly access bottom-right most element
- What if we wanted to do an LL-based implementation?
  - Different rules are required
- Skew heaps are:
  - LL-based
  - Not guaranteed to be complete
  - Not guaranteed to be balanced
  - Merge better than binary heaps
  - Have on average, LogN runtime
- Similar to binomial heaps, all Skew Heap operations follow the same basic algorithm

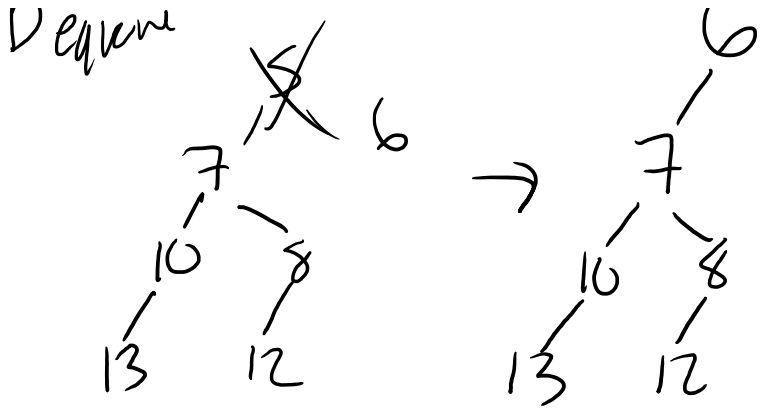
Basic Skew Heap Algorithm (min-heap)

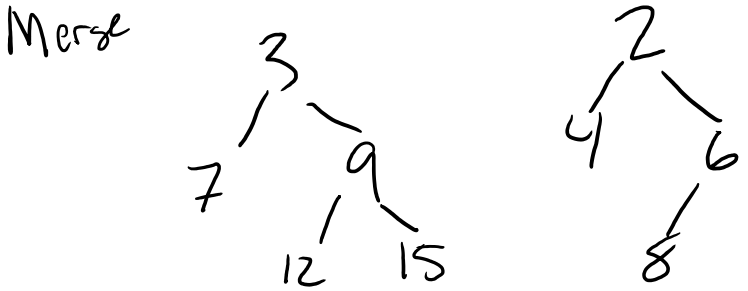
1. Given two heaps, A, B
2. Let smaller = smaller(A, B)
3. Let larger = larger(A, B)
4. Let oldRight = smaller->RightChild
5. Set smaller->RightChild = smaller->LeftChild
6. Set smaller->leftChild = Merge(oldChild, larger) (go back to Step #1)

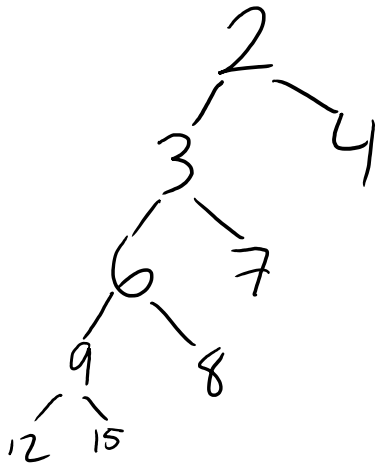
Example #1:



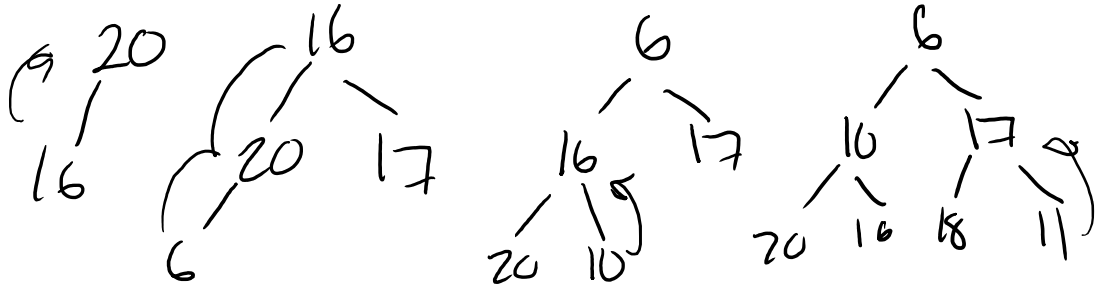

12
Add 5
5
7
8
10
12
5
OR
5
7
8
10
12
Add 6
5
6
7
8
10
OR
12
5
6
7
8
10
12
Add 13
5
5
6
13
7
OR
8
10
12
5
6
7
10
8
13
12
Dequeue
8
6


Dequeue
6
7
10
8
13
12
6
7
10
8
13
12


Merge
3
7
9
12
15
2
4
6
8


2
3
4
6
7
9
8
12
15

Handout


9
20
16
16
20
17
6
6
16
17
20
10
6
10
17
20
16
18
11


6
3
11
8