Priority Queues - Wrapping up Heaps

CptS 223 - Fall 2017 - Aaron Crandall

Today’s Agenda

- Announcements
- Thing of the day
- Queue use examples
- Leftist Heaps

Announcements

- VisuAlgo has added a hash table example: https://visualgo.net/en/hashtable
- I need to pick a target for our next exam. I'll be after sorting.
    - Sorting takes a bit, but it should be before thanksgiving

Glasgow - Coning Duke of Wellington

- Starting in the 1980’s
- City pays £10k/year+ to remove cones
- People love it

Should have been Friday

bit.ly/2ymMXlE

Questions about heaps / priority queues?

- Operations?
- Code?
- Design?
- Big-O?
- Merging?
- Using?

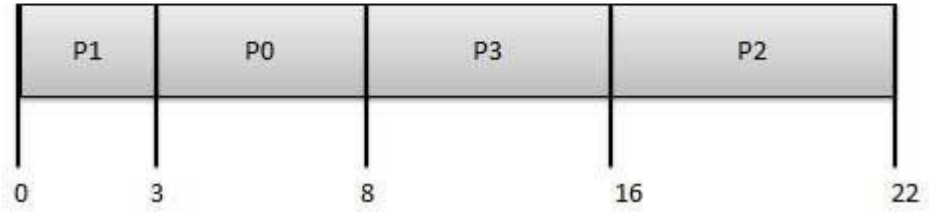
Example Use - Simulation

- Use a tick system to model customers and tellers
- Model for customer entries (Poisson Distribution)
- Model for waiting queue (it’s a queue!)
- Model for teller service time (Distribution)
- Model for current queue in tellers
- Model for exits (just remove customer)
- Instead of calculating at each tick, you can fast forward to the next event (customer entry or teller service finished)

Discussion on Shortest Job First schedule

- Look at list of jobs, pick one that's shortest next
  - Needs a priority queue
- Best to minimize waiting time
  - Common Kernel process scheduler model
  - Does require process profiling
    - Tough in general use

Process	Arrival Time	Execute Time	Service Time
P0	0	5	3
P1	1	3	0
P2	2	8	16
P3	3	6	8



Heaps in the STL - just a reminder

- The STL has a class template for a heap
- It’s called “priority_queue” in header file queue
- Implements a max-heap rather than a min

http://en.cppreference.com/w/cpp/container/priority_queue

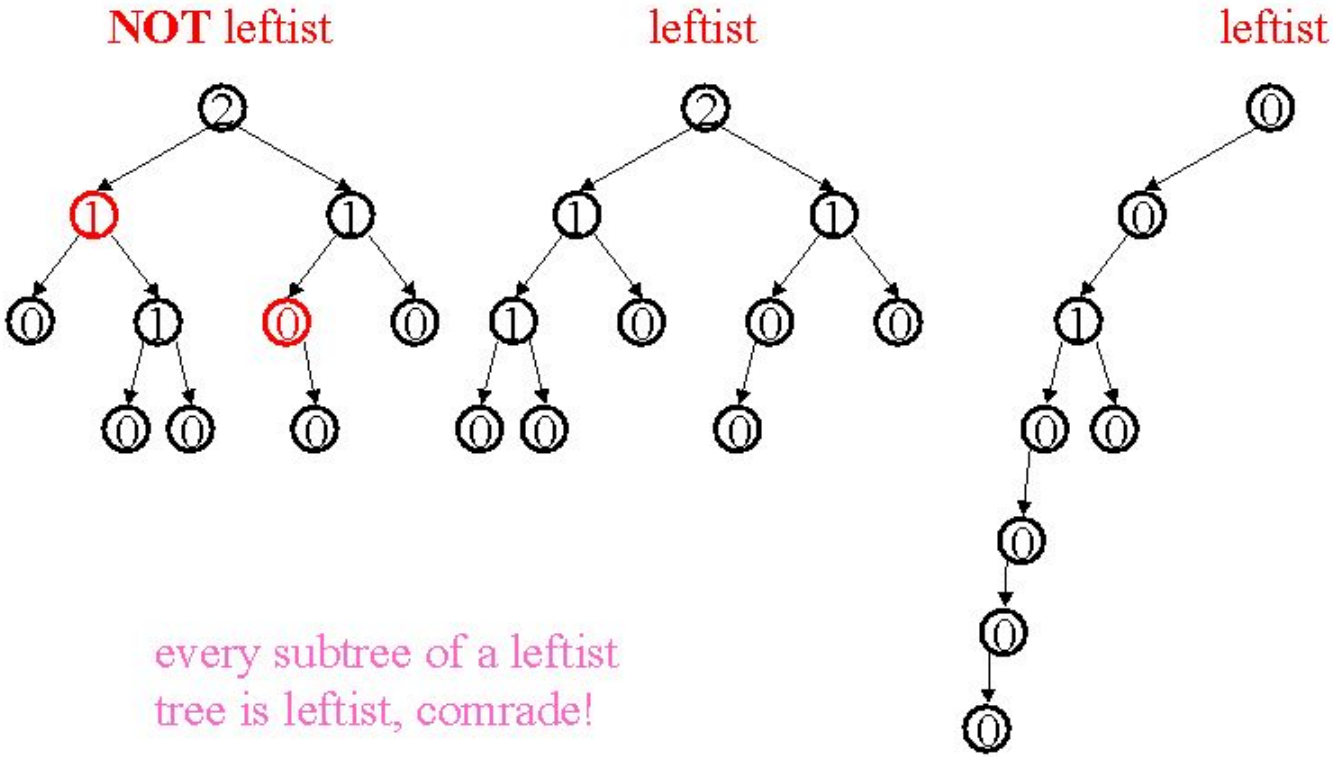
d-Heaps

- d-ary tree instead of a 2-ary (binary) tree
- Makes height log_d (N) instead of log_2 (N)
- Makes inserts: O(log_d N)
- Makes deleteMin: O(d log_d N)
- If d is not a power of 2, it's a penalty on array implementations
  - If power of 2, compiler can use shift register multiplication for HUGE speedups!
  - C++ Syntax: << >> -- Does binary shift of bits in an integer
- Ops are still O(log N), but there's a notable difference in the T(N)
- That said, the compiler will likely translate i*2 or i*4 to i<<1 or i<<2 for you
  - Use + * / - for math and << >> | & for bits

Leftist Heaps

- Adds rule: Left child is height >= right child height
- Height of node is height of shortest child + 1
- Much like AVL tree, it keeps height values in the nodes
- Guarantees left tree is at least the same height as the right, prob more
- Merge/insert are then done only on right trees

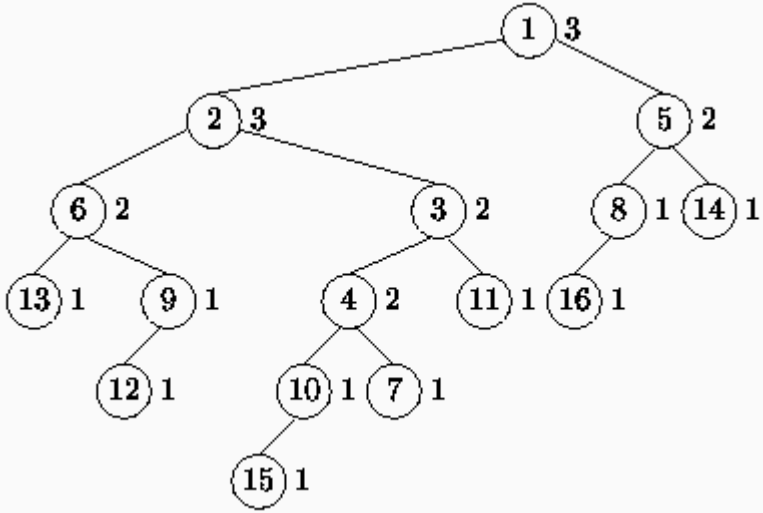
Leftist tree examples



every subtree of a leftist tree is leftist, comrade!

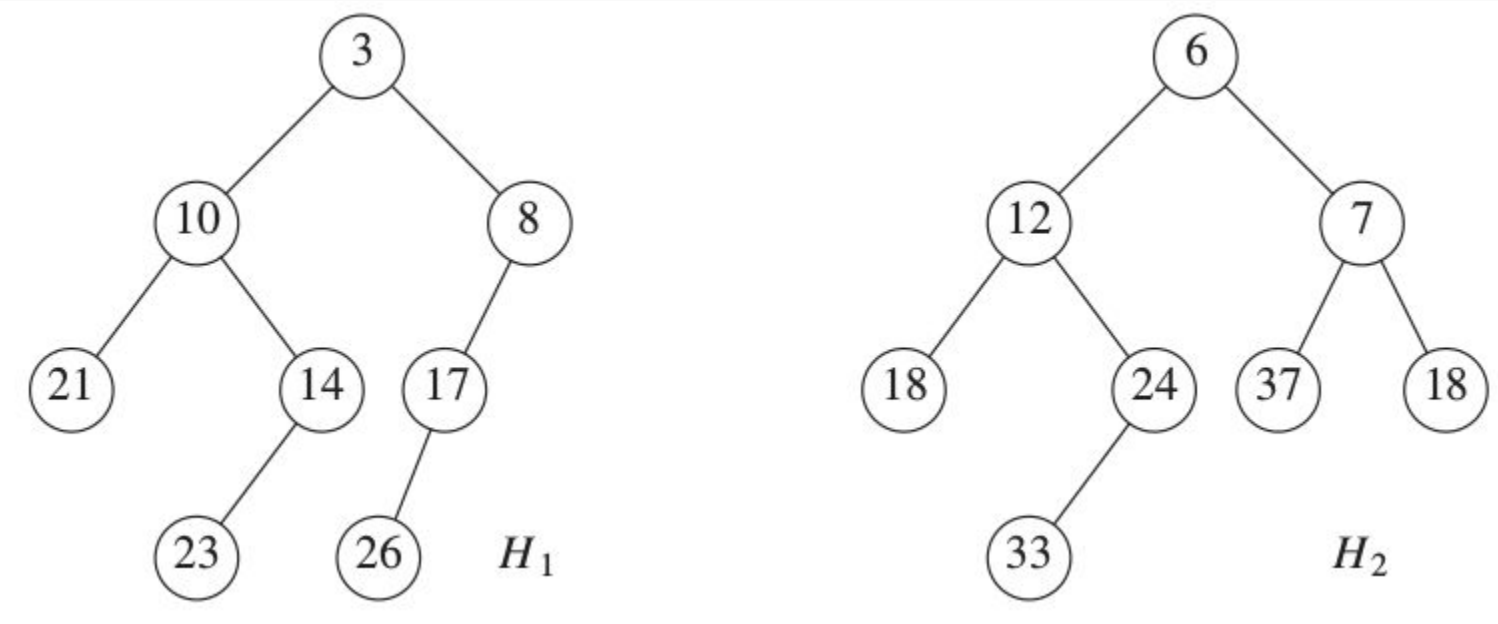
Calculating Heights of Nodes

- Height of a node is height of shorter child + 1
- Also, left tree height >= right tree height
- Biases left!

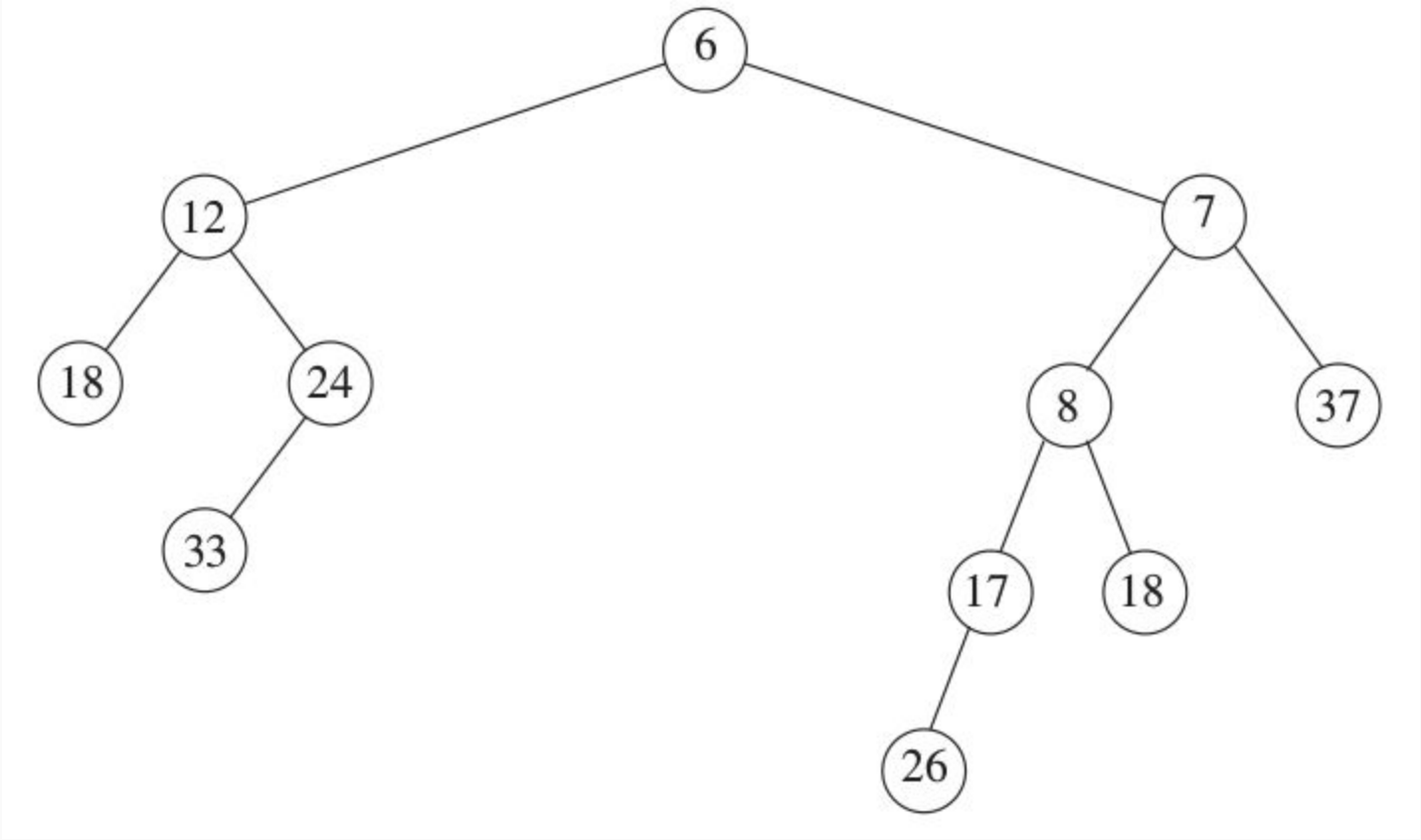


Means right tree is shorter

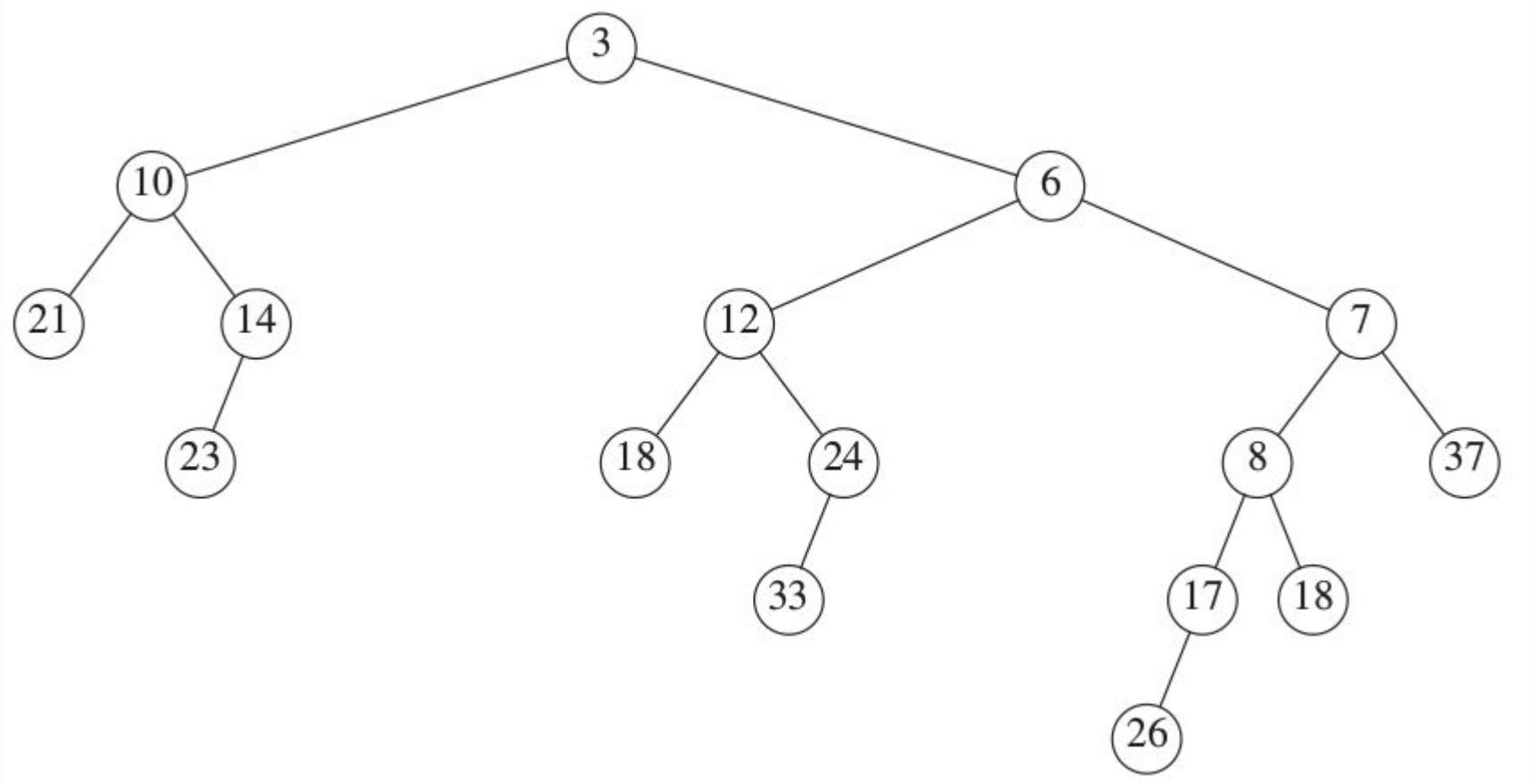
- Right path will have at most log(N+1) nodes
- This guarantees a smaller tree to work with
- Merges (Inserts are merges with a single node tree) are recursively done on right sub tree of the tree with the smaller root
- After the merge, the rest of the first tree is used as the root of a new tree
- Then, heights are re-calculated
  - If a node's children violate the leftist rule, swap them
- No VisuAlgo, but there's this one: https://www.cs.usfca.edu/~galles/JavascriptVisual/LeftistHeap.html



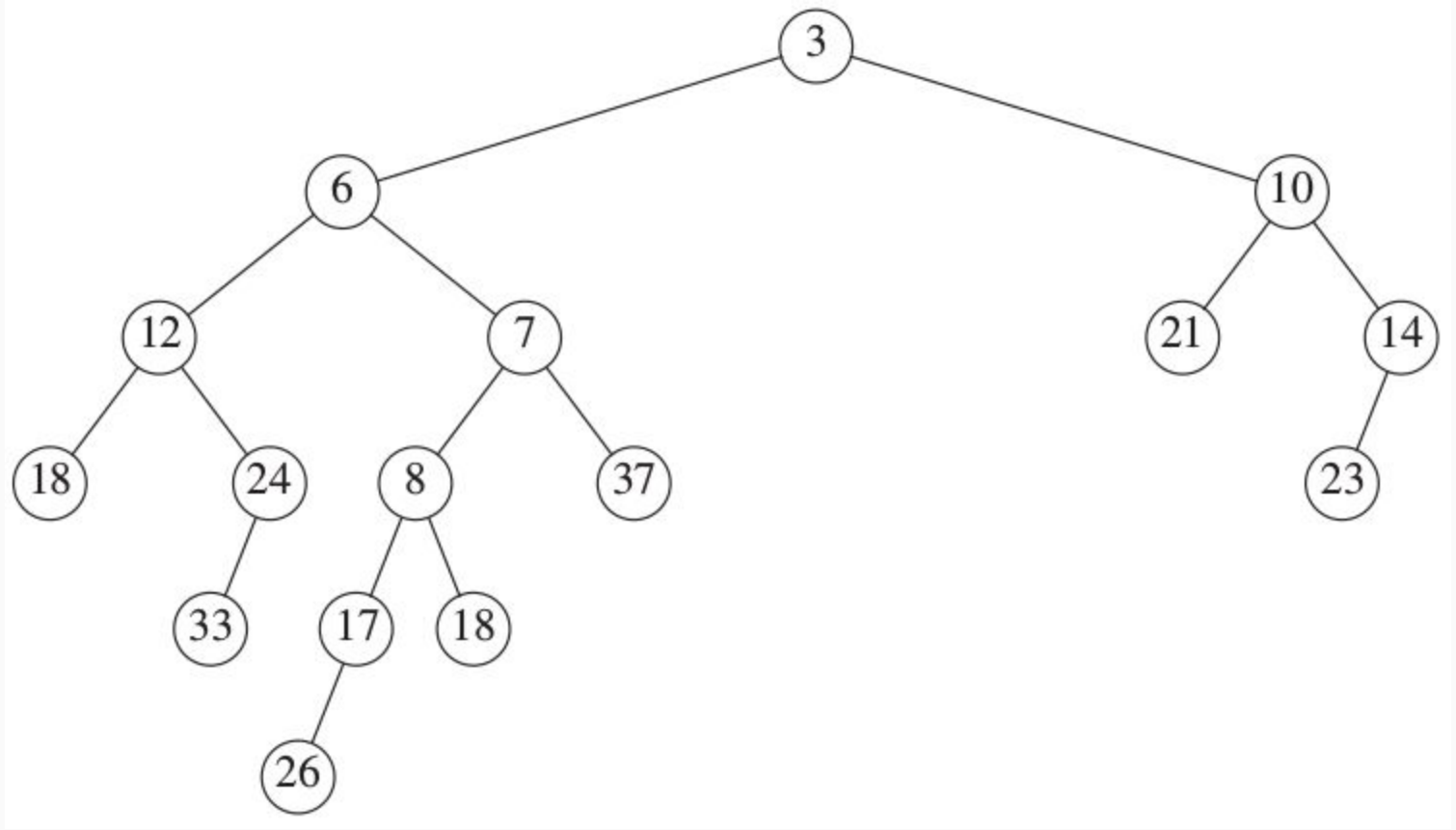
Right subtree of H1 is merged with H2
This is done recursively on each subtree as we go along



Result of top level recursive merger of subtrees
This is the right subtree of Heap 1 with all of Heap 2



Then root and left tree from H1 are made to be the new overall root



To wrap up, the npl values (null pointer lengths) of the nodes are examined and sub trees swapped as needed

Look at the C++ STL priority_queue code

- I’ve got an example somewhere.

There’s an MA4 for you about heaps

- It’s on binary heaps and handling percolations
- You only need to implement percolate up and down