

What's a binary tree?

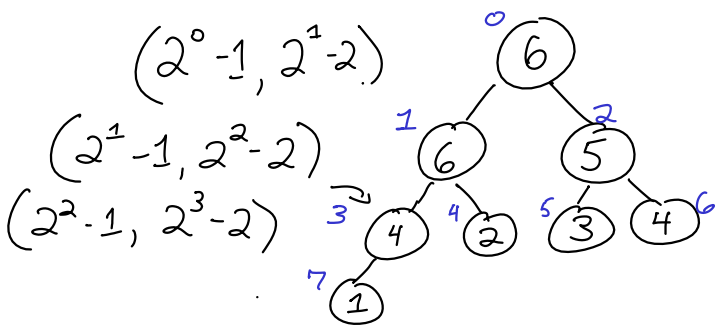
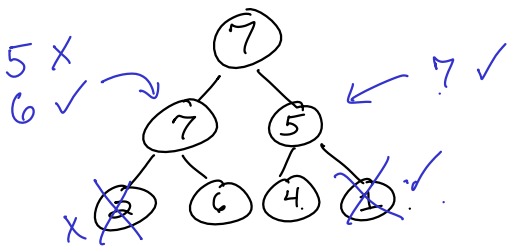
Data structure (tree) w/ at most 1 left child & 1 right child

What's a max heap?

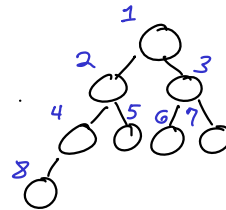
A heap w/ root containing largest priority

Heap is a complete binary tree w/ all descendants ordered in one direction

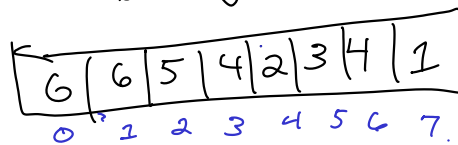
Max heap is a heap w/ all descendants smaller (or equal)



$i/2$



↓ ↓



$$\frac{i+1}{2} - 1$$

Function getLeft(i)

Return $(i+1) * 2 - 1$

$2 * i + 1$

array tree stored in a field called "contents"

Method bubbleUp(i): $O(h) = O(\log n)$

If $i == 0$: Return

$P \leftarrow \text{getParent}(i)$ $O(1)$

If $\text{contents.get}(P) < \text{contents.get}(i)$:

swap $\text{contents}(P)$ w/ $\text{contents}(i)$

bubbleUp(P)

End If

End Method

Method insert(P priority, V value):

$\text{contents.insertAtTail}(\text{pair} \langle P, V \rangle (\text{priority, value}))$

bubbleUp($\text{contents.size}() - 1$) $O(\log n)$

amortized $O(1)$

End Method

Method remove():

swap $\text{contents}[0]$ w/ $\text{contents}[\text{last index}]$

result $\leftarrow \text{contents.removeFromTail}()$

bubbleDown(0)

End Method

Method bubbleDown(i):

$l \leftarrow \text{getLeft}(i)$

$r \leftarrow \text{getRight}(i)$

If $l \geq \text{contents.size}()$:

Return

End If

If $r < \text{contents.size}()$ and $\text{contents}[l] < \text{contents}[r]$ and $\text{contents}[i] < \text{contents}[r]$:

swap $\text{contents}[i]$ w/ $\text{contents}[r]$

bubbleDown(r)

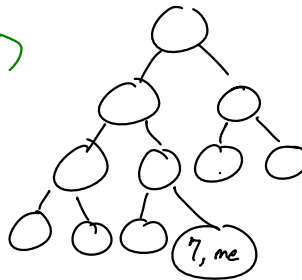
Else If $\text{contents}[i] < \text{contents}[l]$:

swap $\text{contents}[i]$ w/ $\text{contents}[l]$

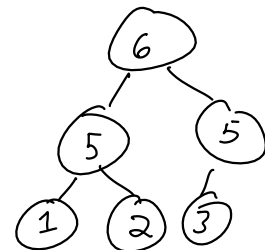
bubbleDown(l)

End If

End Method



insert("7, "me")



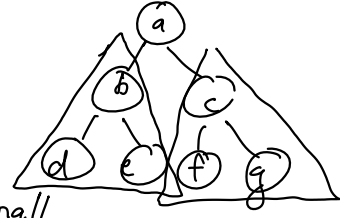
Function heapify(list):

For n from list.size()-1 down to 0:

 bubbleDown(n)

EndFor

EndFunction



In heapify, most bubbleDowns are small.

$$\frac{1}{2}n \cdot 0 + \frac{1}{4}n \cdot 1 + \frac{1}{8}n \cdot 2 + \dots = \sum_{i=1}^{\log n} \frac{1}{2^i} n \cdot (i-1)$$

Heapify $O(n)$

$$= n \sum_{i=1}^{\log n} \frac{1}{2^i} (i-1) < n$$