

Algorithm Analysis

Reagan Shirk

September 24, 2020

Heap

- Max heap is where your parent is always greater than or equal to your child
 - The left child of a node i is $2i$
 - The right child of a node i is $2i + 1$
 - The parent of a node i is $\lfloor \frac{i}{2} \rfloor$
- Max heapify algorithm has a time complexity of $O(\lg(n))$ and a space complexity of $\Theta(1)$
 - Assume that two subtrees are max heaps
 - When you need to move a parent node and its children have equal values, switch it with the left child

Build-Max-Heap

- Space complexity: $\Theta(1)$
- Time complexity:
 - Not Precise: $O(n \lg(n))$
 - Precise: $O(n)$
 - Why is the first one not precise? You're assuming that the height is $\lg(n)$ but that can be improved
 - The height of a node is the longest simple path from it to a leaf
- The number of leaves is $\lceil \frac{n}{2^{h+1}} \rceil$ where h is the height of the node
- The height of the root node is $\lfloor \lg(n) \rfloor$
- I'm ngl guys I haven't been paying attention, I'm so hungry and tired and I'm not feeling the whole class thing. I'm just here for the quiz

Heap sort

- Time complexity is $\Theta(n) + \Theta(n \lg(n)) = \Theta(n \lg(n))$
- What is the problem with heap sort? The data moves down