# Homeworks Heap

Plasencia Palacios Milton Nicolas

May 14, 2020

1) Some stuff about the time insert vs build heap.

6.1-7) To show that the leave nodes are indexed using $\lfloor \frac{n}{2} \rfloor + 1, \lfloor \frac{n}{2} \rfloor + 2, \ldots n$ we must prove that those nodes does not have a left or right child, so, given an index $i$ such that $i \in [\lfloor \frac{n}{2} \rfloor + 1, \lfloor \frac{n}{2} \rfloor + 2, \ldots n]$, by taking $i = \lfloor \frac{n}{2} \rfloor + 1$, we can write its left child as $2i = 2\lfloor \frac{n}{2} \rfloor + 2 > n$, but it cannot be because our heap is made up by at most n nodes. So the other values in the interval, which are greater than $\lfloor \frac{n}{2} \rfloor + 1$, cannot have a left child (and also a right child), then those are leave nodes.

6.2-6) Considering a max-heap in which the elements $i \in \{1, 2, \cdots 100\}$ (so the repetitions are not allowed), and we have the heap build sucht that $i = 1$ is the root. By applying heapify we must call it at every level of the heap, which has height $\log n$, so the worst case will have time complexity $\Omega(\log n)$.

6.3-3) Given a n nodes Binary-Heap we can say that the leave nodes are $\lceil \frac{n}{2} \rceil = \lceil \frac{n}{2^{0+1}} \rceil$. We can prove the thesis by induction, so we assume that, at the $h-1$ level, the equation holds. At the $h-1$ level the number of nodes is at most$\lceil \frac{n}{2^h} \rceil$ so, knowing that we are in a binary-heap to compute the number of nodes at the $h-th$ level we may: $\lceil \frac{n}{2 * 2^h} \rceil = \lceil \frac{n}{2^{h+1}} \rceil$ at most.