Binary Heaps: homework

Eros Fabrici

Course of AA 2019-2020

Ex. 6.1-7

Show that, with the array representation, the leaves of a binary heap containing n nodes are indexed by $\lfloor n/2 \rfloor + 1$, $\lfloor n/2 \rfloor + 2$, $...n$.

Proof

Let's consider the element indexed by $\lfloor n/2 \rfloor + 1$. Then

$$\begin{aligned} LEFT(\lfloor n/2 \rfloor + 1) &= 2(\lfloor n/2 \rfloor + 1) \\ &> 2(n/2 - 1) + 2 \\ &= n \end{aligned}$$

This shows clearly that $LEFT(\lfloor n/2 \rfloor + 1)$ is greater that the number of elements in the heap, therefore it has no children thus it is a leaf. It is trivial that what previously proved holds also for all the elements indexed by $\{\lfloor n/2 \rfloor + 2, \lfloor n/2 \rfloor + 3, ..., n\}$.

Now, to complete the proof, let's show that $\lfloor n/2 \rfloor$ is not a leaf. if n is even, then it is trivial that $LEFT(\lfloor n/2 \rfloor) = 2(n/2) = n$ therefore it has a left children, while if n is odd then it will be equal to $n - 1$.

Ex. 6.2-6

Show that the worst-case running time of HEAPIFY on a binary heap of size n is $\Omega(logn)$ (Hint: For a heap with n nodes, give node values that cause HEAPIFY to be called recursively at every node on a simple path from the root down to a leaf).

Proof

Let A be a MAX-HEAP and let $A[1] = 1 \ \wedge \ A[i] = 2 \ \forall i \in \{2, 3, ..., n\}$. As one is the smallest element in the heap, it has to be brought to the leaf level by swapping through each level of the heap. As the height of a heap is $\lfloor \log n \rfloor$ therefore we have that the complexity in the worst case just described is $\Omega(\log n)$. This holds also for the MIN-HEAP by just setting $A[1] = 2 \ \wedge \ A[i] = 1 \ \forall i \in \{2, 3, ..., n\}$.

Ex. 6.3-3

Show that there are at most $\lceil n/(2^{h+1}) \rceil$ nodes of height h in any n-element binary heap.

Proof

We know for the first exercise that the leaves of a heap resides in the second half of the array plus the middle element if n is odd. As a consequence the number of leaves is $\lceil n/2 \rceil$. Let n_h be the number of nodes at height h (starting to count from the bottom of the heap, namely n_0 is the number of leaf). We proceed now with an induction proof of n_h.

Base Case

$h = 0$ thus $n_0 = \lceil n/2 \rceil$ and we know it true as we stated above.

Inductive step

We need to prove $n_{h-1} = \lceil n/(2^h) \rceil \implies n_h = \lceil n/(2^{h+1}) \rceil$. If n_{h-1} is even, then each node at height h has exactly two children, therefore $n_h = n_{h-1}/2 = \lfloor n_{h-1}/2 \rfloor$. Otherwise, nodes at height h will have 2 children except for one that will have a single child, which implies $n_h = \lfloor n_{h-1}/2 \rfloor + 1 = \lceil n_{h-1}/2 \rceil$. Thus, we have

$$
\begin{aligned}
n_h &= \lceil \frac{n_{h-1}}{2} \rceil \\
&\leq \lceil \frac{1}{2} \lceil \frac{n}{2^h} \rceil \rceil \\
&= \lceil \frac{n}{2^{h+1}} \rceil
\end{aligned}
$$

which completes our proof.