Homework 02: Binary Heaps

Roberto Corti

April 19, 2020

Exercise 1,2,3

Exercise 4

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

If we consider the node at the index $\lceil n/2 \rceil + 1$ and we know that its left child will be at the index

$$\texttt{LEFT}(\lceil n/2 \rceil + 1) = 2(\lceil n/2 \rceil + 1) > 2(n/2 - 1) = n.$$

The left child is in a position greater than `HEAP_SIZE`, so this node has no children and since a heap is a complete tree up to the leaf level, we conclude that the node $\lceil n/2 \rceil + 1$ is a leaf.

Exercise 5

Show that the worst-case running time of `MAX_HEAPIFY` on a heap of size n is $\Omega(\log_2 n)$. (Hint: For a heap with n nodes, give node values that cause `MAX_HEAPIFY` to be called recursively at every node on a simple path from the root down to a leaf).

Consider the following max-heap in which we would like to represent the worst-case scenario:

$$A[1] = 1, \text{ and } \; A[i] = 0 \;\; \forall i \in 2, ..., n.$$

Then, in order to conserve the heap property we have to apply `MAX_HEAPIFY` for every level of the tree for pushing the element of value 0 to the leftmost leaf. `MAX_HEAPIFY` will be then called $\log_2 n$ times, so its running time will be $\Theta(\log_2 n)$. Thus, the worst case running time of `MAX_HEAPIFY` is $\Omega(\log_2 n)$.

Exercise 6

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

We know that for any $n > 0$, the number of leaves of nearly complete binary tree is $\lceil n/2 \rceil$. With this result we can prove the thesis by induction.

- Consider the case $h = 0$; the thesis is satisfied since the unique level of the tree will contain $\lceil n/2 \rceil = \lceil n/2^{0+1} \rceil$.
- Consider the thesis true for $h - 1$.
- Let N_h the number of nodes at leaf level of the tree T_h that has height h. If we know consider the tree T_{h-1} which is composed by T_h up to the leaves of T_h. This tree will have $n' = n - \lceil n/2 \rceil$ nodes. Thus, if we consider N'_{h-1} as the number of nodes at level $h - 1$, this will be equal to N_h. By induction we have

$$N'_{h-1} = N_h = \lceil n'/2^h \rceil = \lceil \lceil n/2 \rceil / 2^h \rceil \leq \lceil n/2^{h+1} \rceil.$$