6-1. Building a heap using insertion

We can build a heap by repeatedly calling MAX-HEAP-INSERT to insert the elements into the heap. Consider the following variation on the BUILD-MAX-HEAP procedure:

```
BUILD-MAX-HEAP′(A)
    A.heap-size = 1
    for i = 2 to A.length
        MAX-HEAP-INSERT(A, A[i])
```

a. Do the procedures BUILD-MAX-HEAP and BUILD-MAX-HEAP′ always create the same heap when run on the same input array? Prove that they do, or provide a counterexample.

b. Show that in the worst case, BUILD-MAX-HEAP′ requires $\Theta(n \lg n)$ time to build an n-element heap.

Answer.

a. No they don't. As a counterexample, Figure 1 shows a max-heap BUILD-MAX-HEAP builds on

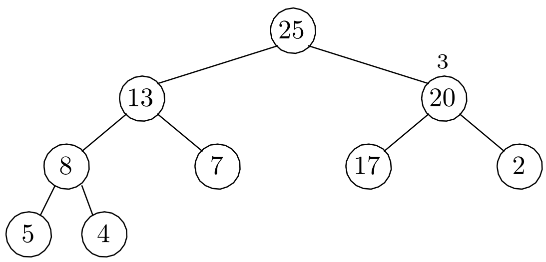

Figure 1. The max-heap BUILD-MAX-HEAP builds on array $A = \langle 5, 13, 2, 25, 7, 27, 20, 8, 4\rangle$.

array $A = \langle 5, 13, 2, 25, 7, 17, 20, 8, 4\rangle$, while BUILD-MAX-HEAP′ builds a different max-heap on the same array, shown in Figure 2. Notice that children of node of index 3 are different in two figures.

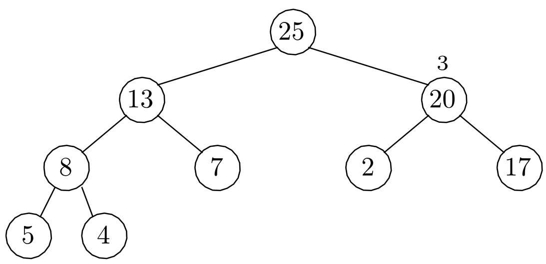

Figure 2. The max-heap BUILD-MAX-HEAP′ builds on array $A = \langle 5, 13, 2, 25, 7, 27, 20, 8, 4\rangle$.

Email address: informlarry@gmail.com

b. In the worst case, BUILD-MAX-HEAP$'$ builds a max-heap from an array of increasing order. Each time a node is inserted into the heap, it trace a path from that node at index i to the root, which has a length of $\Theta(\lg n)$. As the insertion iterates for $n-1$ times, the worst case running time of BUILD-MAX-HEAP$'$ is

$$\begin{aligned} T(n) &= (n-1)\,\Theta(\lg n) \\ &= \Theta(n \lg n) \end{aligned}$$