1. The HEAP-DELETE(A, i) operation deletes an item i from the heap A. Give an implementation of HEAP-DELETE that runs in $O(\log(n))$ time for an n element max-heap. You may describe your implementation in terms of other heap operations.

2. The method we saw in class for building a max-heap worked in a bottom-up manner, that is it started with the leaves of the heap and moved toward the root. We can also implement a version of BUILD-MAX-HEAP in a top-down manner, by starting with an empty heap and repeatedly inserting elements into it. Consider the following psuedo-Python-code:

```
def build-max-heap2(A):
    # A is currently a 1 element heap (as is any array)
    for i in range(1, len(array))
        max-heap-insert(A, A[i])
```

   (a) Does build-max-heap2 always construct the same heap as the bottom-up BUILD-MAX-HEAP procedure we saw in class? Argue that it does or provide a counterexample.

   (b) What is the worst-case running time of build-max-heap2? (HINT: think about the worst possible input.)