CS240, Spring 2022
Assignment 2: Question 1

Q1) Suppose we are working with a heap, represented as an array, and we want to remove an element from it that is not necessarily the root. We are given the index of this element in the array. Describe an algorithm that performs this task and analyse its complexity (since we did not prove correctness of fix-up/fix-down in class, we do not require you to prove correctness).

Given the index i for the element in the heap we want to delete, the very first thing we will want to do is swap it with the last leaf (much like we do for Algorithm 2.6 in the text book).

After we have done the swap we will reduce the size of the heap (effectivly removing this leaf). After this we will do a fix down on this index, the time complexity of this fix down is proportional to the height of the element which is log(n).

Thus since the swap and removal is constant time, it must be the case that this algorithm to remove the element will be log(n) time complexity.