Algorithm 2015 fall Homework 2

1. What is the running time of **HEAPSORT** on an array A of length n that is already sorted in increasing order? What about decreasing order?
2. Argue the correctness of HEAPSORT using the following loop invariant:

   > At the start of each iteration of the **for** loop of lines 2-5, the subarray $A[1 \ldots i]$ is a max-heap containing the i smallest elements of $A[1 \ldots n]$, and the subarray $A[i+1 \ldots n]$ contains the $n-i$ largest elements of $A[1 \ldots n]$, sorted.

3. Show that when all elements are distinct, the best-case running time of HEAPSORT is $\boldsymbol{\Omega}(\mathbf{nlgn})$.
4. Show that the running time of QUICKSORT is $\boldsymbol{\Theta}(\mathbf{n^2})$**,**when the array A contains distinct elements and is sorted in decreasing order.
5. When RANDOMIZED-QUICKSORT runs, how many calls are made to the random number generator RANDOM in the worst case? How about in the best case? Give your answer in terms of Θ-notation.
6. Show that quicksort's best-case running time is $\boldsymbol{\Omega}(\mathbf{nlgn})$.
7. Illustrate the operation of COUNTING-SORT on the array $A=\langle 6,0,2,0,1,3,4,6,1,3,2 \rangle$
8. Prove that COUNTING-SORT is stable.

9. Illustrate the operation of RADIX-SORT on the following list of English words: COW, DOG, SEA,RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX.
10. Illustrate the operation of BUCKET-SORT on the array $A=\langle 0.79,0.13,0.16,0.64,0.39,0.20,0.89,0.53,0.71,0.42\rangle$.