Fundamental Algorithms
Courant, Fall 2017

Homework 5

Instructor: Amir Shpilka
Due Date: Wed Oct 11, 11:55pm

1. Decision trees for comparison-based sorting algorithms.
   (a) (4 points) What is the smallest possible depth of a leaf in a decision tree for a comparison sort?
   (b) (6 points) Show that there is no comparison sort whose running time is linear for at least half of the $n!$ inputs of length n. What about a fraction of $1/n$ of the inputs of length n? What about a fraction $1/2^n$?

2. A sorting algorithm is called *stable* if elements with the same value appear in the output array in the same order as they do in the input array. That is, it breaks ties between two elements by the rule that whichever element appears first in the input array appears first in the output array.
   (a) (5 points) Prove that counting sort is stable.
   (b) (8 points) Which of the following sorting algorithms are stable: insertion sort, merge sort, heapsort, and quicksort? Justify your answers with short proofs or counter-examples.
   (c) (2 points) Give a simple scheme that makes any sorting algorithm stable, by treating the algorithm as a *black-box* (i.e. you can't change the code of the algorithm, but you can add some code that calls the algorithm on whatever input you like). How much additional time and space does your scheme entail?

3. (5 points) Describe an algorithm that, given n integers in the range 0 to k (inclusive), preprocesses its input and then answers any query about how many of the n integers fall into a range $[a \ldots b]$ in $O(1)$ time. Your algorithm should use $\Theta(n + k)$ preprocessing time. Briefly prove correctness and running time.

   Hint: Look at counting sort again.

4. (5 points) Suppose that you have a *black-box* worst-case linear-time median subroutine. Give a simple, linear-time algorithm that solves the selection problem for an arbitrary order statistic. Briefly prove correctness and running time.

5. (5 points) Describe a linear time algorithm which, given an n elements array A and a number $k < n$, returns k elements of A which are closest to the median of A (excluding the median itself). For example, if $A = (10, 5, 11, 1, 6, 7, 25)$ and $k = 2$, the median of A is 7, and 2 closest numbers to 7 are 6 and 5. Briefly prove correctness and running time.

6. (5 points) Suppose you are given two sorted lists A, B of size n and m, respectively. Give an $O(\log k)$ algorithm to find the k-th smallest element in $A \cup B$, where $k \leq \min(m, n)$. Briefly prove correctness and running time.

7. (5 points) (**Bonus**[1]) Show that the second smallest element of an n element (distinct) array can be found with $n + \lceil \log n \rceil - 2$ comparisons in the worst case.

   Hint: Also find the smallest element.

[1] Bonus questions are not mandatory, and not attempting them will not harm you in any way. They are often more difficult (so that you don't complain that the homework isn't challenging enough) and/or teach extra material.