Assignment 5

Algorithms, Spring 2024

Honor code: *Work on this assignment with one partner (highly encouraged) or, if you must, work alone*[1]*. Partner policy: You and your partner will work together on the assignment throughout the whole process, you will write it and review it together, and will submit one assignment. Between different teams, collaboration is at level 1 (verbal collaboration only). It is not allowed to search online for the specific problems in this assignment—doing so violates academic honesty for the class.*

1. The worst-case $O(n)$ SELECT(A, k) algorithm uses the following strategy to find a good pivot: it divides the input elements in groups of 5, finds the median of each group and then finds the median of these medians. The natural question is: what is magic about 5? Will other numbers work?

   In this problem we'll look at what happens if the input is divided into groups of 7 elements instead of 5: the SELECT() algorithm will divide the input in groups of 7 elements, find the median of each group and the median of these medians, and use this median as pivot to partition. Denote the median of medians as x.

   (a) State and prove a lemma on the number of elements in the input that are guaranteed to be $< x$ and $> x$, respectively. Assume all elements in A are distinct.

   (b) Use the lemma above to state the worst-case partition possible with this strategy.

   (c) Write the recurrence for the worst-case running time of this modified version of Select.

   (d) Write the intuitive justification on why this recurrence solves to $O(n)$.

   Note: In general it can be shown that groups of size > 5 lead to a linear time algorithm, and groups of size < 5 do not lead to a linear algorithm. 5 is the smallest size which leads to a linear algorithm.

2. Let A be a list of n (not necessarily distinct) integers. Describe an $O(n)$-algorithm to test whether any item occurs more than $\lceil n/4 \rceil$ times in A.

   (For full points, you may not make any assumption about the input.)

[1] Working with a partner will likely improve your learning—you are highly encouraged to find a partner, and you are welcome to message me and I'll help connect you

3. **New data structure?** You're interviewing for your dream job at an echological ethical tech company with healthy snacks. You already passed 25 stages of interviews, and your final interviewer asks you to design a binary search tree data structure that performs INSERT operations in $O(\sqrt{\lg n})$ time using a comparison-based algorithms. Design such a structure, or prove that it is impossible.

   We expect: Justification that this is impossible.

   Hint: Impossibility results are usually shown by contradicting a known lower bound. The only lower bound we know is comparison-based sorting. So your strategy will be to try to show that if such a BST existed, you could use it to sort faster than $\Theta(n \lg n)$, which would contradict the lower bound for sorting and therefore would be impossible.