

CS 473: Algorithms, Spring 2018

HW 4 (due Wednesday, February 21st at 8pm)

This homework contains three problems. **Read the instructions for submitting homework on the course webpage.**

Collaboration Policy: For this home work, each student can work in a group with up to three members. Only one solution for each group needs to be submitted. Follow the submission instructions carefully.

1. Consider the following variant of Quick Sort. Given an array A of n numbers (which we assume are distinct for simplicity) the algorithm picks a pivot x uniformly at random from A and computes the rank of x . If the rank of x is between $n/4$ and $3n/4$ (call such a pivot a good pivot) it behaves like the normal Quick Sort in partitioning the array A and recursing on both sides. If the rank of x does not satisfy the desired property (the pivot picked is not good) the algorithm simply repeats the process of picking the pivot until it finds a good one. Note that in principle the algorithm may never terminate!
 - Write a formal description of the algorithm.
 - Prove that the expected run time of this algorithm is $O(n \log n)$ on an array of n numbers.
 - **Extra Credit:** Prove that the algorithm terminates in $O(n \log n)$ time with high probability. Does this immediately imply that the expected run time is $O(n \log n)$?
2. Let S be a set of n points in the plane. A point p in S is called Pareto-optimal if no other point in S is both above and to the right of p . Suppose each point in S is chosen independently and uniformly at random from the unit square $[0, 1] \times [0, 1]$. Prove that the number of Pareto-optimal points in S is $O(\log n)$ with high probability.
3. Consider a uniform rooted tree of height h (every leaf is at distance h from the root). The root, as well as any internal node, has 3 children. Each leaf has a boolean value associated with it. Each internal node returns the value returned by the majority of its children. The evaluation problem consists of determining the value of the root; at each step, an algorithm can choose one leaf whose value it wishes to read.
 - (a) Show that for any deterministic algorithm, there is an instance (a set of boolean values for the leaves) that forces it to read all $n = 3^h$ leaves. (*Hint:* Consider an adversary argument, where you provide the algorithm with the minimal amount of information as it request bits from you. In particular, one can devise such an adversary algorithm.).
 - (b) Consider the recursive randomized algorithm that evaluates two subtrees of the root chosen at random. If the values returned disagree, it proceeds to evaluate the third sub-tree. If they agree, it returns the value they agree on.

Write an explicit exact formula for the expected number of leaves being read, for a tree of height $h = 1$, and height $h = 2$.

- (c) Using (b), prove that the expected number of leaves read by the algorithm on any instance is at most $n^{0.9}$.

The remaining problems are for self study. Do *NOT* submit for grading.

- Markov's inequality states that if X is a non-negative random variable then for $t \geq 1$, $\Pr[X \geq t\mathbb{E}[X]] \leq 1/t$. We claim the following corresponding inequality for the lower tail: for non-negative X and $t \geq 1$, $\Pr[X < \mathbb{E}[X]/t] \leq 1/t$. Prove the inequality, or disprove via a counter example.
- See Jeff's home work problems on randomized algorithms from last semester. <https://courses.engr.illinois.edu/cs473/sp2016/hw/hw3.pdf> (Ignore the last problem in this) and <https://courses.engr.illinois.edu/cs473/sp2016/hw/hw4.pdf>
- Jeff's notes have several nice problems including some in the home work.
- Suppose you are presented with a very large set S of real numbers, and you would like to approximate the median of these numbers by sampling. You may assume all the members in S are distinct. Let $n = |S|$. We say that a number x is an ϵ -approximate median of S if at least $(1/2 - \epsilon)n$ numbers in S are less than x , and at least $(1/2 - \epsilon)n$ numbers in S are greater than x . Consider an algorithm that works as follows. You select a subset $S' \subseteq S$ uniformly at random, compute the median of S' , and return this as an approximate median of S . Show that there is an absolute constant c , independent of n , so that if you apply this algorithm with a sample S' of size c , then with probability at least .99, the number returned will be a (.05)-approximate median of S . (You may consider either the version of the algorithm that constructs S' by sampling with replacement, so that an element of S can be selected multiple times, or one without replacement.)
- Consider the following experiment with balls and bins. The experiment proceeds in rounds. In the beginning there are n balls and n bins. At the start of a round each remaining ball is thrown into a bin independently and uniformly into one of the n bins. After the balls are thrown any ball that is *alone* (that is, it is the only ball in its bin) is removed from the experiment. This finishes a round. The remaining balls participate in the next round. The experiment stops when there are no balls remaining after a round.
 - What is the probability that a specific ball i remains after the first round?
 - Prove that the experiment finishes in $c \log n$ rounds with probability at least $1 - 1/n$ for some appropriate choice of constant c . *Hint:* This is similar to the high probability analysis of Quick Sort.
 - **Not to submit:** Also show that the expected number of rounds for the experiment to finish is $O(\log n)$.
 - **Extra Credit:** Prove that the expected number of rounds for the experiment to finish is $O(\log \log n)$.