Advanced Algorithms: Homework 5

Due on Mar. 06, 2024 at 11:59pm EST

Professor Dana Randall Spring 2024

As stated in the syllabus, unauthorized use of previous semester course materials is strictly prohibited in this course.

Exercise 1

Let us roll a fair 6-sided die n times. Give the best upper bound you can find on the probability that the sum of the rolls is at least $5n$.

Exercise 2

We are trying to predict the outcome of an election. Suppose that there are two candidates and the population of N people has exactly $N/3$ supporters of candidate 1 and the rest support the second candidate. If we take a uniform sample of n people from this population to poll (with replacement), what is the probability that the majority of this n prefer candidate 1? How large should n be to guarantee that the probability that candidate 2 wins (in our sample) is at least $4/5$ according to Markov's inequality? Can you get a better bound on n using Chebyshev's inequality? Chernoff bounds?

Exercise 3

Consider the following sorting algorithm for n real numbers chosen independently and uniformly from the range $[0, 1)$. Place each number a_i in one of the buckets $B_1, B_2, \dots B_n$ as follows: a_i goes in bucket B_j if $a_i \in \left[\frac{j-1}{n}, \frac{j}{n}\right)$. Sort each bucket, (using your favorite sorting algorithm) and output the sorted list.

(a) For any integer $0 < M < n$, show that the probability that bucket B_i has at least M entries is at most $1/M$.

(b) Using Chebychev's inequality, give an upper bound on the probability that there exists some bucket with at least M entries.

(c) Can you do better using Chernoff bounds? Give a bound on the probability that there exists some bucket with at least M entries.

Note: Check the restrictions on the Chernoff bound we presented in class carefully. Now observe that $\ln(1+\delta) > \frac{2\delta}{2+\delta}$ for all $\underline{\delta > 0}$. This implies that

$$\delta - (1+\delta)\ln(1+\delta) \leq \frac{-\delta^2}{2+\delta}.$$

Exercise 4

We are given a set of points $\{p_i\}$ on the plane, and we are interested in finding the pair of points that are the farthest apart (called the *diameter*). There is an obvious $O(n^2)$ algorithm, but we want an $O(n \log n)$ algorithm. Prove the following statements and then use them to construct a fast algorithm.

(a) Prove that the pair of points that are farthest apart are both on the convex hull. Thus we can reduce our search to finding the furthest pair of points on a convex polygon: $p_1, p_2, \dots, p_n$, (given in order going around the polygon).

(b) For two points p_i, p_j, we construct the lines l through p_i and l' through p_j so that l, l' are perpendicular to $\overleftrightarrow{p_i, p_j}$. We say that p_i and p_j are *antipodal* if these lines l, l' both do not pass through the convex hull. Show that the pair of points that are the largest distance apart must be antipodal.

(c) Say we are given an edge of the convex hull: $e = (p_1, p_2)$. Give a method to find the point p_i farthest from the line $\overleftrightarrow{p_1, p_2}$. Your method should take $O(i)$ time or faster (not $O(n)$). (Hint: you'll need the slope of $\overleftrightarrow{p_1, p_2}$, and some basic vector knowledge.)

(d) Consider adjacent points p_n, p_1, p_2, and say that the point farthest from line $\overleftrightarrow{p_n, p_1}$ is p_i, and the point farthest from line $\overleftrightarrow{p_1, p_2}$ is p_j. Show that $j \geq i$, and the only possible points that are antipodal to p_1

are $p_i, p_{i+1}, \ldots p_j$. (*Hint:* Why are all other points definitely not antipodal?) Now if p_k is farthest from line $\overleftrightarrow{p_2, p_3}$, what about all possible points antipodal to p_2?

(e) Put it all together: Create an algorithm that takes $O(n \log n)$ time to find the pair of pair of points that are farthest apart, with proof of correctness. Your algorithm should take $O(n)$ steps after finding the convex hull.

Hint: Keep one pointer at p_1, and another at p_i as in part d. Alternate updates of p_i and p_1 in such a way that all antipodal pairs are considered. You can't use part c directly, but the idea should help.