

ECS122A: Algorithm Analysis and Design

Practice Midterm 1

The midterm will be open notes, open book.

These questions are a little harder than the ones that will be on the midterm.

You should also go over all the algorithms and analyses we did in class and make sure you understand the steps, and go over the homework solutions for homeworks 1 and 2 and make sure you understand them (although you would be safe to assume that Pr. 1 from HW 1, and Pr. 4 from HW 2, are both difficult enough that they are unlikely to appear on the midterm...). Know the definition of the expectation of a random variable, a universal hash function, and big-O and big- Ω . Be sure you can do a proof by substitution to show that a function $T(n)$ is big-O of some other function $f(n)$. Be sure you know where to find important formulas in your notes or in the book.

1. We roll a single six-sided die four times in a row. Each time, the top number is one of $\{1, 2, 3, 4, 5, 6\}$, each with probability $1/6$.
 - a) What is the probability that we roll the sequence exactly 2, 3, 2, 2?
 - b) How many possible sequences of four numbers can be produced by the four rolls? Position matters, that is, 2, 3, 2, 2 is different from 2, 2, 3, 2.

2. Give the best big-O upper bound you can for $\lg(n^{\lg n})$. Give explicit constants c, n_0 in the big-O bound.

3. Let us say we are given a long thin rectangle of size $k \times 2$. It contains $2k$ red points and $2k$ blue points, for a total of $n = 4k$. We are guaranteed that the distance between any two red points is at least 1, and also the distance between any two blue points is at least 1.

Consider the following recursive algorithm for finding the closest pair of points such that one point is red and the other is blue. We divide the points into two sets, P_L and P_R , such that the x -coordinate of each of the n points of P_L is \leq the x -coordinate of each of the n points of P_R . We recursively find the closest red-blue pair in P_L and P_R . Let δ be the distance between the closest red-blue pair found so far (which was the closest red-blue pair either in P_L or P_R). The only remaining possibility is that the closest red-blue pair lies in the small rectangle of height 2 and width 2δ around the vertical line separating P_L from P_R .

An argument similar to the “orange packing” argument we used in class shows that with so many points packed into the box, δ has to be less than one. So you can assume $\delta < 1$.

a) Give the smallest upper bound you can on the number of points in this small rectangle. No proof necessary.

b) Write down a recurrence relation describing the running time $T(n)$ to find the closest red-blue pair in the set P .

c) Give the best solution you can for this recurrence relation, and prove it by substitution. By “best” here, we mean the smallest upper bound possible. Hint: You might want to look at the subsection titled “Subtleties” in Section 4.1.

4. Consider inserting n items into a hash table of size m . Let $h(x)$ represent the hash table address into which item x is stored. Assume that the items are inserted into the table independently at random, so that $\Pr[h(x) = i] = 1/m$ for all $i \in \{0, \dots, p-1\}$, and, as we showed in class, $\Pr[h(x) = h(y)] = 1/m$, for all $x, y \in \{1, \dots, p-1\}$.

Consider any choice of $x, y, z \in \{1, \dots, p-1\}$. What is the probability that all three collide in the hash table, that is $\Pr[h(x) = h(y) = h(z)]$?

5. In Exercise 5.3-4, we showed that a function called PERMUTE-BY-CYCLIC(A) does not produce every permutation of input array A uniformly at random. One way to argue this is to show that there are many permutations of A which cannot be produced by PERMUTE-BY-CYCLIC. A nice way to do this involves counting.

1) Array A has n elements. How many permutations of the elements of A are there?

2) How many permutations of A are possible outputs of PERMUTE-BY-CYCLIC?

Consider a function very like the one we used as part of the universal hash function, $f(x) = (ax + b) \bmod n$. Here n is the size of array A , a is chosen randomly from $\{1, \dots, n-1\}$ and b is chosen randomly from $\{0, \dots, n-1\}$. When n is prime, we could use $f()$ to produce a permutation B of A as follows:

$$\begin{aligned} &\text{for } i = 0 \text{ to } n - 1 \\ &\quad B[f(i)] = A[i] \end{aligned}$$

3) Prove that when n is prime, B is a permutation of A . Hint: Prove that if $i \neq j$, then $f(i) \neq f(j)$. Why would this imply that B is a permutation of A ?

4) Assume that n is prime. How many permutations of A can be produced by this function?

5) Does this function produce every possible permutation of A ?