CS 31: Homework 1

Thomas Monfre

January 11, 2019

Problem 1-1

Prove the following. Let A be an event and $X_A = I\{A\}$ be the indicator random variable for A. Then for all real numbers $c > 0$, we have $E[X_A^c] = Pr\{A\}$.

See textbook page 118 and 119 for help on this. What does the superscript c mean?

Problem 1-2

Exercise 2.3-7: Describe a $\Theta(n \lg n)$-time algorithm that, given a set S of n integers and another integer x, determines whether or not there exists two elements in S whose sum if exactly x.

My answer here.

Problem 1-3

In class, we saw a lower-bound argument showing that the worst-case running time of insertion sort is $\Omega(n^2)$. The argument was based on dividing the array of n elements into three sections, each of size $\frac{n}{3}$.

Suppose that α is a fraction in the range $0 < \alpha < 1$. Show how to generalize the lower-bound argument for insertion sort to consider an input in which the αn largest values start in the first αn positions. What additional restriction do you need to put on α? What value of α maximizes the number of times that the αn largest values must pass through the middle $1 - 2\alpha$ array positions?

Going to have to review my notes on this one.

But for the generalization, I'm thinking that it will work so long as the middle section that we multiply by encompasses the whole thing.

Therefore, the middle section should be $1 - 2(\frac{n}{k})$

For the example of $\frac{n}{3}$, $\alpha = \frac{1}{3}$, so the middle section $= 1 - 2(\frac{n}{3}) = 1 - \frac{2n}{3} = \frac{n}{3}$.

Then we just have to multiply $\frac{n}{k} * (1 - 2(\frac{n}{k})) = \Theta(n^2)$

Work on cleaning this up, but that is the idea

ADDITIONAL RESTRICTION is that it must be less than 0.5

Problem 1-4

Exercise 4.2-4: What is the largest k such that if you can multiply 3×3 matrices using k multiplications (not assuming commutativity of multiplication), then you can multiply $n \times n$ matrics in time $o(n^{\lg 7})$? What would the running time of this algorithm be?

Read up on Strassen's Algorithm.

Problem 1-5

Problem 4-1. Use the master method when applicable. Don't worry about base cases.

(a) $T(n) = 2T(n/2) + n^4$

Using the master method we get $a = 2$, $b = 2$, and $f(n) = n^4$. Therefore, we must compare $n^{\log_2 2}$ with n^4.

Since $\log_2 2 = 1$, and $1 < 4$, $f(n) = n^4$ is polynomially larger than $n^{\log_2 2}$. This indicates case 3 of the master method applies. Checking the regularity condition we see $\frac{1}{8}(n^4) \leq cn^4$ for some constant $c < 1$ and all sufficiently large n.

Therefore, we have $\Theta(n^4)$.

(b) $T(n) = T(7n/10) + n$

Using the master method we get $a = 1$, $b = \frac{10}{7}$, and $f(n) = n$. Therefore, we must compare $n^{\log_{\frac{10}{7}} 1}$ with n.

Since $\log_{\frac{10}{7}} 1 = 0$, and $0 < 1$, $f(n) = n$ is polynomially larger than $n^{\log_{\frac{10}{7}} 1}$. This indicates case 3 of the master method applies. Checking the regularity condition we see $\frac{7}{10}(n) \leq cn$ for some constant $c < 1$ and all sufficiently large n.

Therefore, we have $\Theta(n)$.

(c) $T(n) = 16T(n/4) + n^2$

Using the master method we get $a = 16$, $b = 4$, and $f(n) = n^2$. Therefore, we must compare $n^{\log_4 16}$ with n^2.

Since $\log_4 16 = 2$, and $2 = 2$, $f(n) = n^2$ is within a polylog factor of $n^{\log_4 16}$, but is not smaller. This indicates case 2 of the master method applies.

Therefore, we have $\Theta(n^2)$.

(d) $T(n) = 7T(n/3) + n^2$

Using the master method we get $a = 7$, $b = 3$, and $f(n) = n^2$. Therefore, we must compare $n^{\log_3 7}$ with n^2.

Since $\log_3 7 = 1.7712437492$, and $1.7712437492 < 2$, $f(n) = n^2$ is polynomially larger than $n^{\log_3 7}$. This indicates case 3 of the master method applies. Checking the regularity condition we see $\frac{7}{9}(n^2) \leq cn^2$ for some constant $c < 1$ and all sufficiently large n.

Therefore, we have $\Theta(n^2)$.

(e) $T(n) = 7T(n/2) + n^2$

Using the master method we get $a = 7$, $b = 2$, and $f(n) = n^2$. Therefore, we must compare $n^{\log_2 7}$ with n^2.

Since $\log_2 7 = 2.8073549221$, and $2.8073549221 > 2$, $f(n) = n^2$ is polynomially smaller than $n^{\log_2 7}$. This indicates case 1 of the master method applies.

Therefore, we have $\Theta(n^{\log_2 7})$.

(f) $T(n) = 2T(n/4) + \sqrt{n}$

Using the master method we get $a = 2$, $b = 4$, and $f(n) = \sqrt{n} = n^{\frac{1}{2}}$. Therefore, we must compare $n^{\log_4 2}$ with $n^{\frac{1}{2}}$.

Since $\log_4 2 = \frac{1}{2}$, and $\frac{1}{2} = \frac{1}{2}$, $f(n) = n^{\frac{1}{2}}$ is within a polylog factor of $n^{\log_4 2}$, but is not smaller. This indicates case 2 of the master method applies.

Therefore, we have $\Theta(n^{\frac{1}{2}})$.

(g) $T(n) = T(n - 2) + n^2$

asdfasdfasdf

Problem 1-6

Problem 4-3, parts g and j. Don't worry about base cases.

(g) $T(n) = T(n-1) + \lg n$

asdfasdfasdf

(j) $T(n) = \sqrt{n}\, T(\sqrt{n}) + n$

asdfasdfasdf

Problem 1-7

Exercise 5.2-5: Let $A[1...n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an **inversion** of A. Suppose that the elements of A form a uniform random permutation of $\langle 1, 2, ..., n \rangle$. Use indicator random variables to compute the expected number of inversions.

---

asdfasdfasdfasdfafd