CS 6150: HW0 – LaTeX Intro and Background

Submission date: Tuesday, Aug 30 2016

This assignment has 5 questions, for a total of 50 points. Unless otherwise specified, complete and reasoned arguments will be expected for all answers.

Question	Points	Score
Basic Background: Big Oh	10	
Removing Duplicates	10	
Proof Practice: Square vs. Multiply	10	
Background: Probability	10	
Array Sums	10	
Total:	50	

Question 1: Basic Background: Big Oh ..... **[10]**

Write each of the functions below (n is the positive integer variable) in the Big-Oh notation.

(a) **[2]** $f(n) = n^2 + 5n + 20$.

(b) **[2]** $f(n) = 2\log n + 4$.

(c) **[2]** $f(n) = \frac{1}{n^2} + \frac{2}{n}$.

(d) **[2]** $f(n) = \frac{1}{n^2} + 1$.

(e) **[2]** $f(n) = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$.

Question 2: Removing Duplicates ..... **[10]**

Let $A[1 \ldots n]$ be an array of integers. Describe an algorithm that runs in time $O(n \log n)$ and returns an array B whose entries are all the *distinct* elements of A (i.e., with no duplicates).

Question 3: Proof Practice: Square vs. Multiply ..... **[10]**

Suppose I tell you that there is an algorithm that can square any n digit number in time $O(n \log n)$, for all $n \geq 1$. Then, prove that there is an algorithm that can find the product of *any two* n digit numbers in time $O(n \log n)$. [*Hint:* think of using the squaring algorithm as a subroutine to find the product.]

Question 4: Background: Probability ..... **[10]**

Write the answers to the following as functions of k:

(a) **[4]** Suppose we toss a fair coin k times. What is the probability that we see heads precisely once?

(b) **[6]** Suppose we have k different boxes, and suppose that every box is colored uniformly at random with one of k colors (independently of the other boxes). What is the probability that all the boxes get distinct colors?

Question 5: Array Sums ..... **[10]**

Given an array $A[1 \ldots n]$ of integers (not necessarily distinct), find if there exist indices i, j, k such that $A[i] = A[j] + A[k]$. Can you find an algorithm with running time $o(n^3)$? [NOTE: this is the little-oh notation, i.e., the algorithm should run in time $< cn^3$, for any constant c, as $n \to \infty$.] [*Hint:* aim for an algorithm with running time $O(n^2 \log n)$.]