COMS 3110: Homework 2
Due: July 8^{th}, 11:59pm
Total Points: 65

Submission format. Homework solutions will have to be typed. You can use word, LaTeX, or any other type-setting tool to type your solution. Your submission file should be in pdf format. Do **NOT** submit a photocopy of handwritten homework except for diagrams that can be hand-drawn and scanned. We reserve the right **NOT** to grade homework that does not follow the formatting requirements. Name your submission file: `<Your-net-id>-3110-hw2.pdf`. For instance, if your netid is `asterix`, then your submission file will be named `asterix-3110-hw2.pdf`. Each student must hand in their own assignment. If you discussed the homework or solutions with others, a list of collaborators must be included with each submission. Each of the collaborators has to write the solutions in their own words (copies are not allowed).

General Requirements

- When asked to design an algorithm, part of the grade will depend on the efficiency of your algorithm.
- When proofs are required, do your best to make them both clear and rigorous. Even when proofs are not required, you should justify your answers and explain your work.
- When asked to present a construction, you should show the correctness of the construction.

Some Useful (in)equalities

- $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$
- $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$
- $2^{\log_2 n} = n$, $a^{\log_b n} = n^{\log_b a}$, $n^{n/2} \leq n! \leq n^n$, $\log x^a = a \log x$
- $\log(a \times b) = \log a + \log b$, $\log(a/b) = \log a - \log b$
- $a + ar + ar^2 + ... + ar^{n-1} = \frac{a(r^n-1)}{r-1}$
- $1 + \frac{1}{2} + \frac{1}{2^2} + ... + \frac{1}{2^n} = 2(1 - \frac{1}{2^{n+1}})$
- $1 + 2 + 4 + ... + 2^n = 2^{n+1} - 1$

---

1. (15 points) Consider an array A of 0's and 1's such that the 0's appear in the array before all the 1's. The objective is to find the largest index i such that $A[i] = 0$. Write an algorithm that takes as input A and outputs i. Derive the runtime of your algorithm.

2. (20 points) Consider an array A of integers. Write an algorithm that outputs the length of the *longest* subarray where the sum of the elements in the subarray is equal to 0.

   For example, for the input array $A = \{13, 11, -2, 1, 7, -15, -2, 3, -3, 10\}$, the output of your algorithm should be 8, for the array $A = \{-13, 1, 0, 2\}$ the output should be 1, and for the array $A = \{-13, 1, -2\}$ the output should be 0. Derive the runtime of your algorithm. Use of hash tables and expected runtime is acceptable.

3. (10 points) Draw the binary min-heap tree and the corresponding array where the elements are inserted into the heap in the following order:

   $$47,\ 63,\ 88,\ 91,\ 23,\ 14,\ 3,\ 55,\ 30,\ 99$$

4. (20 points) Consider an array of integers that is *k-almost sorted*. By k-almost sorted, we refer to the situation where the index of any element can be at most k indices away from its correct index as per ascending order. For instance, the array

0	1	2	3	4	5	6
2	1	0	6	5	3	4

   is 3-almost sorted because every element is at most 3 positions away from its correct index. Write an algorithm that takes A and k as input and returns a sorted array. Derive the runtime of your algorithm.