CIS 1210 — Data Structures and Algorithms
Homework Assignment 5

Assigned: October 8, 2024 **Due:** October 21, 2024

Note: The homework is due **electronically on Gradescope** on October 21, 2024 by 11:59 pm ET. For late submissions, please refer to the Late Submission Policy on the course webpage. You may submit this assignment up to 2 days late.

A. Gradescope: You must select the appropriate pages on Gradescope. Gradescope makes this easy for you: before you submit, it asks you to associate pages with the homework questions. Forgetting to do so will incur a 5% penalty, which cannot be argued against after the fact.

B. LaTeX: You must use the LaTeX template provided on the course website, or a 5% penalty will be incurred. Handwritten solutions or solutions not typeset in LaTeX will not be accepted.

C. Solutions: Please write concise and clear solutions; you will get only a partial credit for correct solutions that are either unnecessarily long or not clear. Please refer to the Written Homework Guidelines for all the requirements. Piazza will also contain a complete sample solution in a pinned post.

D. Algorithms: Whenever you present an algorithm, your answer must include 3 separate sections. Please see Piazza for an example complete solution.

1. A precise description of your algorithm in English. No pseudocode, no code.
2. Proof of correctness of your algorithm
3. Analysis of the running time complexity of your algorithm

E. Collaboration: You are allowed to discuss **ideas** for solving homework problems in groups of up to 3 people but *you must write your solutions independently*. Also, you must write on your homework the names of the people with whom you discussed. For more on the collaboration policy, please see the course webpage.

F. Outside Resources: Finally, you are not allowed to use *any* material outside of the class notes and the textbook. Any violation of this policy may seriously affect your grade in the class. If you're unsure if something violates our policy, please ask.

1. [10 pts] Sally loves Sally Brown!

Sally is a fan of the Peanuts franchise, especially because she shares a name with Sally Brown, Charlie Brown's younger sister. She looks at all the n Peanuts comics in her collection and wants to label them based on how much she likes each and chooses to label by using a prefix-free string consisting only of the numbers 0 and 1.

To determine this systematically, she decides to consider how many times the character Sally Brown appears in each comic. For each comic c, Sally determines f_c, the number of Sally Brown appearances in c divided by the total Sally Brown appearances over all n comics. She chooses to give shorter labels to the comics where Sally appears more, since she's more likely reach for them (and she hates reading long binary strings). As an avid computer scientist, Sally uses the Huffman encoding algorithm along with $f_1, f_2, ..., f_n$ to create n optimal labels for her comics.

(a) What is the length of the longest label Sally could have created in terms of n? Give an example set of Sally Brown frequencies per comic that would produce this case. Note that your set of frequencies must be defined in such a way that is generalizable for any value of n. Your set of frequencies must also be valid (i.e. all the frequencies sum to 1), but you do not need to prove this. You also do not need to prove that your proposed set of frequencies will produce the desired result.

As an example, you could say that in the first comic Sally appears $\frac{1}{2}$ times, in the second comic Sally appears $\frac{1}{2}$ times, and 0 times in all other $n-2$ comics since it sums to 1 and is generalizable in terms of n (this is probably not the right answer though).

(b) Independent of your solution to part (a), prove that if some comic had a Sally Brown frequency greater than $\frac{3}{7}$, then Sally must have had a comic with a label of length 1.

2. [15 pts] Snoopy Sort

Kevin H. has a large collection of stuffed Peanuts characters, and has carefully organized all the plushies in alphabetical order. To celebrate Snoopy's 70th anniversary, he invited over all the CIS 1210 TAs to see his collection. Unfortunately, they got too excited and ruined his ordering! Luckily, due to the way the TAs placed the plushies, each plush is at most k positions away from their original ordering.

Kevin, irritated from the carelessness of his fellow TAs, only wants to spend $O(n \log k)$ time sorting his plushies. Given that you know names of every plush, help Kevin come up with an $O(n \log k)$ algorithm to sort his plushies. You are provided with k as an input to the algorithm and the name of each plush (as an array $A[1..n]$ where $A[1]$ represents the plush in the first position of the unorganized collection). You are only required to sort A, the array provided to you, and you can assume $k < n$.

The following list would be an example of $n = 5$ and $k = 2$:

[Sally, Charlie, Woodstock, Lucy, Snoopy]

Your algorithm should return:

[Charlie, Lucy, Sally, Snoopy, Woodstock]

Note: You only need to provide an algorithm and running time justification, but we highly recommend thinking about how to go about justifying proof of correctness.