Homework 1

CS 4364/5364
Spring 2022

Due: 31 January 2021

Because of the reliance of the particular assignments in this class on mathematical notation, and the fact that all assignments will be submitted electronically, students are encouraged to use LaTeX to formalize their responses. **For those enrolled in the graduate section the use of latex is *required*.** This assignment (like all others) will be posted on the course `github`[1] as source code as well as in PDF form on the course website. Please submit your assignment to the professor via email, either as a link to your assignment online (i.e. overleaf or github) or as an attachment. Graduate students will need to include the `.tex` files as well as a PDF, this is optional but encouraged for undergraduates.

1. **(10 points) Fill out the welcome questionnaire for the class. The link can be found on the course website. Respond to this question by providing the URL of the questionnaire.**

   `https://forms.gle/NsfFmRSHaEQaBcTV6`

2. **(20 points) Provide an algorithm that given a string $S = s_1 s_2 s_3 ... s_n \in \Sigma^*$ finds the *set of* most abundant characters $C \subseteq \Sigma$. Describe the algorithm, give a justification of its correctness, and determine it's running time.**

   Note that the algorithm description should be a step-by-step description in plain English. (i.e. no psuedo-code). You can use mathematical notation as needed.

   Note also that when the string is empty, all characters are equally present, and thus $C = \Sigma$.

   Algorithm:

   (a) define for each character $c \in \Sigma$ a value $ct_c = 0$ which will hold the count of the character c in S

[1] `github.com/deblasiolab/CS4364-documents`

(b) for each $s_i \in S$ increment ct_{s_i}

(c) let $maxVal = \max_{c \in \Sigma} \{ct_c\}$

(d) define a set $C = \emptyset$

(e) for all $c \in \Sigma$, if $ct_c = maxVal$: $C = C \cup \{c\}$

(f) return C

Correctness: Each subsequent s_i in (b) will increase one of the ct_c's so $maxVal \neq 0$. Steps (c) and (e) combined first find the *count* of the most abundant character (max over the count values) and then collect all of the characters with that count to be returned, which is the value asked for in the problem. Because all of the values ct_c are initially 0 in (a), if $S = \varepsilon$ all $c \in \Sigma$ will be equal (0) since there will be no s_i to execute on in (b), and thus C will contain all of Σ.

Runtime: Steps (a), (c) and (e) will all take time $O(\sigma)$ (they must make a linear scan over a data structure the size of the alphabet). Steps (d) and (f) will take $O(1)$ time. Step (b), depending on the data structure used to store the ct_c's, will take a best-worst case time of $O(\sigma)$-time per character (since we never defined an order over $c \in \Sigma$, and we cannot guarantee for a hash based storage medium we can an even distribution over hash values over the characters). Therefore the running time of (b) is $O(n\sigma)$, and this is the dominating factor. Thus the running time overall is $O(n\sigma)$.