Homework Assignment #9
Due: November 29, 2024 at 5:00 p.m.

1. Georgy would like to build a hash table storing strings of various lengths. He doesn't know in advance how long the strings might be. He wants to come up with a general method for choosing a hash function so that the probability of collision between any two strings is small.

A string x of length ℓ is represented as a sequence of characters $\langle x_0, x_1, \ldots, x_\ell \rangle$ where x_ℓ is a special ETX (end of text) character that cannot appear earlier in the string. Each character is encoded in ASCII as a number between 0 and 127 (ETX is 3).

To make things simple, the size of the hash table (i.e., the number of buckets) is a prime number p, and p is bigger than 128.

To obtain a hash function h, Georgy picks random numbers $a_0, a_1, a_2, \ldots$ (uniformly and independently), where each $a_i \in \{0, 1, 2, \ldots, p-1\}$ and uses $h(\langle x_0, x_1, \ldots, x_\ell \rangle) = (\sum_{i=0}^{\ell} a_i x_i) \bmod p$

[4] **(a)** Let $x = \langle x_0, x_1, \ldots, x_\ell \rangle$ and $y = \langle y_0, y_1, \ldots, y_m \rangle$ be two different strings (possibly with different lengths).

(i) Explain why there must be some k such that $0 \le k \le \min(\ell, m)$ and $x_k \ne y_k$.

(ii) Show that $h(x) = h(y)$ if and only if $a_k(x_k - y_k) \bmod p = (\sum_{\substack{i=0 \\ i \ne k}}^{\ell} a_i y_i - \sum_{\substack{i=0 \\ i \ne k}}^{m} a_i x_i) \bmod p$.

(iii) Show that the probability that $h(x) = h(y)$ is $\frac{1}{p}$.
Hint: focus on the choice of a_k.

[2] **(b)** Where, exactly, in part (a) did you use the assumption that $p > 128$?
Where, exactly, did you use the assumption that the a_i's are independent?

[4] **(c)** Suppose Georgy wants to represent a set S of n strings in a hash table of size $p > n$. He will pick a random hash function as described above and use chaining to resolve collisions.

(i) How should he store $a_0, a_1, \ldots$? He cannot choose infinitely many a_i's before starting to build the hash table, so when should a_i be chosen?

(ii) Once the hash table for S is built, what is the expected time to perform a search for a string of length ℓ? Justify your answer. Include the time to compute the hash function and the time to look for the string inside a bucket.
Assume that you can do arithmetic operations and comparisons on integers in $\{0, \ldots, p\}$ in constant time.