Computational Problem Solving CSCI-603
Ciphers Lab 2

5/26/2023

1 Introduction[1]

You would like to send messages back and forth with a friend (or co-conspirator!) but want to make sure that other people cannot easily read those messages. However, rather than use a fixed encryption scheme, you decide to take your message string and apply a series of transformations to it to generate the encrypted message.

The transformations you have agreed to use are the following:

- S_i shifts the letter at index i forward one letter in the alphabet. So, BALL→ S_0 → CALL.
  This can be applied multiple times to shift multiple letters forward, and if so would be designated S_i^k to shift letter i by k forward. If the shift takes the letter past the end of the alphabet, it will wrap around. Negative exponents shift the letter backward in the alphabet.
- R rotates the string one position to the right. So, TOPS→ R → STOP.
  This function can also be used with an exponent (positive or negative). For example, TRAIN→ R^2 → INTRA.
- D_i duplicates (in place) the letter at index i. So, HOPED→ D_2 → HOPPED.
  This can also be used with a positive exponent to produce multiple duplicates, but not with negative exponents.
- $T_{i,j}$ swaps the letters at index i and index j. So, SAUCE→ $T_{0,3}$ → CAUSE. You can always assume that $i < j$.

To more effectively obfuscate your message, you can apply several transformations. For example, CANAL → $R^2\, D_2\, S_2^9$ → ALLCAN. These transformations will be applied in order from the left to right, the ouput generated by an operation will be the input to the next operation in the sequence.

[1] This problem is inspired by a puzzle by Dan Katz
(`http://web.mit.edu/puzzle/www/2007/puzzles/transmogrifiers/`)

2 Problem Solving (20%)

Students will be organized into groups. Each group will work together to deliver answers to the following questions:

1. What are the results of the following transformations?
   (a) ZOO $\rightarrow S_0^2 \rightarrow$
   (b) SUCES $\rightarrow D_2\,D_5 \rightarrow$
   (c) HORSE $\rightarrow T_{2,4}\,S_4\,R \rightarrow$

2. Consider the simple rotation (R) operation that rotates the string one position to the right. For a given string `msg`, how would you implement this operation in Python? Write this as a Python function called `rotate(msg: str) -> str`.

3. Modify your previous answer so that your function takes also a positive exponent as an additional argument and performs the appropriate operation.

4. Python provides the `ord` function to convert a single character to an integer UNICODE value, and the `chr` function to convert from a UNICODE integer to a character. With that in mind, write a code snippet that shifts a single uppercase character `c` by a positive amount k, keeping in mind that if this takes you past the end of the alphabet, it should wrap around.

5. The operation S_i^k will be represented in the input to your program with the text string `Si,k` — so, for example, the operation of question 1(a) would be given to your program as `S0,2`. Assume you have already implemented a function with the following signature `shift(msg: str, index: int, exponent: int) -> str` **(you don't need to write this function)**. If you are given that operation string (in a variable `s`) and a message string `m`, show how to generate the correct call to the `shift` function. Keep in mind that `i` and `k` can be arbitrarily large integers!

6. You will want to be able to decrypt as well as encrypt! Luckily, many of the functions can be inverted in a simple fashion. For example, if you get the information ? $\rightarrow S_0 \rightarrow$ CALL from your colleague, you can retrieve the original message by applying S_0^{-1} to CALL. For the following encryption operations, show (if possible!) the operation(s) to perform the decryption using the same operation itself.
   (a) S_2^3
   (b) R^2
   (c) D_1
   (d) $T_{2,4}$
   (e) $R\,S_1$