MATH 75: Cryptography **Spring 2022**

PSET 2 — April 15, 2022

Prof. Asher Auel ***Student: Amittai Siavava***

Credit Statement

I worked on these problems alone, with reference to class notes and the following books:

(a) ***The Code Book*** by **Simon Singh**.
(b) ***Cryptography*** by **Simon Rubinsen-Salzedo**

I also wrote some code to help in automating some of the problems.
Rather than upload all the source code, I tried to demonstrate the logic in my responses.
Please let me know if you would like to see the source-code.

Problems

1. A disadvantage of the general substitution cipher is that both sender and receiver must commit the permuted cipher sequence to memory. A common technique for avoiding this is to use a keyword from which the cipher sequence can be generated. For example, using the keyword `CIPHER`, write out the keyword followed by unused letters in normal order and match this against the plaintext letters:

```
plain:   a b c d e f g h i j k l m n o p q r s t u v w x y z
cipher:  C I P H E R A B D F G J K L M N O Q S T U V W X Y Z
```

If it is felt that this process does not produce sufficient mixing, write the remaining letters on successive lines and then generate the sequence by reading down the columns:

```
C I P H E R
A B D F G J
K L M N O Q
S T U B W X
Y Z
```

This yields the sequence: `C A K S Y I B L T Z P D M U H F N V E G O W R J Q X`.
Such a system is used in the following decoded ciphertext:

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
itwasdisclosedyesterdaythatseveralinformalbut

VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
directcontactshavebeenmadewithpolitical

EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
representativesofthevietconginmoscow
```

Determine the keyword.

First, I rearranged the plaintext letters, each with the corresponding ciphertext letter, to see if any keyword stands out.

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
S A H V P B J W U - - X T D M Y - E O Z I F Q - G -
```

No pattern is apparent in the sequence of letters.

I then wrote a program to simulate the splitting of words, in similar fashion as when an n-sized key is used but the letters are written on successive rows then read column-wise.

```
n: 2  SD                 -- Doesn't make sense.
n: 3  S-O                -- Doesn't make much sense.
n: 4  SWMI               -- Doesn't make sense.
n: 5  SJX-F              -- Doesn't make sense.
n: 6  SB-MOQ             -- Doesn't make sense.
n: 7  SPUT-I-            -- Doesn't make sense at first, but...
n: 8  SPUXMEI-           -- Doesn't make sense.
n: 9  SVJ-TYOFG          -- Doesn't make sense.
n: 10 SVJ-TYOIQG         -- Doesn't make sense.
n: 11 SVJ-TM-OIQG        -- Doesn't make sense.
n: 12 SVJU-TM-OIQG       -- Doesn't make sense.
n: 13 SHPJU-TM-OIQG      -- Doesn't make sense.
```

None of the possible keywords immediately form an English word. However, filling in two of the missing letters into the keyword generated with $n = 7$ gives `SPUTNIK` a Russian satelite launched in 1957. My bet is on that being the keyword.

The full keyword - alphabet generation:

```
S P U T N I K
A B C D E F G
H J L M N O Q
V W X Y Z
```

2. Let $n \geq 3$ and S_n be the symmetric group on $\{1, \ldots, n\}$. We say that $\sigma \in S_n$ *has a fixed point* if there exists $k \in \{1, \ldots, n\}$ such that $\sigma(k) = k$. Prove that the probability that a random $\sigma \in S_n$ has a fixed point is $\geq 5/8$ and $\leq 2/3$. Here, "probability" means that the number of those permutations with a fixed point divided by the number of all permutations. (Conclude that a random substitution cipher, realized as a random permutation in S_n, is likely to fix at least one symbol.)

Let $p(n)$ define the probability that a random $\sigma \in S_n$ has some fixed point. Then:

$$p(n) = p(\text{single fixed point}) + p(2 \text{ fixed points}) + \cdots + p(n \text{ fixed points}) \tag{1}$$

We also have to account for permutations that have multiple fixed points.
While isolating repeated points, some are deducted twice and have to be re-added into the sum.

$$p(n) = \binom{n}{1} \cdot (n-1)! - \binom{n}{2} \cdot (n-2)! + \binom{n}{3}(n-3)! - \cdots \pm \binom{n}{n} \cdot (n-n)! \tag{2}$$

We can expand this sequence into an alternating series.
We also have to remember to divide by the numner of all possible permutations because we want the probability and not the counts.

$$p(n) = \frac{\sum_{k=1}^{n} (-1)^{k+1} \frac{n!}{k!(n-k)!} \cdot (n-k)!}{n!} = \sum_{k=1}^{n} \frac{(-1)^{k+1}}{k!} \tag{3}$$

$$p(3) = 1 - \frac{1}{2!} + \frac{1}{3!} = \frac{2}{3} \approx 0.6666 \tag{4}$$

$$p(4) = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4} = \frac{5}{8} = 0.625 \tag{5}$$

$$p(5) = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4} + \frac{1}{5!} = \frac{19}{30} \approx 0.6333 \tag{6}$$

$$p(6) = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4} + \frac{1}{5!} - \frac{1}{6!} = \frac{91}{144} \approx 0.6319 \tag{7}$$

$$p(7) = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4} + \frac{1}{5!} - \frac{1}{6!} + \frac{1}{7!} = \frac{177}{280} \approx 0.6321 \tag{8}$$

$$p(8) = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4} + \frac{1}{5!} - \frac{1}{6!} + \frac{1}{7!} - \frac{1}{8!} = \frac{3641}{5760} \approx 0.6321 \tag{9}$$

$$p(9) = 1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4} + \frac{1}{5!} - \frac{1}{6!} + \frac{1}{7!} - \frac{1}{8!} + \frac{1}{9!} \approx 0.6321 \tag{10}$$

$$\vdots$$

After expanding a few terms of the sequence, we see that the series queickly converges to $p(n) \approx 0.6321$.

3. Let $a, b \in \mathbb{Z}$.

(a) Let $\gcd(a,b) = g \neq 0$. Prove that $\gcd(a/g, b/g) = 1$.

By factorization let:

$$a = \prod i^{a_i}, \; b = \prod i^{b_i} \; \forall i \in \mathbb{Z}^+ \tag{11}$$

Then, since the gcd of two numbers must divide both numbers:

$$g = \gcd(a,b) = \prod i^{\min\{a_i, b_i\}} \; \forall i \in \mathbb{Z}^+ \tag{12}$$

And;

$$\frac{a}{g} = \prod i^{a_i - \min\{a_i,b_i\}}, \; \frac{b}{g} = \prod i^{b_i - \min\{a_i,b_i\}} \; \forall i \in \mathbb{Z}^+ \tag{13}$$

However, we know that:

$$\min\{a_i, b_i\} = a_i, \textbf{ OR } \min\{a_i, b_i\} = b_i \; \forall i \in \mathbb{Z}^+ \tag{14}$$

The equation (4) implies that *at least* one of the following is true:

$$a_i - \min\{a_i, b_i\} = 0, \textbf{OR } b_i - \min\{a_i, b_i\} = 0 \; \forall i \in \mathbb{Z}^+ \tag{15}$$

Thus:

$$\gcd(\frac{a}{g}, \frac{b}{g}) = \prod i^{\min\{0, a_i, b_i\}} \; \forall i \in \mathbb{Z}^+ \tag{16}$$

$$\gcd(\frac{a}{g}, \frac{b}{g}) = \prod 1 \; \forall i \in \mathbb{Z}^+ \tag{17}$$

$$\gcd(\frac{a}{g}, \frac{b}{g}) = 1 \tag{18}$$

(b) Prove that $\gcd(a+kb, b) = \gcd(a, b)$ for all $k \in \mathbb{Z}$. We stated this in lecture, but now you can check it carefully yourself!

Let $g_1 = \gcd(a, b)$ and $g_2 = \gcd(a+kb, b)$

By definition of g_1 as the gcd of a and b:

$$g_1 \mid a \text{ and } g_1 \mid b \Rightarrow \exists x, y \in \mathbb{Z}^+ \ni a = g_1 x \text{ and } b = g_1 y \tag{1}$$

Similarly, by definition of g_2 as the gcd of $a+kb$ and b:

$$g_2 \mid (a+kb) \text{ and } g_2 \mid b \Rightarrow \exists m, n \in \mathbb{Z}^+ \ni (a+kb) = g_2 m \text{ and } b = g_2 n \tag{2}$$

We *know* that g_2 divides b (by definition of it being the gcd of $a+kb$ and b).

Therefore, subtracting a multiple of g_2 (in this case, b), from $a+kb$ does not change the divisibility status of result.

In particular,

$$g_2 \mid b \land g_2 \mid (a+kb) \Rightarrow g_2 \mid (a+kb-b) \Rightarrow g_2 \mid (a+kb-2b) \Rightarrow \cdots \Rightarrow g_2 \mid (a+kb-kb) \Rightarrow g_2 \mid a \tag{3}$$

Similarly, we know that g_1 divides both a and b (by definition of it being the gcd of the two numbers). Therefore, adding a multiple of g_1 (in this case, b) to a does not change the divisibility status.

In particular,

$$g_1 \mid a \land g_1 \mid b \Rightarrow g_1 \mid (a+b) \Rightarrow g_1 \mid (a+2b) \Rightarrow \cdots \Rightarrow g_1 \mid (a+kb) \tag{4}$$

Thus, we have shown that $g_1 \mid (a+kb)$ and $g_2 \mid a$.

However, we earlier defined $g_1 = \gcd(a, b)$ and $g_2 = \gcd(a+kb, b)$.

This implies:

$$g_1 \leq g_2 \land g_2 \leq g_1 \Rightarrow g_1 = g_2 \tag{5}$$

4. Compute some inverses!

(a) Use the extended Euclidean algorithm to compute 367^{-1} in $(\mathbb{Z}/1001\mathbb{Z})^{\times}$ and 1001^{-1} in $(\mathbb{Z}/367\mathbb{Z})^{\times}$. [Do this by hand.]

367^{-1} in $(\mathbb{Z}/1001\mathbb{Z})^{\times}$

First, we can use the Euclidean algorithm to compute the gcd (and decompose the coefficients).

$$x \equiv 367^{-1} \pmod{1001} \tag{1}$$

$$1001 = 2 \cdot 367 + 267 \tag{2}$$

$$367 = 1 \cdot 267 + 100 \tag{3}$$

$$267 = 2 \cdot 100 + 67 \tag{4}$$

$$100 = 1 \cdot 67 + 33 \tag{5}$$

$$67 = 2 \cdot 33 + 1 \tag{6}$$

We can now use the extended Euclidean algorithm to back-substitute and compute the inverse.

$$1 = 67 - 2 \cdot 33 \tag{7}$$

$$1 = 100 - 3 \cdot 33 \tag{8}$$

$$1 = -2 \cdot 100 + 3 \cdot 67 \tag{9}$$

$$1 = -8 \cdot 100 + 3 \cdot 267 \tag{10}$$

$$1 = -8 \cdot 367 + 11 \cdot 267 \tag{11}$$

$$1 = -30 \cdot 367 + 11 \cdot 1001 \tag{12}$$

$$1 \equiv -30 \cdot 367 + 11 \cdot 1001 \pmod{1001} \tag{13}$$

$$1 \equiv -30 \cdot 367 \pmod{1001} \tag{14}$$

$$1 \equiv 971 \cdot 367 + 11 \cdot 1001 \pmod{1001} \tag{15}$$

Thus: 367^{-1} in $(\mathbb{Z}/1001\mathbb{Z})^{\times} = 971$

1001^{-1} in $(\mathbb{Z}/367\mathbb{Z})^{\times}$

First, we can use the Euclidean algorithm to compute the gcd (and decompose the coefficients).

$$x \equiv 1001^{-1} \pmod{367} \equiv 267 \pmod{367} \tag{1}$$

$$367 = 1 \cdot 267 + 100 \tag{2}$$

$$267 = 2 \cdot 100 + 67 \tag{3}$$

$$100 = 1 \cdot 67 + 33 \tag{4}$$

$$67 = 2 \cdot 33 + 1 \tag{5}$$

We can now use the extended Euclidean algorithm to back-substitute and compute the inverse.

$$1 = 67 - 2 \cdot 33 \tag{6}$$

$$1 = 100 - 3 \cdot 33 \tag{7}$$

$$1 = -2 \cdot 100 + 3 \cdot 67 \tag{8}$$

$$1 = -8 \cdot 100 + 3 \cdot 267 \tag{9}$$

$$1 = -8 \cdot 367 + 11 \cdot 267 \tag{10}$$

$$1 \equiv -8 \cdot 367 + 11 \cdot 267 \pmod{367} \equiv 11 \pmod{367} \tag{11}$$

Thus: 1001^{-1} in $(\mathbb{Z}/367\mathbb{Z})^{\times} = 11$

(b) Compute 314159265^{-1} in $(\mathbb{Z}/2718281828\mathbb{Z})^{\times}$. [You may use a computer!]

My first approach, which was really slow, was to count up from 1 to the base, checking if any number multiplied by the target to give a product congruent to 1.

```
-- | Compute the inverse of a number modulo N
invN :: Integral p => p -> p -> p  -- Type must be integral (i.e. support div, mod)
invN _ 0 = 0
invN _ 1 = 1
invN base a = iter base a 1
  where

    -- | Iterate from 1 to n, checking every number.
    iter n a b
    | a > n || b > n = 0
    | a ‘mul‘ b == 1 = b
    | otherwise = iter n a (b + 1)

    mul = mulN n
    -- | Compute multiplication modulo N.
    --  The result is always in the range [0,N).
    mulN :: Integral a => a -> a -> a -> a
    mulN base a b = (a * b) ‘mod‘ base
```

This program was not very promising (it was doing too much unnecessary computation).

My next approach was to write a version of the euclidean algorithm that tracks Bezout’s coefficients as it runs the Euclidean algorithm.

```
-- | Run the Euclidean algorithm while tracking Bezout’s coefficients
euclid :: Integral b => b -> b -> (b, b, b)
euclid 0 y = (y, 0, 1)
euclid x y =
  let (gcd, x’, y’) = euclid (y ‘mod‘ x) x
      x = y’ - (y ‘div‘ x) * x’
      y = x’
  in (gcd, x, y)

--- Results
number: 2718281828
number: 314159265
(1,-143182048,1238891233) --> (gcd, coefficient 1, coefficient 2)
```

Therefore, the inverse of 314159265 in $(\mathbb{Z}/2718281828\mathbb{Z})^{\times}$ is 1238891233.

5. Let $f_0 = f_1 = 1$ and $f_{i+1} = f_i + f_{i-1}$ for $i \geq 1$ denote the Fibonacci numbers.

(a) Use the Euclidean algorithm to show that $\gcd(f_i, f_{i-1}) = 1$ for all $i \geq 1$. (Again, we did this quickly in lecture, but now do it carefully!)

For arbitrary $i \in \mathbb{Z}^+$,

$$f_{i+1} = f_i + f_{i-1} \text{ (by definition of the Fibonacci numbers)} \tag{1}$$

Let g be the greatest common divisor of $f_i + 1$ and f_i.,

$$g \mid f_i \wedge g \mid f_{i+1} \Rightarrow g \mid (f_{i+1} - f_i) = f_{i-1} \tag{2}$$

$$g \mid f_{i-1} \wedge g \mid f_i \Rightarrow g \mid (f_i - f_{i-1}) = f_{i-2} \tag{3}$$

$$g \mid f_{i-2} \wedge g \mid f_{i-1} \Rightarrow g \mid (f_{i-1} - f_{i-2}) = f_{i-3} \tag{4}$$

$$\vdots$$

$$g \mid f_3 \wedge g \mid f_2 \Rightarrow g \mid (f_3 - f_2) = f_1 \tag{5}$$

$$g \mid f_2 \wedge g \mid f_1 \Rightarrow g \mid (f_2 - f_1) = f_0 = 1 \tag{6}$$

$$g \mid 1 \Rightarrow g = 1 \tag{7}$$

(b) Find $\gcd(11111111, 11111)$.

Using the Euclidean algorithm:

$$\gcd(11111111, 11111) = \gcd(11111, 111) \tag{1}$$

$$\gcd(11111, 111) = \gcd(111, 11) \tag{2}$$

$$\gcd(111, 11) = \gcd(11, 1) \tag{3}$$

$$\gcd(11, 1) = \gcd(1, 0) = 1 \tag{4}$$

(c) Let $a = 111\cdots 11$ be formed with f_i repeated 1s and let $b = 111\cdots 11$ be formed with f_{i-1} repeated 1s. Find $\gcd(a,b)$.
[Hint: Compare your computations in parts (a) and (b).]

Let $S = \{s_1, s_2, s_3, \dots\} \subset \mathbb{Z}$ be the set of all integers formed by some i repeated 1s.
For arbitrary i and $j \in \mathbb{Z}^+$ such that $i \geq j$, $s_i \mod s_j = s_{i \mod j}$

Proof. In S, multiplying elements by positive integer powers of 10 is equivalent to a shift of the 1s in the number. For instance $11 \cdot 10 = 110$. This does not map elements in S into S, but it allows us to deduct deduct these new shifted values from elements in S and get rid of leading 1s. For two powers i and j such that $i \geq j$, we can repeatedly erase the first j 1s in s_i by deducting a shifter version of s_j, until we are no longer able to (at which point the remaining value has less 1s than s_j.) This is analogous to taking the $\mod$ of i by j and defining the remainder, $i \mod j$, to be the number of 1s that we get after fully erasing occurrences of s_j in s_i.
A good example can be seen in (b) above. Here is another example:

$$11111111 \equiv 11 \pmod{111} \tag{1}$$

Or, generally:

$$s_8 \mod s_3 \equiv s_{(8 \pmod 3)} = s_2 \pmod{s_3} \tag{2}$$

□

Thus, following from part (a) above:

$$\gcd(s_{f_{i+1}}, s_{f_i}) = \gcd(s_{f_i}, s_{f_{i-1}}) = \cdots = \gcd(s_{f_1}, s_{f_0}) = \gcd(1,1) = 1 \tag{3}$$