Ciphertext-only attack for affine ciphers.

Algorithm:
- Enumerate all possible keys \((a, b)\)
- Let \(B\) denote letter frequencies in ciphertext.

\[
A_0 = \begin{pmatrix} \vdots & \vdots & \vdots \\ a & b & \vdots \\ \vdots & \vdots & \vdots
\end{pmatrix}
\xrightarrow{\text{Enumerate}}
\begin{pmatrix} \vdots & \vdots & \vdots \\ a & b & \vdots \\ \vdots & \vdots & \vdots
\end{pmatrix} = B
\]

- Permute \(A_0\) according to 
  \[C = (m \cdot a + b) \mod 26\]
  Call this permuted vector \(A_{(a,b)}\)
- The \((a,b)\)'s with higher inner product \(\langle A_{(a,b)}, B \rangle\)
  are more likely to be the key.

Breaking Vigenère Cipher
- Too many possible keys
- "E" could be mapped to \(Z, I, G, X, S, U\) in the example (key = "vector")

Step 1: 
Algorithm for find key length:

- Count the number of collisions.
  Higher # of collisions \(\Rightarrow\) displacement is more likely to be the key.

Why does it work?
- Assume each letter in plaintext is drawn from \(A_0\).
- Assume each number in encryption key is different.
Q: What is the probability of a collision?

(When one letter is shifted by \( i \), and the other letter shifted by \( j \))

\[
\Pr[\text{resulting letters are the same}] = \Pr[\text{both letters are } "A"] + \Pr[\text{first letter is } "A" - i \land \text{second letter is } "A" - j] + \ldots
\]

\[
= A_i(0) \cdot A_j(0) + A_i(1) \cdot A_j(1) + \ldots = \langle A_i, A_j \rangle
\]

Algorithm works because if we guess the correct key length.

plaintext

<table>
<thead>
<tr>
<th>key</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 0 18 2 0 18</td>
</tr>
</tbody>
</table>

\[
\Pr[\mathbf{x}_1 = \mathbf{x}_4] \iff \langle A_2, A_2 \rangle \text{ large}
\]
\[
\Pr[\mathbf{x}_1 = \mathbf{x}_3] \iff \langle A_2, A_{18} \rangle \text{ small}
\]
Step 2: Algorithm for finding the key:

- Suppose key length = 6. If we look at letters in position 1, 7, 13, 19...
  it is as if they are encrypted by a shift cipher.
- In general, let n be the (guessed) key length.

  For \( i = 1 \) to \( n \)
  
  \[ B = \text{frequencies of ciphertext letters at positions } (i \mod n) \]
  
  Let \( k_i \) be the \( j \) that maximizes \( \langle A_j, B \rangle \)

End for

- The key is probably \( (k_1, k_2, \ldots, k_n) \).