Part III
Periodic polyalphabetic substitution ciphers

Unit 29
Periodic polyalphabetic substitution cipher

A *periodic polyalphabetic substitution cipher* is cipher in which there is a set of key alphabets which are used in cyclic order as a text is enciphered or deciphered. The number of key alphabets is the *period*.

Let's run through an example. Suppose we want to encipher the message

```
SECRET MEETING TONIGHT PREPARE THE VEGAN PIZZAS
```

with these five randomly generated key alphabets, which are written under the plaintext alphabet for convenience:

```
     abcdefghijklmnopqrstuvwxyz
0    GAEUPDOXKYTZJWIMBQVHRCSNLF
1    FXBNKVWQAJLECTMHPOSGIRUYZD
2    ZWCPVIHLFXOJEYNTGRDMKUQABS
3    LNVJIEPMADCQZOTYGXWKBSUHFR
4    SUNXTEKWQZLVRIDJACGFBPOYHM
```

To make things more clear, we can label the characters of the message by which key alphabet we will use to encipher each. The first letter 'S' is enciphered with the first alphabet to 'V.' The second letter 'E' in enciphered with the second alphabet to 'K.' The third letter becomes 'C,' the fourth letter becomes 'X,' the fifth letter becomes 'T,' and the sixth letter is enciphered with the first alphabet to an 'H.' We cycle through the five key alphabets until we reach the end of the message.

```
SECRET MEETING TONIGHT PREPARE THE VEGAN PIZZAS
012340 1234012 3401234 0123401 234 01234 012340
VKCXTH CVIFKTH KDWAHMF MOVYSQK MMT CKHLI MASRSV
```

Python tips

In Python, an array (list) can contain just about anything, including strings or other arrays.

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; chapters XII-XV and XVIII.

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, chapter 4 and pages 236-239.

Auguste Kerckhoffs, "La cryptographie militaire," *Journal des sciences militaires* IX (1883) 5-39 and 161-191, www.petitcolas.net/kerckhoffs/crypto_militaire_1_b.pdf, www.petitcolas.net/kerckhoffs/crypto_militaire_2.pdf, part III.

Programming tasks

1. Write a function that takes a plaintext and a set of key alphabets and returns a ciphertext. Use your functions for monoalphabetic substitution if you wish.

2. Write a function that takes a ciphertext and a set of key alphabets and returns a plaintext. Use your functions for monoalphabetic substitution if you wish.

Exercises

1. Take these three key alphabets and encipher the following text.

```
LBRUVCJAWZYSHXINOQEPFTGKDM
SLNAXDIGOBKCEYQHTMWJFUVPZR
IFWVBXNGKHZQYOELPCDTJRUSAM
```

```
In his book published in eighteen sixty-three, Kasiski
presented a method for finding the period of a polyal-
phabetic substitution cipher. The method uses the
positions of repeated sequences of letters in the
ciphertext.
```

2. Decipher the following ciphertext with the same key as in the previous exercise.

```
IYBIDTAXYIWTNQLFCIQHESZISHGLLBPOWROLAXCGSDPGIPQXBCIWWB
UXRWIBXXCVOTGSDCOCEJLFLQWWGVAKXDKCJBVYBWIGPZXWUBBFTGSD
RQOEOVVMBUFOBMBLKIBCBFYTWCBWIGPXBXWKKJAPGCVX
```

Unit 30
Finding the period: Kasiski examination

The *Kasiski examination* is a method for finding the period of a periodic substitution cipher. It involves finding repeated sequences of letters in the plaintext. When more than one repeated sequence can be found, the period is likely to be a common factor (possible the gcd) of the distances between them.

Let's look at an example. Consider this ciphertext:

```
THZBAROLASYZFKHFNYCEYXOQMWHXLELXLAUHNPMIAZTLVDWNNHRDOW
SIHUCCMGNTTTCWSIHUCCMHTEEDCBUGMHZBAROLTSONNSHUDWQFZXRP
NABMHTZDPRYHUCMMNTWADUBUKAOCCMUKELRSDREHULXIAYPECDPNZR
OFVTRTWOCCMUKLAWGILYHNLCBRGWYNYCEYXTLVSGUFIDDMEKW
```

Notice that the sequence `ZBAROL` occurs twice. The distance from the 'Z' of the first occurence to the 'Z' of the second is 84 letters. The sequence `SIHUCCM` also occurs twice, with a distance of 14 letters. The sequence `OCCMUK` occures twice, with a distance of 35 letters. The greatest common division of 84, 14, and 35 is 7, so the period is likely to be seven.

Reading and references

Friedrich Kasiski, *Die Geheimschriften und die Dechiffrir-Kunst*, 1863; digital.onb.ac.at/OnbViewer/viewer.faces?doc=ABO_+Z224431001

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; chapter XIV.

Fletcher Pratt, *Secret and Urgent*, New York: Bobbs-Merrill, 1939, 147.83.93.163/cops/fetch.php?data=4538&type=pdf&id=2942, chapter IX, section II.

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, pages 207-210.

Programming tasks

1. If you think that it will be possible to write a function that employs this method, go for it.

Exercises

1. If you wrote a script for this method, test it with the example ciphertext.

Unit 31
Finding the period with the index of coincidence

A ciphertext that has a period m is like mixing m different ciphertexts, each enciphered with a different key. As a result, the chance of picking two characters randomly and obtaining identical letters is reduced, as compared to an unencrypted English text. Without proving it, we state that the measured index of coincidence of the entire ciphertext is related to the period in this way:

$$\mathrm{IoC} = \mathrm{IoC}_{\mathrm{random}} + m\,(\mathrm{IoC}_{\mathrm{English}} - \mathrm{IoC}_{\mathrm{random}})$$

Solving for the period and using $\mathrm{IoC}_{\mathrm{random}} = 1$ and $\mathrm{IoC}_{\mathrm{English}} = 1.75$ (in the normalization that we use), we obtain

$$m = \frac{\mathrm{IoC} - 1}{0.75}$$

In tabular form:

IoC	period
1.75	1 (monoalphabetic)
1.38	2
1.25	3
1.19	4
1.15	5
1.13	6

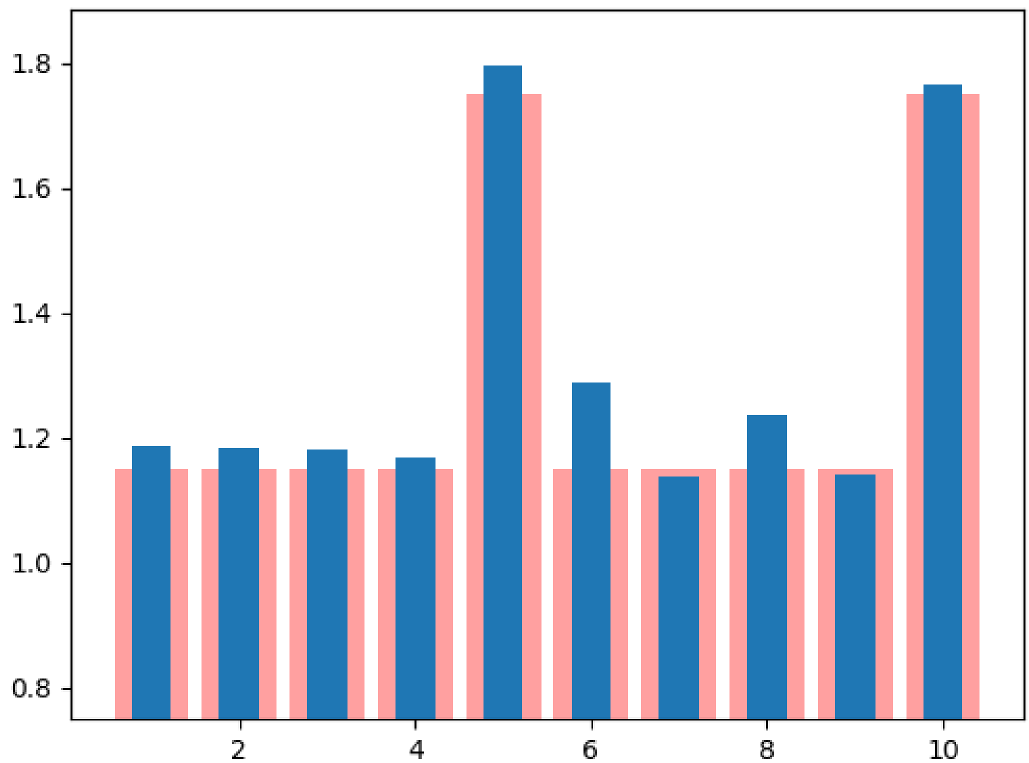
As we can see, as the period increases, the values of the IoC get closer together. Since these numbers are approximate and there is a lot of variability in the IoC, *we should not rely on this method for finding the period*.

A better way to use the IoC to find the period is to guess a value for the period m and then slice the ciphertext into m slices and find the IoC of the slices. The n^{th} slice is composed of every m^{th} character, starting with the n^{th} character. Partitioning the text this way gives us slices that would each be encrypted with one key alphabet, if we guessed the period correctly. The average of the IoCs of the slices serves as a good measure of the IoC of the plaintext that you would obtain by deciphering the ciphertext with the guessed period. If this IoC is not close to the IoC of typical English text, then the guess is a bad one. By guessing periods until the IoC calculated in this way is close to that of English, we can find the period, and we are usually correct.

Let’s look at this ciphertext, for example:

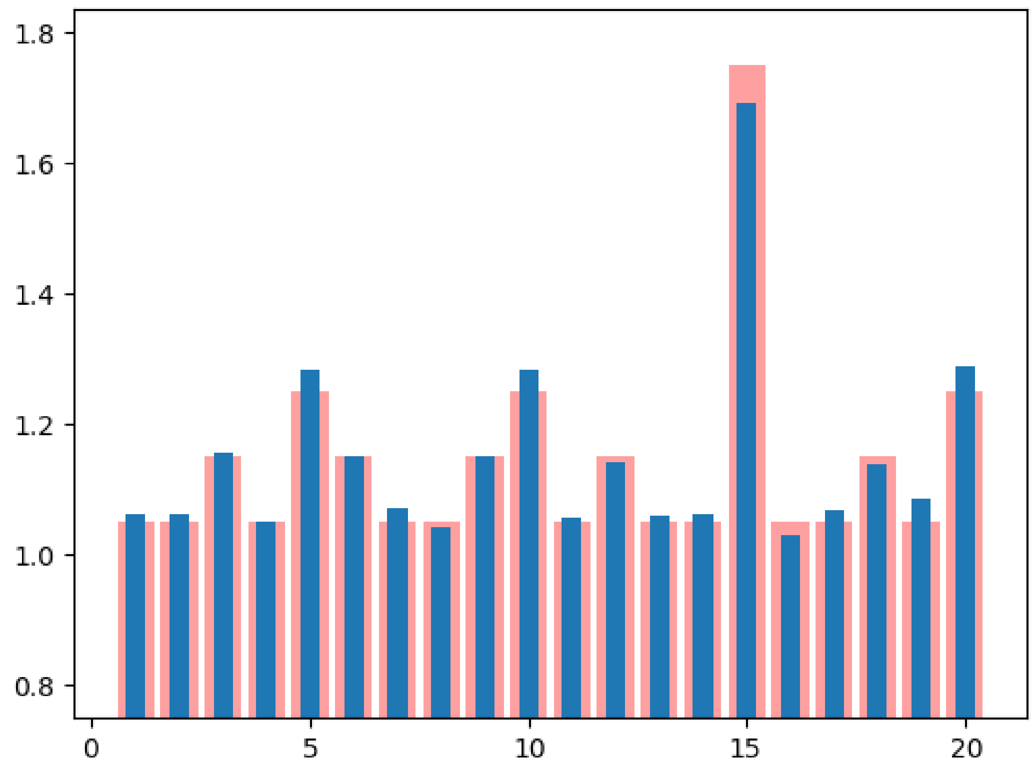
```
BUHMKLRASCKBLRZQQHRZMVVZBZLXWBNHOMKKEBTQWTUEMPLWLBQAGI
UWSFSFVPLHBVPHGXVOHYPMQWSCQAGXEMCHVFWQRJXXRUMLLVFTLTLD
PMPNEIQQPVYYAGLXRBVRRZQCKIOBUHFBAGZEVBBXWBBUHMKLRASCKB
LRZQQHRZMGRJFVQWLBXRUMLLVVXLKHWXEMPLTEMEWIUBVQXLAYLGBA
NQHXDRUEDMGKIFWPRJQPRVPFKRVMCBUHESMEDKBQBFMPKYRWBBBWLX
BBIIKOYLWEBUHRQPRQYJJRUSCAYLGBAVVPFSROCQWOHXEMCHVFWQ
```

And here is a graph of the IoC averaged over the slices, for periods one through ten. The measured IoC is in blue and the theoretical values for a period of five is in pink. The peak at ten is due to the fact that when we take ten slices of the ciphertext, then each slice is half of one of the slices that we took for period five. We conclude that the period of the cipher is five.



But the astute reader will notice that five is a prime number. What happens if the period is not prime? Below is a similar graph for a text with a period of 15. Again, the measured IoC is in blue and the theoretical values are in pink. The secondary peaks at periods 5 and 10 and the tertiary peaks at 3, 6, 9, and 12 are due to the fact that when we take a number of slices that shares a factor with the true period, then each slice contains some letters that were enciphered with the same key alphabet, i.e., that are in the same “true” slice. The expected theoretical value at period *n* when the true period is *m* is

$$\mathrm{IoC}_{\mathrm{theory}} = \mathrm{IoC}_{\mathrm{random}} + \gcd(n, m) \cdot (\mathrm{IoC}_{\mathrm{English}} - \mathrm{IoC}_{\mathrm{random}})$$



Reading and references

William F. Friedman, The Index of Coincidence and Its Applications in Cryptography, Riverbank Laboratories Department of Ciphers Publication 22, Geneva, Illinois, 1920, www.marshallfoundation.org/library/methods-solution-ciphers

William F. Friedman and Lambros D. Callimahos, Military cryptanalytics, Part I, Volume 2, Aegean Park Press, 1956, reprinted 1985.

M. Mountjoy (1963) The bar statistics, NSA Technical Journal VII (2, 4).

Practical Cryptography:
practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, pages 376-380.

Programming tasks

1. Write a function that cuts a ciphertext into a number of slices as described above. If the number of slices is n, then the first slice contains every n^{th} letter, starting with the first letter of the text; the second slice contains every n^{th} letter, starting with the second letter of the text; etc.

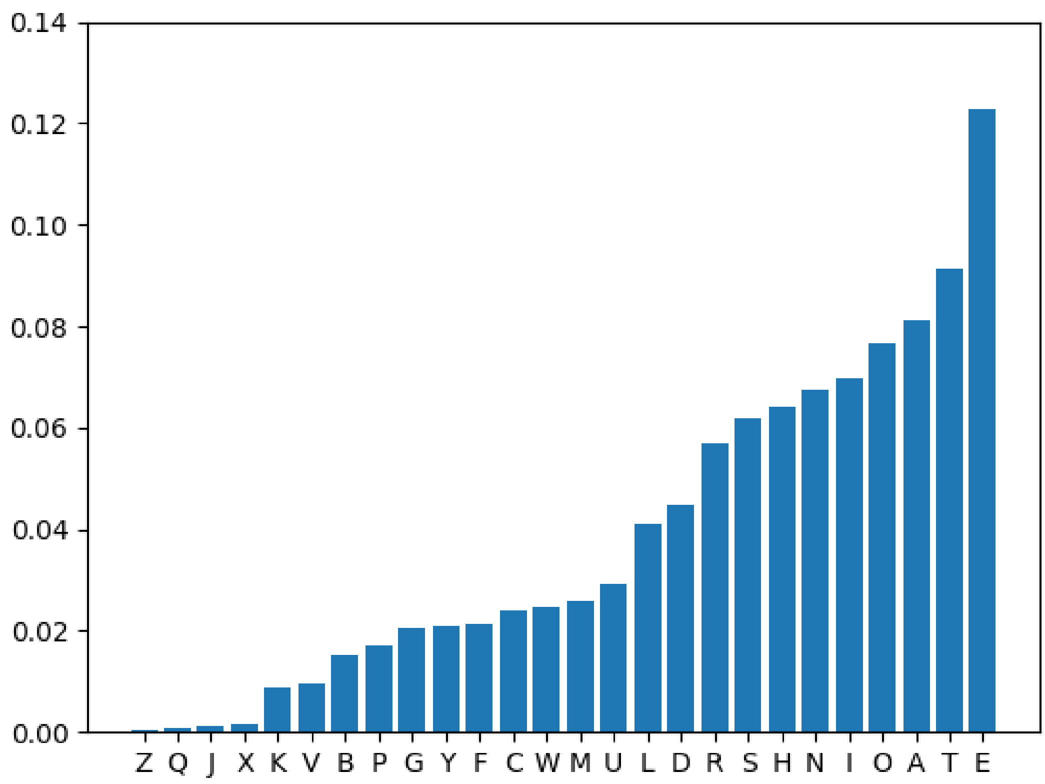
2. Write a function to find the period of a ciphertext with the method described above. You will need to decide on an appropriate cut-off, which you can base on the results of the exercise in Unit 10.

Exercises

1. Use your function to find the period of the example ciphertexts from this unit and the previous unit.

Unit 32
Finding the period: twist method

The *twist method* was published in 2015 by Barr and Simoson. Let’s see if we can understand it. First, look at the monogram frequencies of English, sorted in ascending order:

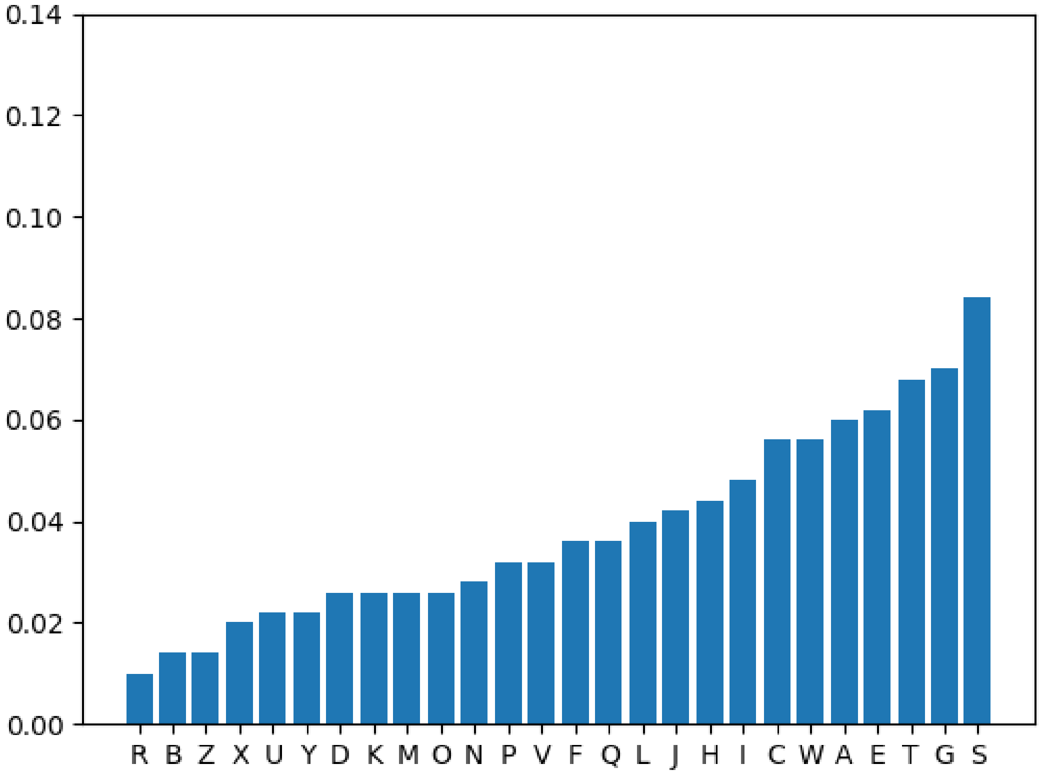


Now let’s look at this ciphertext, which was enciphered with period five:

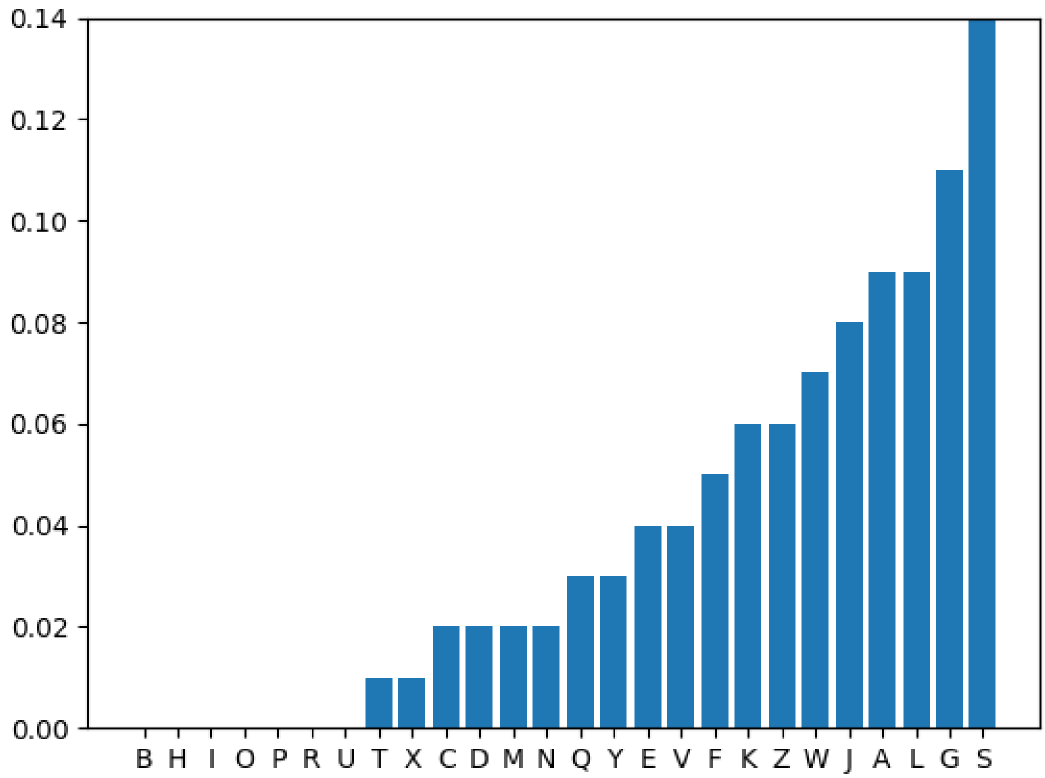
```
ZTSCALWAVEXTWAEJSSCASNSVSGSAUQSALEVSUTQJKDMGOACDCWEPLN
LGKETGJPFVEJIHWVYJEUWWSIVWZTDCHABPQSDDFNMYWTCVGJNFMLIH
GQSCLQSCTDCXSGTJYJHAFPQXTUIWBEFCGJCJSGHECGGADPMYWTVSNX
SKXGJRFISSPNEFTTJIKPIFAZDWJSSGEASMHTCQETRGHSGTJYJIHGQS
CLQSCTDCASNAIEACAMMFSOHWSSNGWKHEGQWSTQGJDSULAHFCGWBYPE
```

```
ETIURGIIOTGGTCRLWEUEASHGWWTMGHLDHCZWHOOILWIPKGCHKWEXNF
GGCVGVKPTKSFLAUGDTATPQHOOILWIPKZTFGPLWEFMVCTJENTTQVMHH
CXSGTJYJUENXSLKYEJSIGVQDUUXSGTNIVBEJIKPIFPSBENCLWEOEFA
OQOWSRQYFSTQLABAIEACAPHKAIILLAYTEAHEFLAHEAITGOYWZBMOQZ
TSCMVXSCMVNOWW
```

Here are the monogram frequencies for it:



Notice that it is much flatter than the previous graph. But, now, here is a graph for every fifth letter of the ciphertext:



The slope of this graph resembles that of English. This is the basic idea behind the twist method: If we divide the ciphertext into slices so that each slice has been enciphered with the same key alphabet, then the sorted monogram frequencies of each slice will resemble English.

Now we need to develop this into an algorithm that we can use in a program. Following Barr and Simoson, we define the *signature* of a set of of letters (a text) as the sorted list of its monogram frequencies. The *twist* of two signatures $A = \{A_i\}$ and $B = \{B_i\}$ is defined as

$$A \Diamond B \ = \ \sum_{i=0}^{12} (A_i - B_i) + \sum_{i=13}^{25} (B_i - A_i)$$

Notice what this does: it adds up the amount by which A exceeds B in the lower half and the amount by which B exceeds A in the upper half of the graph.

To find the period, we try various periods n. For each trial period, we slice the ciphertext into n slices, where each slice takes every n^{th} letter from the text starting from a different point. For example, with the sample ciphertext above, if we try a period of five, then we assign letters to slices like this:

```
ZTSCALWAVEXTWAEJSSCASNSVSGSAUQSALEVSUTQJKDMGOAC...
01234012340123401234012340123401234012340123401...
```

For each slice, we find its signature. Then, we average the signatures and take the twist of English monogram frequencies with the average signature. The trial period for which the twist is the greatest is likely to be the true period of the cipher.

Python tips

Arrays (lists) can be sorted with the `sort()` function, like this:

```
myArray = [1, 3, 2]
myArray.sort()
```

Reading and references

Thomas H. Barr and Andrew J. Simoson, “Twisting the Keyword Length from a Vigenère Cipher,” *Cryptologia* 39:4 (2015) 335-341, DOI: 10.1080/01611194.2014.988365

Seongmin Park, Juneyeun Kim, Kookrae Cho, and Dae Hyun Yum, “Finding the key length of a Vigenère cipher: How to improve the twist algorithm,” *Cryptologia* 44:3 (2020) 197-204, DOI: 10.1080/01611194.2019.1657202

Programming tasks

1. Write a function to find the signature of a piece of text.
2. Write a function to find the twist between two signatures.
3. Write a function to find the period from a ciphertext using the twist method. Feel free to use the function that you wrote for slicing a text in the previous unit. You might want to write a separate function for averaging signatures.

Exercises

1. Use your function to find the period of the example ciphertexts in Units 30 and 31 and in this unit.

Unit 33
Vigenère cipher

The *Vigenère cipher*, which was actually invented by Bellaso, is our simplest and one of the most constrained periodic polyalphabetic substitution cipher. Essentially it is a periodic Caesar shift cipher. The key alphabets are shifted versions of the regular alphabet, and the key is the set of shifts, which is usually expressed as the equivalent letters by a keyword. For example, if we want to use the keyword `SPACE`, then the key alphabets are

plaintext:	`abcdefghijklmnopqrstuvwxyz`
0	`STUVWXYZABCDEFGHIJKLMNOPQR`
1	`PQRSTUVWXYZABCDEFGHIJKLMNO`
2	`ABCDEFGHIJKLMNOPQRSTUVWXYZ`
3	`CDEFGHIJKLMNOPQRSTUVWXYZAB`
4	`EFGHIJKLMNOPQRSTUVWXYZABCD`

And here is how we might encipher a secret message:

```
PSST THE UNIVERSE IS REALLY BIG PASS IT ON
0123 401 23401234 01 234012 340 1234 01 23
HHSV XZT UPMNTRUI AH RGEDAY DMY EAUW AI OP
```

We can also understand the Vigenère cipher in terms of modular arithmetic. If we express the key as an ordered collection of L shifts $\{k_i\} = k_0, k_1, k_2, ..., k_{L-1}$, then encipherment of the plaintext $\{p_i\}$ to a ciphertext $\{c_i\}$ with a key $\{k_i\}$ is done with this equation:

$$c_i = p_i + k_{i \bmod L} \mod 26$$

and decipherment by

$$p_i = c_i - k_{i \bmod L} \mod 26$$

Note that if the period is one, then the Vigenère cipher degenerates to a Caesar cipher.

Some prefer to use a full table of all twenty-six possible ciphertext alphabets. This table is called a *tableau* (the plural is *tableaux*) or *tabula recta* (“right table” or maybe “square table”). The tableau for the Vigenère cipher is this:

key	plaintext alphabet
	abcdefghijklmnopqrstuvwxyz
A	ABCDEFGHIJKLMNOPQRSTUVWXYZ
B	BCDEFGHIJKLMNOPQRSTUVWXYZA
C	CDEFGHIJKLMNOPQRSTUVWXYZAB
D	DEFGHIJKLMNOPQRSTUVWXYZABC
E	EFGHIJKLMNOPQRSTUVWXYZABCD
F	FGHIJKLMNOPQRSTUVWXYZABCDE
G	GHIJKLMNOPQRSTUVWXYZABCDEF
H	HIJKLMNOPQRSTUVWXYZABCDEFG
I	IJKLMNOPQRSTUVWXYZABCDEFGH
J	JKLMNOPQRSTUVWXYZABCDEFGHI
K	KLMNOPQRSTUVWXYZABCDEFGHIJ
L	LMNOPQRSTUVWXYZABCDEFGHIJK
M	MNOPQRSTUVWXYZABCDEFGHIJKL
N	NOPQRSTUVWXYZABCDEFGHIJKLM
O	OPQRSTUVWXYZABCDEFGHIJKLMN
P	PQRSTUVWXYZABCDEFGHIJKLMNO
Q	QRSTUVWXYZABCDEFGHIJKLMNOP
R	RSTUVWXYZABCDEFGHIJKLMNOPQ
S	STUVWXYZABCDEFGHIJKLMNOPQR
T	TUVWXYZABCDEFGHIJKLMNOPQRS
U	UVWXYZABCDEFGHIJKLMNOPQRST
V	VWXYZABCDEFGHIJKLMNOPQRSTU
W	WXYZABCDEFGHIJKLMNOPQRSTUV
X	XYZABCDEFGHIJKLMNOPQRSTUVW
Y	YZABCDEFGHIJKLMNOPQRSTUVWX
Z	ZABCDEFGHIJKLMNOPQRSTUVWXY

In the tableau we have highlighted the encipherment of the first letter of our example message.

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; chapters XII and XV.

Blaise de Vigenère, *Traicté des chiffres ou secrètes manières d’escrire*, Paris: Abel l’Angelier, 1586, HDL: 2027/ien.35552000251008, gallica.bnf.fr/ark:/12148/bpt6k1040608n, gallica.bnf.fr/ark:/12148/bpt6k94009991

Wikipedia: en.wikipedia.org/wiki/Vigenère_cipher

Practical Cryptography: practicalcryptography.com/ciphers/vigenere-gronsfeld-and-autokey-cipher

Crypto Corner: crypto.interactive-maths.com/vigenegravere-cipher.html

Fred B. Wrixon, *Codes, Ciphers & Other Cryptic & Clandestine Communication*, New York: Black Dog & Leventhal, 1998, pages 207-211.

Giovan Battista Bellaso, *La Cifra del Sig. Giouan Battista Belaso* [sic], 1553.

Paolo Bonavoglia, “Trithemius, Bellaso, Vigenère: Origins of the Polyalphabetic Ciphers,” Proceedings of the 3rd International Conference on Historical Cryptology, 2020, ep.liu.se/ecp/171/007/ecp2020_171_007.pdf, DOI: 10.3384/ecp2020171007

Fletcher Pratt, *Secret and Urgent*, New York: Bobbs-Merrill, 1939; 147.83.93.163/cops/fetch.php?data=4538&type=pdf&id=2942; chapter VI, sections I-III, chapter XI, section I.

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, pages 148-150 and 240-242.

Programming tasks

1. Write a function that enciphers a text with the Vigenère cipher and a given keyword. Feel free to use your function that enciphers with the Caesar cipher, or to use the equation, or to use the tableau.

2. Write a function that deciphers a text with the Vigenère cipher and a given keyword. Feel free to use your function that deciphers with the Caesar cipher, or to use the equation, or to use the tableau.

Exercises

1. Encipher this text with the keyword PACMAN.

```
DOOT DOOT DOOT DOO DOO DOO DOOT DOOT DOOT DOO DOO DOO
DOOT DOOT DOO DOO DOO DOO DEET DEET DEET DEET DEET DEET
DEE WOCKA WOCKA WOCKA WOCKA EEEEEEEOOOOP GAME OVER
```

2. Decipher this text with the keyword VIGENERE.

```
OPKZVKVRZZKGVTYIMEGWSMIWOLKWPVZFZLHCTMFZVVHEGXZWOIHIYPR
WJQTJVJKIZVLMSXPXCZKINRUMOZKQNMEIYCTFESBIICTXVPKLZUOHRM
XLOMKRUYEHMMJWVXJVZXAXNXZSIMGVAIUMOBNIAMTOIISIYITLDNLVR
MEHZKNMSJIEWTKAUMTLDALVRRTLAWXXUIZRYMIMCLVVVJRIPMGLZZ
```

Unit 34
Brute-force attack on the Vigenère cipher

To do a brute-force attack on the Vigenère cipher, we need to try all one-letter keywords, then all two-letter keywords, etc., until we find an acceptable plaintext.

Python tips

The `product()` function from the `itertools` module can generate all possible tuples containing items from a set. For example, this block of code will create a list of all possible three-letter combinations:

```
from itertools import product
letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
combos = list(product(letters,repeat=3))
```

The length of the combinations (3, in the example) can be replaced with a variable.

Programming tasks

1. Implement a brute-force attack on the Vigenère cipher. Use tetragram fitness to decide whether you have found the correct plaintext.

Exercises

1. Use your implementation to break this ciphertext from the 2015 British National Cipher Challenge:

```
WLHJLVVXLXHQLRRYUPLXWPHEXGWMRRZMOPEIWLHPRGDXLSQSIEVEII
KSXWHMQXKIXOVIFXRVRJEIUPLRLXLWDQLRRVVXRTRZHVRRWLHVDXOM
QIVFXXBSXRHZHVNRRABSXQLKKXJIWPXGNCDRGJLRGWRQHSQILRWIUI
VXLRJLLHLRJXKIUIDXWLHZHVBPHEVXBSXALPOMQGRRYIQMHRFIWLHV
HMFLVHROWSUMICRYWENISSVWHWVMRRRJLXKSZQXGKARYOHWLDXEIZS
UXKXRCRYGSLLHEUEEMGSIJLZHLXRGVHHWLRYVEQHIVDRFWIVRQRYUJ
UIQGKJUMHRGWSIULDTVXKIEVLXLWKARYOHSEBQRVHSUQDCEIWLHCFE
QRRXDJISUHWSLARRGIULRAWLHCIIHPDFRYWXKEWTHVKESWBSXWKSXP
```

```
GEVOWLHQLJBSXADRWXRSXXEMGCRYUWRGDPOIGJUMHRGWWLHROIDZHX
KIPSQIBMQYQQDVNIGXUIDWXVBFLPOWLROSFOHVDXWLHJDVHRGSIXKI
SPDXISUQLRIVLIGVLGKWWVDWVILALPOPHEYIWLHHHXDMOWLROSFOHV
BSXALPOJLRGXKINIBMQHRRRXWVBXRHRYEPHGUSVWPILXZMOPQSWARV
NEQHRYUPLXWPHKDQHALPOIQHEIISUILXKEVIYIQTUSSIUPBFHKXR
```

2. Now try another ciphertext from the BNCC (2002). This one has a slightly longer key, but it may take a lot longer to find it. The time it takes grows exponentially with the length of the key.

```
BPPOFDATNWBDLJZOIACTQJJXTJZOTSIUQTPZLPZAJDQUUAXUBIBTFM
AVDMUTIUUIDOMQFGPGZPRNFDBPPQTOCTEBIQDBXCFANUTMNMBFDQBX
KPZKFDVJZOUTMFZOMUAIQVDDGQFQPZMOSQOQWONMIMTGANNKUBEBFD
KEMAZACDMVTQMJTIWQUPHMERZPYAPGBIMUQFWOFWBZIEPZFEAJZTPZ
LPZIOPAVSOFEBZACTTWVXLNQMUYMUTMSQIUZWPZIXQMLAVXQLOQAEM
GQXMBEMDAUFMTPZMBEQGQISFPFYIUQZJMTJEAIMTMIMTMGJZNMMUNM
JMQQUIWVXLCQUPEBVZNPDBVZIUQQGMABDMTGTUANBZIUFIDWWGZMSH
MTUEFDMUASOAKLADFDIMMVUQZOAZQQZIMXTPZPBIMUOIFMEOQHMZJZ
BIQDJOQOUBZANUTMTFWDWWGOPFYQDMTTUPBHMTFWSQLIQZFOPFYQTF
ZZUATGKIMXMQITMVUPQWQZTUWORWSFPFOCSUWVEUJZLBZLXUBIEWNM
VZRITOQOMBJZOBZLSMBIQZQXMBEQOSIQBTJOIUUWOEQOFPFYWEQZOI
WSXLJYAVDMZACXUTMMOSQMPRKPGZTQBIQXMQITGZFUVTGKIMBSMVTM
KUUWOUAOQKFEABDQMKUVFCBXIOPQIAXFFPBFIMXCOBTFMABZBOQATM
DPULFPQXUTMNMBNTFFWTMBJENZKWVDKVDQPEQUKKPZKFDVJZOQDWGN
ICNIHQAEQDJOMBZLZACXUTMNMBNTFFWTMBJENZYGNMAUQZTZMFPNPD
KVDZFZKZMABSMOFTFYIOUINECSQPFRMMFCOMJMQBPPQTOCTEBFDUTI
QUTGPGJVFITTQTRIJFPGGTTQZWMVUUUVEBPRKPGZTQMOPMBHWVDBPM
ZSUDFMBBZIHDMFYMOFNBHWVDICXMUAPJYQUDCTFEFGVEQZTFIOPWOQ
IOABIQZBZLUTIUACSYMFFQOSEJXTJZLFQLCQIQXMBECSQNPDCTNWUT
XMQITQZFBTZUVUTMVECBXNBEPJAVZACSEMT
```

Unit 35
Attacking the Vigenère cipher with cribs

Attacking the Vigenère cipher with cribs is not as easy as it is for the Caesar and affine ciphers. To use a crib, we subtract the crib from the ciphertext at some position, thus revealing a segment of what might be the key. If the crib is sufficiently long and the revealed segment shows some repetition, then we have a good candidate for the keyword.

Exercises

We will not ask you to program the attack, since it may be unnecessarily complicated.

1. Break this ciphertext with the crib `NATIONALSECURITY`.

```
JGVROAEAQNFRFEZGUFQAEAQNNOSPUWRVWYLYGNPBSAJKSZSRZYTAZF
OYLKNHHTZZCVRZOCDFWVGECARPYHEQXGCGVNPSTLLWWZEQNGKSLXVE
EXSQWFEEDLAJQSRFUEGTSPKACYGDAVAHZKSGOEMDQWRUEOOCRQVNZO
FEAZIEZGSCLOYSIENQDEZGFGRFRGXEEQMPFVPERPPJVYSRRMDQWVQG
EZGLVGOQXQFGKENDCNQHSEAPEFXRGWKLYDNNWRRBJRLEVHRFOXHCNL
WHLLUEPXRPVUNBZDPFUEYHCEJQNVFCZEOUALCLLKOAVWTLJJBXRYSN
IFWSLFFIAWECFCTVRNLDQFSLCTSNSUDSDZWTQRWYAVSRQCCQRTRGEX
SKLFHRGAEEF
```

Unit 36
Dictionary attack on the Vigenère cipher

If we have reason to believe that the keyword used to encipher a particular ciphertext is an actual English word, then we can perform a dictionary attack. Of course, the keyword might be a name, so we could use a list of known names and previous keywords as our dictionary.

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; pages 112-116.

Programming tasks

1. Implement the attack. Use tetragram fitness to determine when you have found the correct plaintext.

Exercises

1. Break this ciphertext. The keyword is a common English word.

```
ARVYIMZVGVQHFXJJWBFGGVHPSVFJRHOHGEFXYEPYMLOLKJXEWFNFAB
FKERQISMIEEZSMPJXMZIPRXBGRCCWXUYTZXRSKGEGRLLGWSKEITXSO
WVPDIGLGQEXKSGVFVASWGOTHKIFKLXSKGEGRKQCJWBNIQEPBFIGRZX
KHTFTIARIVJYGVVJEGVEVKIFHXUKSVAVELQOWRVVRVJCRKMHFYUVHM
GWGTYKWHKXMMSPEFQFMRKTEMASPJXAWPCKILLENCIZSXKFRLARFZGT
LIVYIGKEORRBHYNRXXVEPUAXSOGEIWSGTPTMGKTRTAQWVRRWSVFKLX
FEVZSGSPKEWMAXWKIHXWVRRWSVFJEGVXGTLGGPQXCASHKJWNWHUVZX
JENPITJWCXSTKETVWNDXPZWMDEUKAXWORLFEAGNPHBKGQLVTYIFKIV
ZGQDTTFMGJJKGQWJMGYXJRXVJCRKSZJERYMVSTRISTULCEHIJSOZWX
VXQXMOWXJVTNTPKTEGGTRFVMMRKKCMGAGZKAARQEEKWZKJIWKXCEHT
JHVYIYABORCGGXDVEEDXJRXWAJHZGNDXVYIMSMPKIWHETKSYLLGJXT
FHCIHBKEJZKADCKEIYXMEZIGLENXSKAXJDXASXUVGNJMVPIQHITKWB
VIPKMYAIFRWTHVQSPXEPQEKTYSKEJTUXVYIUAKIVWMECUKIKQXJFWX
```

WBRVVMKWCPMLOLAKLXFWCKLHMKJKEGQGQDTTFCQIKHNITEQXFXCXIG
UCYFYEVAKCPBFKNPYLWXJRXISVVZGNDETRPZGVKKLFLSRISMWGVKLX
AVFRXT

Unit 37
Hill-climbing attack on the Vigenère cipher

In this attack, we first find the period using the method described in Unit 31 or 32. Then we start with a key that is all 'A's. We start with the first letter of the key and replace it with each of the letters of the alphabet. The choice that gives the best textual fitness for the deciphered plaintext is kept. Then with the new first letter in place, we move to the second letter of the key and do the same to it. We continue until all letters of the key have undergone this process. Then we repeat again from the first. We continue in this way until the fitness can no longer be improved. This attack is very reliable, even for shorter ciphertexts.

The algorithm:

1. find the period m
2. set the key as m copies of the letter 'A'
3. set the current fitness to the fitness of the undeciphered ciphertext
4. set a flag equal to FALSE
5. while the flag equals FALSE
    a. set the old fitness equal to the current fitness
    b. for each position i in the key (i from 0 to $m-1$)
        i. set maximum fitness equal to current fitness
        ii. for each letter x in the alphabet
            - set the i^{th} letter in the key equal to x
            - decipher the text with the key
            - calculate the fitness of the new plaintext
            - if the new fitness is greater than the maximum fitness
                · set the maximum fitness equal to the new fitness
                · set the best letter equal to x
        iii. set the i^{th} letter of the key to the best letter
        iv. set the current fitness to the maximum fitness
    c. if current fitness equals old fitness
        i. set the flag equal to TRUE
6. output the key

Reading and references

Practical Cryptography,
practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher-part-2

Programming tasks

1. Implement the attack. Use tetragram fitness.

Exercises

1. Break this ciphertext.

```
HVSWYTMSBBDIYGKDIJSWNJVCWITWDMTIHSCIJHVFIRFSTCCMOKTNHW
RTJHVJAVBUYWFTHMTBSM
```

2. This ciphertext was encrypted with two Vigenère ciphers. The order in which they were applied is irrelevant. The result looks like it was enciphered with a single long key. The period is the least common multiple of the lengths of the two keywords.

   a. Find the period.

   b. Use the hill-climbing attack to find the plaintext. At the same time, you will find the combined key that looks like gibberish.

   c. Use the hill-climbing attack to “break” the key that you found in part (b), and recover the two keywords for the two Vigenère ciphers. It may be helpful to know that the lengths of the keywords are factors of the overall period.

   d. Now that you know the two keywords, decipher the text with two Vigenère ciphers.

   e. Repeat part (d) with the order of the two Vigenère ciphers reversed. Verify that the resulting plaintext is the same.

```
KBLFZROYITGKACGXWGSWSYOKTSYMRQZEPCZSRLAOWXUYRHVMTEFIQY
ZNVGULHCBZVBOKCJWVLKDCMNEXIYFNZLWFLTJBQFCUBFTEDBCXZDLZ
IMJFLAFSQZROCMNUKISZGOWOBWZLGVIIICTXMOZXCFRHKCWRZSPYAX
LJOIVPMAOLVRNUBFXEBIBWOGZGCIYZLJKHGLWYQWCRXHRUHAHQMFOB
SUFGQKBMCOCUNFQFERDRQLRDMXLRCTFQXEPLNHCYACOHJFBEDDERTI
JDOBUOLLNERICAMBSDVINVIZJHUJBRTGKAVXOOCHTXMUWGSOIIBXLR
GAZAVNCJMBOYRYJXXFBTMDD
```

Unit 38
Attacking the Vigenère cipher as a periodic Caesar cipher

Since the Vigenère cipher is a periodic Caesar cipher, we can use the technique of Unit 19 to break it. First, we find the period m with the index of coincidence or the twist method. Then we cut the ciphertext into m slices, where the n^{th} slice includes every m^{th} character of the text, starting with the n^{th}. Each slice contains characters that were enciphered with one Caesar cipher. So for each slice we can find the shift using the technique that we used to break the Caesar cipher and convert that shift into one letter of the Vigenère's keyword.

Reading and references

Practical Cryptography,
practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-vigenere-cipher

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, pages 210-213.

Programming tasks

1. Implement the attack.

Exercises

1. Break this ciphertext with the attack.

```
BNAGGUTAGGOTGNTALTNTVEWFZSMARGSGWGNWKTBTWEGQGQGWBTWAME
KTNBZKEHZRRGGMYOFXAFZZTVNTQFIAIGNTQHWTZEKKVCRSWUMSUJXQ
SGDEQAZGWMFVSBZVRDLMAJSGABYLHBUKOHZYJAAWCKLAIGYGFMBTWZ
MGYERURYKTOROFTJBZLEMNEWTZUGKIIFYWWAVTUXQJXGMMZEFHBROK
AWHRVAIIKCGWJTLAQFXAZPGLJHUGNWLBNXLHVYEZHXRISGSRKHFMGU
YXBUKJEWIKUTVZKFWGBAJEQSKTNBYUNXKNTTKMNQQHCENWTZGCSESR
JGNBGNALUBXFBVTOVHVGHWEQRBWPPNZALIJGZNVQXWWUVRDBWAHGMB
```

YKKPIFNWWCCUFMPRYZHZRYWXUFOEGWGGDHVROFUMVTYTTBTWTPHTVK
MQSAETVUFVIFZSPILYDHWXOFZNBXSAWZK

Unit 39 (optional)
Gronsfeld cipher

The *Gronsfeld cipher* is the same as the Vigenère, except that the key is a string of digits rather than letters. Because the key is made from digits, the largest shift is nine.

The attacks that we developed for the Vigenère cipher also work for Gronsfeld, except for the dictionary attack. Notice that they only needs to try shifts zero through nine for each digit of the key, rather than run over 26 letters.

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; pages 117-118.

Practical Cryptography: practicalcryptography.com/ciphers/vigenere-gronsfeld-and-autokey-cipher

Fred B. Wrixon, *Codes, Ciphers & Other Cryptic & Clandestine Communication*, New York: Black Dog & Leventhal, 1998, pages 213-214.

Fletcher Pratt, *Secret and Urgent*, New York: Bobbs-Merrill, 1939, 147.83.93.163/cops/fetch.php?data=4538&type=pdf&id=2942, chapter VIII, section I.

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, chapter 4 and pages 245-46.

Programming tasks

1. Write a function to encipher a plaintext with the Gronsfeld cipher and a given key. Feel free to simply make a wrapper around your function for the Vigenère cipher.

2. Write a function to decipher a plaintext with the Gronsfeld cipher and a given key. Feel free to simply make a wrapper around your function for the Vigenère cipher.

3. Write a function or script to brute-force a ciphertext enciphered with Gronsfeld. Feel free to copy your work for the Vigenère and make the appropriate changes.

4. Make a copy of your hill-climbing attack on the Vigenère and modify it for Gronsfeld.

5. Make a copy of your attack on the Vigenère as a periodic Caesar cipher and modify it for Gronsfeld.

Exercises

1. Decipher this ciphertext with key 78345024.

```
VCWHFMPIKASSYOWXPADCTNGXDWZLDTJIUQWMXTKQLBRHTIVLLTOMXM
WVRGIMJMAPVZGJNECWVTGMJRCRKIIIFRFAOIWRJEFALNHEWWJSRVRA
XIVAOMQRTNGGHVFEQLQYYXRAJRVSHKFSZNVCLBZLTWQYSLKEAEVLVC
JLYTJIVTGQFNVSOIYIMAFWVUXGMBNSVLLRMIO
```

2. Brute-force this ciphertext.

```
TNMDFHBUXUYQLOXBWWLDXMTDYQRMJRMTDUIVFXNSKFPYQPSJMTALXC
KFARKXYYQLXCYFRNMCMTADFAKUJAMFHDQHIKLJWSTBAFQTJQVBQHWM
HYQPSPPYRZYXZFHDMJABJZYQPXOVWNZYBHAJNJAVZPOFWKXCLWWDMR
JMRUYQLANYDCOTDNMCYJWLBBAMNMJJYXXINCAJAPXRAINHYQPXUPYC
SJVVWNIZCVKCOJPVTMATCYJJABQPHQAMNYJRMTDUIBWJJRBRSQRVKC
OJXAMNYYQPSPZNBHBCOJALNLHSWVYFLQUYJYLFCOTFAMNYJRLSCLWN
KXXMZUSBJZNXMXUBRKLWJAYQLRXTJWANWDMRJMROFMHGJUIXUJMAMN
AWDLBJF
```

3. Break this ciphertext however you please.

```
CVPJIWAGLSMGXWVLNLKXAQSEYNHBRHCYXHXHFORSXWKVMFROCJHPTC
HAURFTRGNWUOUKKJXHESZHAGGISZRJJRLDGOUKDNHKWXGAOWILBGWH
NYVYGFTWKSXRHASJFICUMJZWNTQIHNDWFJUNFRCHUESIKACGIHGOBC
QDLKUVCQPTRLLGVPNKVFJHRJOVCJHPFXWHKQOUVRBKWITTWQCWHFYQ
XVEEJTGNEUIJJDBERMRAQRUWIHHBRVVOUVRWGQTXOQYQZEWCKNTHIX
ZKNQSPTYLCKRNYNDCJDSSUWQWULJJEJENTMKEACQDNTJAGSRTGFQQI
CTSPDPLSRGJJKQSYZKNORRJGGECQCJJRYRRSNZLXPSAWZLNUDSBKOU
CVALGLWUWIYYUNCFTNUQJTBAIBHAUDRNKV
```

4. Break this ciphertext.

```
NQRTSGRRTSGFQHCWYWSNBFLPWPTZIRMLRSDTNWXLMGXRNQJVAEVYGS
WVYKILISHXOOFWLEZGOECSGHVRQJVFUBUHXEZBKSWIXYIRGSDYGPYB
VAVXWVAQLKXAEFBXOUWPGGZJJSRALDVDMSDRDAVXIEHJGYNLJUXHML
DXENNUWTPJDXEATPILMYWYCMXDRDATPIFZJQGHJJDRSISGXHMSKIAB
JVSMMWDHIAMHWAVIWLEVKHILQSJVABMHVSQHNLEEJQXTWQRSKNTUWO
UJSERAQHC
```

Unit 40
Beaufort cipher

The *Beaufort cipher* is a periodic polyalphabetic substitution cipher in which the key alphabets are shifted and reversed versions of the regular alphabet. Encipherment and decipherment are the same operation. The key is series of shifts, which is typically represented as a keyword. Each letter of the keyword gives the first letter of the key alphabet that it represents.

For example, let's encipher a short message with the key `BEAU`. The key alphabets are written here under the plaintext alphabet:

plaintext:	`abcdefghijklmnopqrstuvwxyz`
0	`BAZYXWVUTSRQPONMLKJIHGFEDC`
1	`EDCBAZYXWVUTSRQPONMLKJIHGF`
2	`AZYXWVUTSRQPONMLKJIHGFEDCB`
3	`UTSRQPONMLKJIHGFEDCBAZYXWV`

Here we have labeled each letter of the text with the key alphabet that is used to encipher it:

```
HARRY SAYS BEAUFORT CIPHERS ARE BEAUTIFUL AND STRONG
01230 1230 12301230 1230123 012 301230123 012 301230
UEJDD MAWJ DWUHZMDI CSFUAJC BNW TXEGBTZGJ BRX CINMHV
```

We can also understand the Beaufort cipher in terms of modular arithmetic. If we express the key as an ordered collection of L shifts $\{k_i\} = k_0, k_1, k_2, ..., k_{L-1}$, then encipherment of the plaintext $\{p_i\}$ to a ciphertext $\{c_i\}$ with a key $\{k_i\}$ is done with this equation:

$$c_i = k_{i \bmod L} - p_i \mod 26$$

and decipherment by

$$p_i = k_{i \bmod L} - c_i \mod 26$$

Notice that the two equations are the same, but with p and c exchanged. This means that encipherment and decipherment are the same process; a cipher with this property is a *reciprocal cipher*.

Here's a fun fact: The Beaufort cipher is the same as a Vigenère followed by an atbash cipher. Try it and see. The key for the Vigenère in this case is the Beaufort's key enciphered by the atbash. We can reverse the order of the atbash and Vigenère, so long as we also adjust the key.

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; pages 121-125.

Wikipedia, en.wikipedia.org/wiki/Beaufort_cipher

Practical Cryptography, practicalcryptography.com/ciphers/beaufort-cipher

Crypto Corner, crypto.interactive-maths.com/other-examples.html

Fred B. Wrixon, *Codes, Ciphers & Other Cryptic & Clandestine Communication*, New York: Black Dog & Leventhal, 1998, pages 214-216.

Fletcher Pratt, *Secret and Urgent*, New York: Bobbs-Merrill, 1939, 147.83.93.163/cops/fetch.php?data=4538&type=pdf&id=2942, chapter XI, section I.

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, pages 202-203 and 240-242.

Programming tasks

1. Construct the tableau for the Beaufort cipher.
2. Write a function to encipher a text with the Beaufort cipher and a given keyword.
3. Write a function to decipher a text with the Beaufort cipher and a given keyword.
4. Make a modified copy of your script for the Vigenère cipher that can perform a brute-force attack on a ciphertext encrypted with Beaufort.
5. Make a modified copy of your script for the Vigenère cipher that can perform a dictionary attack on a ciphertext encrypted with Beaufort.
6. Make a modified copy of your script for the Vigenère cipher that can perform the hill-climbing attack on a ciphertext encrypted with Beaufort.
7. Make a modified copy of your script for the Vigenère cipher that can attack a ciphertext encrypted with Beaufort by finding the individual shifts by matching monogram frequencies of slices of the text with frequencies from English.

8. If we encipher a text with a Beaufort cipher using the key `BEAUFORT`, then what must the key be to get the same ciphertext with a Vigenère followed by an atbash cipher? What must the key be if the atbash is done first?

Exercises

1. Decipher this ciphertext with key `HUSBAND`.

```
ZBKJAUMNBLHNFIDDABPCFHSIOMRSDRMXXUWHBSJSAXWQGBNFQSGAJW
VLZGFNVNXTGPWMWKNHOPGVKGQKOENQOGNBEFYDNEFWSZQJKIHCZXHE
FNUWDPOXPFQBCEKFFZLCEWITBAUGBNBDJTONNGVPPKKIUZUBOKSAXH
HOTUGCTABUMZAONKJHWJONKJIZHDJHWSQZEMFITJRZHPJMIKAQAHJW
PNHPTNHYHIKQSJLONSITJVPSEOIFADDOYHGZQMMUHIJWFBNLJMOWEW
IZRDGFXMWPONOKMIKAQKKXNJBNZXJV
```

2. Brute-force this ciphertext.

```
QEPHSWKMTGEWHMMGACOEJAEUERWXAJGWNPXMWTMFYLBAYKMWHGTSJS
FYPKLHHAMBRMGQNOGSFSCDKJAAAFBNELTKRUDEJZWYYTABLMKRHONW
ALOOQNAXMBASZXMYYTZWEEKPPMAKTARQGZEPKLHHAAFBIOAHLAESLW
BGLSNHEZHLAEENPPWHKMAWECKXAXAMNBJKJSFYMNZHLAIDTWORAFBJ
OYGHEHKRUZXWQWJQKPSLSERELANOJWBWVKRXGGIOTVMNMWWNMYJKSA
RQGZLTOSMYLTWXWFAFONSZWISBAGPXBWBTTCFQFOSZONSFSCAQGHIT
ORWXAJKZSFBOUMWHZSFJMHKNZEJKTCDEGAWNMDWNQJOCMNZSFIABAT
EKIOMAFBZBWNMWNMKLZXWBAABQVOJWBGVYRWBEPKSWOLAFBWAPWQWA
HTCWXWFAFONOUXCDQIMAHAKYLENYPLONTSRXEZOOLTSLSZNWCKSBAI
SMHBQNMSMBEPDNSFCSDTWZQLBAFORHGAVBQOPAPKDWBEHOTCALWDPS
FYSFLMZXWALJOAHSRXGAHLQXKCAHTCIRMQUSFYLOQLHAILEHAQVNLT
ORSSCYEKNZWHLWULLSGAHEYWZLMAAASMIEQNSMSQEN
```

3. Break this ciphertext with a dictionary attack.

```
WWOQTJIVANTKVMCOEILZMBYIGISXAZHDITZIEQARODBLNMKTXFECYS
JJVTQHOTILGSCMTFDVBAVMKTCGIJTXJRASNVOWTAWESFQVWVCRFDVE
RHWSMGFJVFAUAGBWHPVQXFLKTXJTKKABVKVMAISHACLYSWJDDOIFDH
YKUWBJLGSJENJUXMCDLKJCGSKVZTLDOIFDHERFDSMMNUAAZZFVMDCS
XYQBDHYKHPVFALWXXAUAOUZNTKJEATVFAICXZWUCXIJJFQXTNOBHQW
WIYQKHGKLTFKYNUAAFKAQAZLGSSMTCENYQDEJETCSLTXJFKKAVEKUC
GFQUEADHISTKWMRKUHVWVDCMMWIWIMMLAYSJTGZLGSSERKISUEATOX
TJWWMRUWVAQVHHINJIJFQBSZIELSZINSWLIBFD
```

4. Break this ciphertext with the hill-climbing attack.

```
ZDJPPAIENXVBYODKGKPZNZKTDIGZUDUMSIKZCUYWNGZAXNGZVMZCEH
LMXSODCWBWZYSMQZAWTTZIKVSQJWTJDFB
```

5. Break this ciphertext with the attack that matches monogram frequencies.

LTTFHWGKAFAASOUXHJRDSOZCNCWPRDJYSULDAGHGWZLKEKGKARKJHV
UXLZMVLQKKFEXJTHCJKOMVHXYTOBKGLXURWHMALXYFEQXJZVOJONPR
GVYTZDRRSBUELOEEIHYDSEWEQOLCSGTTKXYKICEZSKZKFEYDYKOQZG
AMXXYCBTMNDBJDOVQAYGPEATSNSGQGAAAZWKPRLZMOLQCRZAAPEKQC
QHTGLNOTSOQANWOYOOWFHWUXZAAODWKGQOWEFXOWGPEEIAAISPWDYO
RZUZRGXBWOSWUZPHUILHPCHOLGPOQYLEYDKVWESAJDOWQVHWJVEQLJ
XBKWNOXNJGJMFTLKDBWIENTOHRAGOEXJTKPGBBGVEKPRLDIDUEUXMC
WYPOKQDTRKDCBQZUQVLBZKMHWELXYKQOWDYSOQHEMNDPYDKVEEIJOT
PHGZXBYXPAAVXURHHZAXWAPRLDMDGOUXZAAZDVJCFBNDXCVKBETTRV
JGBZRLLSOTPTNFXTUXMHSXTMRGNQQFZALXKUYYXBUELO

Unit 41
Variant Beaufort cipher

The *variant Beaufort cipher* (also sometimes called the *German Beaufort cipher*, or simply the *variant cipher*) is the inverse of the Vigenère cipher. Encipherment with variant Beaufort is the same process as decipherment with Vigenère. In fact, by modifying the key, a variant Beaufort cipher can be converted to a Vigenère. The modification to the key is to apply an atbash and a Caesar shift of one.

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; pages 121-125.

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, pages 202-203 and 240-242.

Programming tasks

1. Construct the tableau for the variant Beaufort cipher.
2. Write a function to encipher a text with the variant Beaufort cipher and a given keyword. Feel free to merely make a wrapper around your Vigenère decipherment function.
3. Write a function to decipher a text with the variant Beaufort cipher and a given keyword. Feel free to merely make a wrapper around your Vigenère encipherment function.
4. Make a modified copy of your script for the Vigenère cipher that can perform a brute-force attack on a ciphertext encrypted with variant Beaufort.
5. Make a modified copy of your script for the Vigenère cipher that can perform a dictionary attack on a ciphertext encrypted with variant Beaufort.
6. Make a modified copy of your script for the Vigenère cipher that can perform the hill-climbing attack on a ciphertext encrypted with variant Beaufort.

7. Make a modified copy of your script for the Vigenère cipher that can attack a ciphertext encrypted with variant Beaufort by finding the individual shifts by matching monogram frequencies of slices of the text with frequencies from English.

Exercises

1. Decipher this ciphertext with key `TOWER`. What is the equivalent key for Vigenère?

```
YMTQWGQPCALIMJCVFLAVVEXXNHGXEOBXGDRSPYJMLDXDNZGRSQLZWD
NDMWPFLXZAHLMVOXSPXDNLZGDJUFVABZELQCOQVEWAAEPXDQVSQPOL
HJFURWOVDIOCHZHDJKZIECOQVOCHUVOWVDHKXYNYPZBUXAJAFLACVB
AWBHXMPCSQAEWKAASQLZXDNLZGDJUFVABZIEJCLPXKPVUROQLBPWLL
PLAAZQPBKLZIWCOUXWWKOVENKDELDULIHAHBYJILXPACKAAJHVGVDJ
PD
```

2. Brute-force this ciphertext. What is the key? What is the equivalent key for Vigenère?

```
IHPHYYEHPRYKFNDELKOGYLUIPYZTIWNOLRCCHNAAMALMPOIZTBAFOH
LFANAPHJKRNNACPOZWYIQNAIAHKFYTTLWOLZIHWRSLELOOHWLVAAOP
YUJDCJFLKNNKFCPSIIEFETNHEXESNWNWAAQWYQWSMETNENAPHYWRNE
SNDIGOEFBBUOIFDAFHWUNDQDOMASOZDYJDCOAJLEUNAHYEMKMYUEUN
SUCOWWUMADUPTBATCIEMQCBLUVHIWAXWETYIEHPAHZGURELESYPOMK
MUJYMPRUJGYYOHFEWPULAS
```

3. Perform a dictionary attack on this ciphertext.

```
POOAQRXNAJHEAZXOHRHOZZDJLBOTWXVWUMWDDKWPBAAHMGCWUMYOTJ
RBZLDOWTNNZROLITMOHWZMIXXKUBWNLHDYJJJEMXBWUMXUZONZNXHD
ITMWSUPHMUUZYXIAVINNLCKLLKGWJCHYQTCDLJJCQKWPDJUCITYHLJ
OEIYVAAQAYXRNWZNZONUUZAQAWQYNNFXQNOYCAYBKFBUMWFBKBMOCW
UMBATRXJPOJOBOOOADZIWABUVDPHVUUKUPARKXJRLQESITLELWPAXV
NPPCASNUACVCGIVMBPVWKRJGUHHWPYVKCDLKNADKXNJXKPMXXBAQAW
WUMWUMSADKBKINETIRBKHWZMIERWUMWLTSHLPAWTMYBDHAATPKPNHE
AWPKAAAQASMGWZAQAIZIAAHCEOVYUEL
```

4. Perform a hill-climbing attack on this ciphertext.

```
WXDJUOJOYHKPDGPTAWDWSIWPOAXVYSHXLGJGIGYONPAJPHTJCOAEGP
LOXKCYLDKCXEWALZUHSLZHTKRNEPVGJGDFKUTPAJOETZMSEPYCNLNL
FALDTQPEGKYJDIZCPUGLJASPDJWDKSLYEIZCUAGWUWIIALCOCLFASW
ALCLTOGHLNGSYOBWYJDYGGJTWWBWNRW
```

5. Break this ciphertext with the attack that matches monogram frequencies.

```
ETWZQJLEGVORLBGWDJTPGEIUZXADQQTZSTAAPXQKAGEMYMUAQDGVFB
QFZMOBEFSOQJLESOMEOQFETRCQAVEGZAVBIBCAKMFEPQKWZRZROPUP
SNGZQJSULMMAOFZMAGSQJZQQCAKMEFSQZIPGHAUPUYODWVIUZIWZQY
TWWBTREIGZAFPFJMQFLZVWZRHMKKMYWQVAZBHIZQFRLZVBTRZFZMDE
```

ZEWZQQETWGIRCQSASBZPSVPULBHGMFMGKGMAOOZMQEQGDIERGQJBIB
NTATPEPZAVFUPIGZXQHQJMAAWKKVAJHTABQJLEEWDRBGAMFNYPYMZG
WQLPMACAKMDRODGAQEPPDQWRONWBFRCFGZGALNGCFVYFZMYRLPGEEN
YPXQQYOEKMQXTZYNXBHQJAMAOOSBOUTZYJGGEQJNXVPETCFFYAOETV
EQKIFNETGUQJTFZPQEXALPQELZVPQYAQVPQEHULPTRCTGCERHAJSAE
CQSLFBSQJETRYFZMDRHMKVAGSUFOFBOA

madness's book on classical cryptography
unit 42: porta cipher
last modified 2020-10-18

Unit 42
Porta cipher

The modern versions of the *Porta cipher* (actually originally invented by Giovan Battista Bellaso) use a set of thirteen key alphabets. The key is again a keyword, but there are two common versions for assigning key alphabets to keyword letters. Below is their tableau. Notice that each of the key alphabets is *reciprocal*, i.e., it is its own inverse. Thus, the Porta cipher is a reciprocal cipher and is also its own inverse.

key (version)		plaintext alphabet
1	2	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
A/B	A/B	N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
C/D	Y/Z	O P Q R S T U V W X Y Z N M A B C D E F G H I J K L
E/F	W/X	P Q R S T U V W X Y Z N O L M A B C D E F G H I J K
G/H	U/V	Q R S T U V W X Y Z N O P K L M A B C D E F G H I J
I/J	S/T	R S T U V W X Y Z N O P Q J K L M A B C D E F G H I
K/L	Q/R	S T U V W X Y Z N O P Q R I J K L M A B C D E F G H
M/N	O/P	T U V W X Y Z N O P Q R S H I J K L M A B C D E F G
O/P	M/N	U V W X Y Z N O P Q R S T G H I J K L M A B C D E F
Q/R	K/L	V W X Y Z N O P Q R S T U F G H I J K L M A B C D E
S/T	I/J	W X Y Z N O P Q R S T U V E F G H I J K L M A B C D
U/V	G/H	X Y Z N O P Q R S T U V W D E F G H I J K L M A B C
W/X	E/F	Y Z N O P Q R S T U V W X C D E F G H I J K L M A B
Y/Z	C/D	Z N O P Q R S T U V W X Y B C D E F G H I J K L M A

Let's work through an example with each version. Here is a short message, which we encipher with the keyword `PORTA`.

plaintext:	`GIOVANNI DELLA PORTA PUBLISHED IN FIFTEEN SIXTY-THREE`
key letters:	`PORTAPOR TAPOR TAPOR TAPORTAPO RT APORTAP ORTAP ORTAP`
version 1:	`NPGMNGGQ ZRSSV GBKMV GHVSQJUYX QE SPZLNRG LQBGE MPIRY`
version 2:	`ZOJENHHN URRRS LBLAS LHURNBUXW NJ SOYBVRH MNGGF AZARX`

The modern versions of the Porta cipher are descendants of ciphers invented by Bellaso. Recently, his first cipher (from 1552) was discovered in Venice, Italy. Here is its tableau:

key	plaintext alphabet
	a b c d e f g h i l m n o p q r s t u x y z
A	N O P Q R S T U X Y Z A B C D E F G H I L M
E	Z N O P Q R S T U X Y B C D E F G H I L M A
I	Y Z N O P Q R S T U X C D E F G H I L M A B
O	X Y Z N O P Q R S T U D E F G H I L M A B C
U	U X Y Z N O P Q R S T E F G H I L M A B C D
B	T U X Y Z N O P Q R S F G H I L M A B C D E
C	S T U X Y Z N O P Q R G H I L M A B C D E F
D	R S T U X Y Z N O P Q H I L M A B C D E F G
F	Q R S T U X Y Z N O P I L M A B C D E F G H
G	P Q R S T U X Y Z N O L M A B C D E F G H I
H	O P Q R S T U X Y Z N M A B C D E F G H I L
L	M L I H G F E D C B A Z Y X U T S R Q P O N
M	A M L I H G F E D C B Y X U T S R Q P O N Z
N	B A M L I H G F E D C X U T S R Q P O N Z Y
P	C B A M L I H G F E D U T S R Q P O N Z Y X
Q	D C B A M L I H G F E T S R Q P O N Z Y X U
R	E D C B A M L I H G F S R Q P O N Z Y X U T
S	F E D C B A M L I H G R Q P O N Z Y X U T S
T	G F E D C B A M L I H Q P O N Z Y X U T S R
X	H G F E D C B A M L I P O N Z Y X U T S R Q
Y	I H G F E D C B A M L O N Z Y X U T S R Q P
Z	L I H G F E D C B A M N Z Y X U T S R Q P O

He used a 22-letter alphabet, which was all the rage in Italy at the time. Notice that all of the key alphabets are reciprocal, and so the cipher itself is also reciprocal. Notice also that there is only one key letter assigned to each alphabet, so that keywords are unambiguous.

We can modernize the *Bellaso 1552 cipher* by using the 26-letter English alphabet and putting the key letters into standard order to get this tableau:

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

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

Here is a short example of the encipherment of a message with this cipher. The keyword is `PLAGIA`.

plaintext:	`BELLASO BEAT YOU TO IT BY ELEVEN YEARS`
key letters:	`PLAGIAP LAGI APL AG IA PL AGIAPL AGIAP`
ciphertext:	`ATYSSFW QRUB LWF GH NG AJ RSWIKL LYSES`

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; pages 119-121.

Practical Cryptography, practicalcryptography.com/ciphers/porta-cipher

Paolo Bonavoglia, "Trithemius, Bellaso, Vigenère: Origins of the Polyalphabetic Ciphers," Proceedings of the 3rd International Conference on Historical Cryptology, 2020, ep.liu.se/ecp/171/007/ecp2020_171_007.pdf, DOI: 10.3384/ecp2020171007

Paolo Bonavoglia, "Bellaso's 1552 cipher recovered in Venice," *Cryptologia* 43:6 (2019) 459-465, DOI: 10.1080/01611194.2019.1596181

Augusto Buonafalce, "Bellaso's Reciprocal Ciphers," *Cryptologia* 30:1 (2006) 39-51, DOI: 10.1080/01611190500383581

Giambattista della Porta [Giovanni Battista della Porta] [Ioan. Baptista Porta], *De Furtivis Literarum Notis*, Naples [Neapoli]: Ioa. Maria Scotus, 1563, HDL: 2027/gri.ark:/13960/t37142x6g, book 2 chapter XVI.

Programming tasks

1. Write a function to encipher a plaintext with the Porta cipher and a given keyword. Allow for the possibility of choosing the version of the tableau. You can hard-code the tableau into your code, or you can find a way to generate key alphabets algorithmically or with shifts, or you can use the magic of modular arithmetic.
2. Write a function to decipher a plaintext with the Porta cipher and a given keyword.
3. Write a function or script to brute-force a ciphertext that was encrypted with the Porta cipher. Note that you only need to consider a subset of the alphabet when generating keywords. Use tetragram fitness.
4. Write a script to search your word list for words that match a keyword. For example, if your brute-force attack finds the keyword `SECQES`, then `SECRET` matches because it gives the same set of key alphabets.
5. Write a function or script to perform a dictionary attack on the Porta cipher. Use tetragram fitness.
6. Make a copy of your code for the hill-climbing attack on the Vigenère cipher and modify it to attack the Porta cipher.
7. Make a copy of your code that attacks the Vigenère cipher as a collection of Caesar ciphers and modify it to attack the Porta cipher as a set of monoalphabetic substitutions. You will not be able to use your Caesar cracker as part of this attack.
8. Implement an encryptor for the modernized Bellaso 1552 cipher.
9. Implement a decryptor for the modernized Bellaso 1552 cipher.
10. Implement a brute-force attack on the modernized Bellaso 1552 cipher.
11. Implement a dictionary attack on the modernized Bellaso 1552 cipher.
12. Implement the hill-climbing attack on the modernized Bellaso 1552 cipher. Feel free to copy and modify your attack from Exercise 6.
13. Implement an attack on the modernized Bellaso 1552 cipher that is similar to the attack in Exercise 7.

Exercises

1. Encipher this text in version one with the keyword `KEYWORD`.

   ```
   The Doors were a rock band from Los Angeles. The name
   was taken from the title of Huxley's book The Doors of
   Perception, perhaps because the band enjoyed
   hallucinogenic drugs.
   ```

2. Decipher this text in version two with the keyword `CIPHER`.

   ```
   KNRVTOIJAQDBUEAEIJKETYDCHWHRDCFWZEIJCTTVDJWWLEJIPJXOLZ
   UYNMYGHQXMTIPXRELAKQXHPBTNRSIBXNZSGQGRHJSWUINEWQMAIEOT
   IEZXWJKJTHPGCLTVJUWCRSIBXNRXOGUEARPUUKFEQQUPNJYMQCIKYQ
   IYQBWNHKROWSPCODIZQYOJAJRUOQSBCIPKHBZEIJSWFUIIISBPXVTI
   HQXZTBMFYDTYTKRXLJYWH
   ```

3. Brute-force this ciphertext and find a short English word that could be the keyword.

   ```
   RGKYIPFKZHFKJONSGYELLKNIIPMUYEIYZAYYEHRLNUEXYUEUULFORX
   NKFHVLFKWKNUJMQUKUIYLGKPZERGKYIPFKZHFKJWFTNPEUAPZYMUIP
   NMCHOLKEUYJCQPYOVUTYJPKYWLCMFTWMYOFKYHVIUYVYEMWGNDRLKP
   ENKKRTWGZZRGRLQPEEFAIOFTNLZYYHILGYYPOPYKFHVLJAYOWLXUKO
   IHFTJVNXIHFTJUEXYSFLNMJKNJLPIYZPOZNKNGKXFHIZLGYMRHEUUP
   KEJHXYJAIYKHYHELRXNKAONKNEFAAPUSXYRGJMWSUPENCHLKEYAXFH
   IUEXRMJUGINUIUEWNVNZFKNTWRRGPEFAIILKYOWLNIWGNSZHFKJUUL
   FREHAGWLJSWVZHFKJWFGKURGGUKMNKELFZJJLUIYFKFVUHENGUEYUL
   WGZUIYWWUUJLRWYOFPYYWXFHITWEYHEMWPEMAHUUINNIWGNSJLRDJT
   WSUHEYJHILFTNHKONKYHEZRNLKWMRHEWQHFLNIWGNSJCRMQLKKWPPO
   KHIWLKMYZMFIJMFLLPKEFAIXNWFKWMRGPGNYZLWWYHIXRHEXFHILWK
   NPZYWSOHIWUHJYKLWGZYEMIEAUCLAPKOJIWWNWFGJMIURGKLKONEVU
   CWFGJPJMFZWZNCUUINNIWGNSJHITLSKPGSNLVUUSGUEYULKOWMOHUX
   RGFGKONTJYUBNLWGZKNLKUPURGJMFGNLRXNHOMQYZHFKFINGRGP
   ```

4. Perform a dictionary attack on this ciphertext.

   ```
   SOOKZSUIFAEPAXWXAMUIGCWMJYFAQNKMNCENRBEUDJAIFQEARHKAWT
   BMKKDPXXIKAOBTEUAEYXIBNPPRNAEARVNBWLHILAVMQXYYAMBXXZTM
   PWZKACBAQRFUKRZSOROAFKMTBOZVAUCNNKBTJOEXACSOOKZSUIFAEE
   YTKWANUUWTYEKUFCVDZWNBAUCNNKBTJOEXPPAHZKACBIOWSCRVURED
   ZVIDEEZVUKAZCNNHVPJZOAAORAFLDPZRWPZMFOEXFWVSKKENZWNKHT
   AWFKDMLXEZMVBDFKRTJIIQWCAIIVRDPOZZMVUIFAEPAXWXAMUVQKWR
   VYFAQNKMNCEEYXCDETBJWTSTTIEKNXTTUPKPJWVZDCKLNURMKLJUAS
   KBXPSSDACPOCXXVZDCKLJRZDKSWOSPAIFQOALXWANMAXJLORZILB
   ```

5. Perform the hill-climbing attack on this ciphertext and find an English word that could be the keyword.

   ```
   TQWUPJPGIAHRFZUVNKYQHXXKFZJKEDXVLZCYCUXBIYQDJADEUYXROU
   QYMVNDFQAHYBKZRPYBZLIDNYCNDYTQGDAKYACGSCSVMMMDYPXDARVV
   HPXWDAYGYRCKILXAZKCSVRHVHPTCHPUYXBDAFGARIVGLIAPEQVRRSZ
   ```

```
PYXJGKCUXJIAZFVVSKKZCVGCTZMVOKCJMTYJNZVAPRHZWTJBHGSZBV
PLIVCTCUJRHBYGLVPPZWIAYCQYXBTXBK
```

6. Attack this ciphertext with your code that treats the Porta cipher as a set of monoalphabetic substitutions. Find an English word that could be the keyword.

```
IBPTSPAHRUDMEEESWPCGVONLETNDXATPPARNHAYQVZSGHDUFAVVULM
BRXGRXMESXSWDCQOYYVECVQOGYPJNDORJRXASOEVOHVFRNBDNVRMBK
OUVPOETDCRTEERSTNHBSVPNGVXGBNNBEOECDEUOJBWYDBZSQFDMQRE
BAIWREOZDQUYSIBDCSOSJGVGEUODUKWNBIZBAPSBAHF
```

7. Decipher this ciphertext with the modernized Bellaso 1552 cipher and keyword `PLAGIARIZE`.

```
OTYSBUMXDISETKWNPRLEOWBYDRREWFRTZYSYMBUNTHNLJHRYTNSEBS
NIMMRNXSRSQUUQMNSWRCSFDQVFRWRZNEQBFIKPGSNGMMLIZONGNRNW
WJRMYYSAXBERXVSKJZDIIKHPGPAUUEDYBORMJJRNUNRMUPRNVVENRM
ZYEUMIMCJXELBOUJBABDAYENLMDNXSBZRVVWVWGSGHRRPZHZKSVWSG
IJNRSGRKGAMSUCKDVLSVANUQWFTOBVPELJCJSKNRVVVWGSVUQJDGVW
LOVKWQIBHMKLOYXBRWDJBHVMNAPZHRXLBWWAPALXTMNXNADBTIBNYE
ANIVRQBEQHGBORHWXHUYMRANICWFRBWEQWHRRQRZJEMEHUDXFUEVPX
DISEFHRRXWLISPTHSFAJJKWCUHQZMAVZKSVWSGIJNKWWVLABVYVRXO
NGGXMGV
```

8. Break this ciphertext that was encrypted with the modernized Bellaso 1552 cipher. What is the keyword?

```
YITCPACDICCCRDYCDGOZPSNLGLNOTYRDPASGSHNQYSCKWOYACZJGOR
NTMFWUHGSRJRVDNPYFDSYLOIARCDANDXESSNNSJRDDHMSSONJYCZJB
WSBPILQVWBOIGGWLNVYYLFDHTOGGLHRHQFXKCRNLXTADSGPAXGKVHW
XWATAMHSTGPGOQJKMTAMGARDPETPUPNJFLPMKEYANDPBMVRWIUKLPY
VQASVLUBCTQHORTWNWUIBAYYBXNUXYNTKCNDSFDHGVMSSAGIJBPYHC
HYBRTGVQRHWXXVSRRYQKLPANSJRXTCRPKWTPEKIAXHIEFXWDGOHRDY
GVZRKCHWFNBZGTKTPEVXRMDQEKRGRGCRCYYLKWVNICKDJENTSAGRNT
IXSIGRFXAXHTAFQJOWPPSCKSVSRXSXDGHGPPJRSYFAJCDNQESJHHZP
BRWJODDPFXJVNJEKCHWUIBAIGKDDHOPTDCVRFCSSVSRVHRTXDHTOJL
DTZFCKBWVRCSFCDTFTYBKDPJFHEQWINEDRRVDTKKJGVLUBCGKWJQCP
YDQGRGPBAGGRCIGKNERTTYICCPKOJXVTAGGTRGKRTHXXWHGSGNHGLN
OHXJCWYBOPGVSUSGUWXYGLYQGOREBYCGWPQKCGGWDJTEJTWDANSARV
PEPTMDGYACLJDMLEKISKOHPBCJBOONNSAPCCTCDYKKNPKOJXVYYHAS
HNJTLTYJKNGXSARVSRGGGZHPBAQFMVTBTSALHDABLFRDORHTMKXYGB
TYRGCGNSARDPEFCAOKANOSATHYFLPLWUWQMDGKSYFLDPKTPEYPUZIW
BOPCZFDYHDPKHYACQFXKWYVSJRNDXPPIQEPVTYTQXRRPASHGOBSGXJ
SSVDSGBZERTPKRNZRSEJXSSRPPGKGMCHYFXGOGSYNWUCTFHTGWINTI
PHNSNREJXGOGSZJRDPEBPOBIIYVLGBVSNCDLMUGEBOSHGSHGYYGVYE
KYYMXDJCLJTGTACJLKHIBMDAOKABTSFSETIKCHRDPZFHBBMWFZXXJL
DABAAXHYFODXXKHGLPLWKHGXYTMMPNPSJDGQGFHGKZYFOPKWUIGLPY
BLPBTIGKJGNOHKWNGCHYSKNQGLPTQXRRPALBURUNSFSGARKIPBWURI
ACDKIYBYTRCIFLDRDHSNYPMKGC
```

Unit 43
Periodic affine cipher

Suppose we construct a periodic polyalphabetic substitution cipher in which the key alphabets are generated as if for an affine cipher. Then we have a *periodic affine cipher*. The key for such a cipher is a set of pairs of integers.

For example, suppose we want to encipher a message with a period of three, and want to use these affine keys (multipliers and shifts): 5, 8; 11, 2; and 21, 18. The key alphabets are these:

	abcdefghijklmnopqrstuvwxyz
0	INSXCHMRWBGLQVAFKPUZEJOTYD
1	CNYJUFQBMXITEPALWHSDOZKVGR
2	SNIDYTOJEZUPKFAVQLGBWRMHCX

And here we encipher a short message:

```
ANY MONOALPHABETIC SUBSTITUTION CIPHER CAN BE USED PERIODICALLY
012 01201201201201 201201201201 201201 201 20 1201 201201201201
IPC QAFACPFBSNUBWY GENGZMBEDEAP IWLJCH IIP NC OGCJ VCHEAJESCPLG
```

When the multipliers of all of the affine ciphers are the same, then we have a special case in which the cipher can be factored into a single affine cipher followed by a Vigenère cipher. Decipherment is in the opposite order. There are 26 choices for the affine cipher, for the 26 choices of the shift. For each of those choices there is one Vigenère key so that the combination of ciphers is equivalent to the original periodic affine cipher.

Programming tasks

1. Write a function or script to encipher a text with a periodic affine cipher with a given key.

2. Write a function or script to decipher a text with a periodic affine cipher with a given key.

Exercises

1. Encipher this text with key 3, 4; 5, 6; 7, 8; 9, 10; 11, 12; 25, 24.

```
AN AFFINE PLANE IS A SET OF POINTS AND A SET OF LINES
SUCH THAT ANY TWO POINTS LIE ON EXACTLY ONE LINE AND
GIVEN A LINE AND A POINT NOT ON IT THERE IS EXACTLY
ONE OTHER LINE THROUGH THAT POINT THAT DOES NOT
INTERSECT THE FIRST LINE
```

2. Decipher this ciphertext with key 11, 9; 9, 8; 7, 6; 5, 4; 3, 2.

```
HTEWPQSVNSRMGRWXEATJNAVYCKGVYEJFIVXBFINOOSDWQUSGAHBEQQ
MJVYSPQCCWFOIXYPBYEWBAJJNCKTGQEVADBOHNFYAWCJ
```

Unit 44
Attacking the periodic affine cipher as a collection of affine ciphers

In Unit 38 we built an attack on the Vigenère cipher by partitioning the ciphertext into slices, each of which was encrypted with the same Caesar shift cipher. We then used the attack from Unit 19 to break each of the Caesar ciphers by using monogram frequencies. Here, we will do the analogous thing and partition the ciphertext, but use the technique from Unit 25 to break each affine cipher with monogram frequencies.

Programming tasks

1. Implement the attack.

Exercises

1. Break this ciphertext:

```
EMNMGYUQNIXTEMNMGYUQNVGUKOYJUZKRKCINNCEKZLGLGXMEUJXKFE
UKYJUNJEXKRETSGEMKQNLUUPGQXCVOAXYCHDKJXHUPGQXMAKPNAXAL
QSUHFVWJYVSKWDSFGEUETDQNQKEMXAUTYXSKWRYGLECDSMOCWDAMQL
CRXPUMDLAMQLCLJPUMIDAIZTGEJLGKHENCDPVUXXGYDLAOSCGLFITK
WJWCIQQMOSQNOETIMEVZAPUCVSGUSKWKGVTYVCOSQNDQEMXTVUXFUO
QDDKHLCXSLVEASYVZUCLELVEYEWMAMPJXCMKNQXOUCGQUHFAKPALVE
OETKAVCCNCDXQUXEUJGDWWVRKGXOKCVZAUOYUJKCTQVZAWKRTCAXQL
GJPCIAJJGVLLQSFSXZACXHFCGQOEUZFCTYPRGGAVZCVUYVUCQSTRUE
VETDVKYUVDZLGNUXUIVCYVQUKLJHGKM
```

2. Break this ciphertext from the 2017 British National Cipher Challenge. You will notice something about the resulting key. Can you factor this cipher? Can you find a meaningful keyword (it won’t be in English).

```
YKUFRQHUDDQRGZPPISMXOOEYZUDOGMPPRLZQAERLOPNLOVKTVFNXOY
ZCQDZOVDVECXUAIBKVGLBEJRVPSLARHDTWOBNIMLWDFGGTAUQGONBL
RFLNMGMDRPYBVIGKXVXDBOQXUGBUVOKLQLPGHROVAMGIUKRDPHDOQE
```

OEIGIUKXIVDLYJLZIPEGELWVFDBMDBSGZKCGSYSOOOIOEOBOPTEMMK
HCROCVNLWDJOKSIGWCODTJGZCVLSYOYVTOURIWMHFDLZQKUAIIAWBO
OAMKHDNGEQBTOEICUGKFGYUCZIESQPUGAKANFMELCLZTMVINRRIRDB
OBVGHJBQYHGCKMSJILPIZQJLXEELPZJSNOYCQIMJMDHKIPIAWBOAOB
MBOLWESOGLYVOABXCENDBOQXUGBUVNJCDNVBQUUQGVAJTCLIOXGBJV
PUBAUTDSQYCTMNNTRHDAUFEKVPFBSNVYVWAXKZQJAAMMRIZOVJGMMU
MXELJTHYBAJVCPKLWHMRBVCMWIGKQRNVLYROVAONGUKWBOXGGDMVTG
JJOKEUMCTGDYDHPAYTVGNKAUBSFNDMYGNEEBODYLSIVMWEIUHOIYXI
IXRZILMINLCRNYBODTQGKACUKPJNVLYROVGSBIGCRKFYRNLDPIAEGE
EZSHVGMROREIAWGMEFFKAILLHWPUMTIPJLBEQBADDORIELFILZDJSN
CTQCTMEGRJAAVQQWEIAVTIBNFWAOFTGLYDDLEERFYBGUOVCMWTLLDQ
AWYZOQSQENAPVPYRHGBKXNTOUXTXWXOXGNISDDMOIRVXRICLWKDZEI
WINWCUNDYJSNQYWWIRVXSYBOTNVHFLAUTECBGYBODTKGBSTLLYIUDE
LGTIAKCVTIYKNGLLQMQIINGXKAOHURMNQSINWTLXXSGAAQBHSGIUQQ
CPLNATQBHNNJECMKCRIPGXIVHGMDJOKPQJWYBXLIUODKUQMSZSDYRO
XAIMMSIFLXYEISEHTOACTDJAOXGKSMVANPIRDRYRUOBROTHJCDRDJQ
NNTOULKGHHSSCGKFNYZROVGSBIGCRKZUKSOEOZGGKUMCBKUAPBLWIS
CKEIDPFERPSEJHJCGXPGRPUOIKRSOIDKBEBYZKDLYNTAPXIEKAEFRG
BBLIYVDDTJRICERAQOVCMWUOIRENVCQROPGAKVVVSRICNBOHXDOBIR
WEMPDBAOQVZGNKRIPJICEHQMWBOPIUCQIYJTGBUHELQGFSYSOYEIIE
SQPXIELZOODBONLPAGJJOKQIGCJKONIGDCGQKXKLZOIRVDAKPAUKWX
IEAILLHWMDJOKPQJWIZOQVHPCVUCQOOUZMRTOHJCPVRFLGAQTQANTO
DDTACNNNZIIAZEYETUFEANATGXUQGAUTNSBKIEFIIAZEBOQNIRDNOI
ZYNHDOGIEEGSLNZLYFXOIHGTPGYHXYZIGTHWWETIRQNFFBQNPDYODR
XSQGHQGYGZYIDMBVGXPOGZRRGZCPIRLEUMCTGDMNKQDKRVFIHCSWDT
UERPAXIPOPNHJQENTPXIGVUSYQZGUNTRHMIUUWYJPIPQUNKOUPUFEN
FWAGXUQGAUBTNWCZKDPIGDKYUQGHEOGCTMEGXSGZGOIYOKPGMJDUEQ
VJOKDDPOGGXNDSJAFLEYVVDIVALDZGNAUXECFRGBROVSGVIWPEXFPG
PLTUEIWKEYWXOXGYQRIHJCTDCMULMLOKBIGPXGUUWYJEDLZLXRJEMD
CMRNZDSQUQEPETUKBNPQGZCTCTGGGFIDRVPGHKSFAWQWTEFKOXEPRO
VHBAKUCRCLZUMCBPBMWJOKQIGCJMBMXASACNMPEIEAIYSUYBDIGEGB
OXGZIGWBKXKQKFLNHEBVRIIMNCUFEUHCDDQNTTOKTQMUQKMHGURLGS
KUOCLRICSGJTYSMZWRWMKPWEZQMDSMDIMXWULNEZJOVKWWJYKYRFBF
OJYVNGPIOKQOYVGIRGNQPXINKQYAEHYIAWBOEMSKRSLZSQPOVTIDDT
WWJNFCQNQAGBCULZEDNVYBALDQGKVAIISKMEZJTKGBMAEKOIVSRGMP
XDLBGDLZQMPIAQRGZIUJLLZQHCVOBUQFMPJLRFLNDTWQNWYBMAAISD
POYLOVGCJEUOLZQUAHOIAWBONKELZNNQIGBRGZGJJOKQIGCJQRNVDV
YKQHGBOEKOVTIGDCGQKXYYDILROVRQDEVHBVNQOMDMVZTMJOKCBZEG
ZRYTLYZAEMCBBUZHQNPYVYGG

Unit 45
Quagmire 1 cipher

The *quagmire 1 cipher* (also called polyalphabetic *type 1*) uses a mixed alphabet for the plaintext and shifted alphabets for the ciphertext. The mixed alphabet is generated from a keyword. The shifts of the ciphertext alphabets form another keyword, as they do in the Vigenère cipher.

Here is an example. The keywords are `QUAGMIRE` and `CIPHER`. First, look at the table of key alphabets. Notice that the shift keyword appears under the 'A' in the plaintext alphabet.

```
plaintext:   quagmirebcdfhjklnopstvwxyz
---------------------------------------
   C         ABCDEFGHIJKLMNOPQRSTUVWXYZ
   I         GHIJKLMNOPQRSTUVWXYZABCDEF
   P         NOPQRSTUVWXYZABCDEFGHIJKLM
   H         FGHIJKLMNOPQRSTUVWXYZABCDE
   E         CDEFGHIJKLMNOPQRSTUVWXYZAB
   R         PQRSTUVWXYZABCDEFGHIJKLMNO
```

Now we encipher a short message with this key table. The center row indexes the key alphabet.

```
THIS MESSAGE IS ENCRYPTED WITH A QUAGMIRE CIPHER
CIPH ERCIPHE RC IPHERCIPH ERCI P HERCIPHE RCIPHE
USSY GWTZPIJ UT NDOINSAUP YUUS P FDRDKSLJ YFYZMI
```

There are some special cases of the quagmire 1 cipher:

- period = 1: monoalphabetic substitution in which the keyword is used to generate the plaintext alphabetic
- mixed alphabet = regular alphabet: Vigenère cipher
- mixed alphabet generated as in affine cipher: periodic affine cipher

Because the plaintext alphabet is the only one that is mixed, the quagmire 1 cipher can be factored into a monoalphabetic substitution cipher followed by a Vigenère cipher. However, the key for that monoalphabetic substitution is the inverse of the quagmire's plaintext alphabet, and the Vigenère key must be shifted by a Caesar shift until its first letter is ‘A.’. For the example above, the substitution key is

`CIJKHLDMFNOPEQRSAGTUBVWXYZ`

and the Vigenère key is `CIPHER` shifted by a Caesar shift two steps back to `AGNFCP`.

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; chapter XVIII.

American Cryptogram Association, www.cryptogram.org/downloads/aca.info/ciphers/QuagmireI.pdf

Programming tasks

1. Write a function that takes the keywords for a quagmire 1 cipher and outputs the key alphabets for the periodic polyalphabetic substitution cipher.
2. Write a function or script to encipher a text with the quagmire 1 cipher and given keywords. You may use the function from Exercise 1, but there are other ways to accomplish this.
3. Write a function or script to decipher a text with the quagmire 1 cipher and given keywords. You may use the function from Exercise 1, but there are other ways to accomplish this.
4. Write a function or script to perform a dictionary attack on a ciphertext encrypted with a quagmire 1 cipher.
5. Write a function that takes the two quagmire 1 keywords and outputs the keys for the monoalphabetic substitution and Vigenère ciphers into which the quagmire can be factored.

Exercises

1. Verify that the example above does indeed factor into a monoalphabetic substitution and a Vigenère, and that the keys are the ones given.
2. Encipher this text with the keywords `ULTIMATE` (alphabet) and `QUESTION` (shifts).

```
O PEOPLE WAITING IN THE SHADOW OF DEEP THOUGHT!
HONOURED DESCENDANTS OF VROOMFONDEL AND MAJIKTHISE,
THE GREATEST AND MOST TRULY INTERESTING PUNDITS THE
UNIVERSE HAS EVER KNOWN... THE TIME OF WAITING IS
OVER! SEVEN AND A HALF MILLION YEARS OUR RACE HAS
WAITED FOR THIS GREAT ANSWER! NEVER AGAIN, NEVER AGAIN
```

```
WILL WE WAKE UP IN THE MORNING AND THINK WHO AM I?
WHAT IS MY PURPOSE IN LIFE? DOES IT REALLY, COSMICALLY
SPEAKING, MATTER IF I DON'T GET UP AND GO TO WORK? FOR
TODAY WE WILL FINALLY LEARN ONCE AND FOR ALL THE PLAIN
AND SIMPLE ANSWER TO ALL THESE NAGGING LITTLE PROBLEMS
OF LIFE, THE UNIVERSE, AND EVERYTHING!
```

3. Decipher this ciphertext with the keywords `WAR` (alphabet) and `PEACE` (shifts).

```
IMFUUBNTVNDLUSTVXIHDEFMGNDXPVMQJFJFUETUYFJBUXPXFUDPWAQ
JKJOVDXNDKHEYMGGUEOGNDIFHIOESITQJSHJDGFITQJTKJCNEGQUTG
VMUXZHSINFVMSJOVYQBZLXXXIHERWPDUINPQTVJUDTFJCZFJWSLEPS
ERKIMFEFYXJUMURQLFUGXVMUYOLXUISVEIJSVDETGVMUXIDJUJWLWI
NOJUEDFDWUYBCWYEAQIEHFCSYEZHFUEMDJPIXHKVJDVNKJMZNDGFLS
WXIHXXNBGJPWTCWYEPQQMJNHFWJECWPIJUXYSDVYDNTCKIJBCSEXIH
FTJDCIUTGFTDKUUJTKJJMINOJXXJGDTDXJHFHGFVDUJOVMUXIDJUWU
SJQWTCXUWADJYSSRRUUMCHUWADGYXBCFMEOGNDTTKJQWTKJMKMWHIY
AVJPHDRFTNOJXEXIHKEFTWSUWPIDPFCWXYSHHSUFAOXXJXITBQPBLU
THDEFMJFEBQJQJHJUDEHNAFTCUBLWUWTKJOMPOJEKTKJDTBVMUFOSE
QXPIXXJFWFEUFCSPSECWYETLHBEOGRPWSIFERQRFIYHCQITTKJRJBL
SWWTDEYXPFJPSJCHERQDNHJSVMUENHFYHAUXXJAVQPSTLHYWMCSTWJ
QHBYELSWXIHGQNTLWXNSOJHEUUXQEMCWYEAQIIMFUTJXIHFDUPDXYT
ORKPKBLHPJAUXPWJCWCEMOJQXICSIMFRXXJBUEDIZLXXEMHWHIFIND
NTHDUWTHFDKBRSINFDHERQDNHJSFMYSACSTXIHHEYOVFYJSVTIMFUT
JXIRKYX
```

4. Break this ciphertext with a dictionary attack. Both keywords are common five-letter English words.

```
VXUHEXTOLHJMJVCVCHQCQENAZOZWKOMQNHEJRGQUAOJVCYZFAGVFXA
UAMDQCKZMQKUTYGVEXDENTGGQKFCGCNEJIVIMXDBNVROMESMYCBTQI
TTTULNFLHTQMKZNCSYOJFUAMJVCVCGMVXTFAULMWSJVPYHCUOSMEKO
JFHTXYCTUOSMEKOWATLFUIHLTGXEEXBAEJNWACWOJFHTXYCGAURFST
PBAEJWNCMBZNDPBRTDUOOJSUGARCZOUYHHTXYCGAURFSTPIDHTAXIN
GPXCSTDYINEHTQTTENAZJMYNPKEZOKYETZCVLRAUGCGCHGCBQKLMFA
TBRGRUAMDVCVCFLEXWBHKXTORTMQNCBTOXSNQQTLPMCOMTGNRCHBYI
ZCLO
```

Unit 46
Two-stage attack on the quagmire 1 cipher

Because the quagmire 1 can be factored into a monoalphabetic substitution followed by a Vigenère cipher, the shifts are exposed to cryptanalysis. In this attack, we find the shifts by comparing the monogram frequencies of slices of the ciphertext. What remains then is a monoalphabetic substitution, which we can break with the technique in Unit 28.

Here’s how the attack works:

1. find the period *m*
2. partition the ciphertext into *m* slices; each slice contains every m^{th} letter, but each slice starts from a different one of the first *m* letter of the ciphertext
3. find the monogram frequencies for each slice
4. fix the frequency table for the first slice; for the others, shift them left (with wrap-around) until they match as closely as possible the table of the first slice
5. the shifts needed to align the frequency tables forms a Vigenère key; decipher the ciphertext with this key
6. break the resulting text as a monoalphabetic substitution

To recover the original keywords of the quagmire cipher, look at the key that you find for the monoalphabetic substitution. Take the value of its first letter, where we start at ‘A’ = 0, ‘B’ = 1, etc. Add that value to each of the shifts that you found in step 4. Convert the resulting shifts to letters to get the shift key of the quagmire. To find the alphabet keyword, invert the key of the monoalphabetic substitution. You may not always get a recognizable keyword, due to problems with infrequent letters.

Programming tasks

1. Implement the attack. Use the cosine of the angle between vectors to find the best match for the frequency tables. Feel free to also use your function that performs the hill-climbing attack on the monoalphabetic substitution cipher.

Exercises

1. Break this ciphertext and find the original quagmire keywords.

```
TNBOWMQSCQFMFOBZVKYAVNUBJVKSLLKESDBGLJPVFNEFLHUGNKVRDO
QCVKNRKJFXWKIUDNVWBTBHPSESLTBJIVUHUIIAGGKMQCMINEOUAMYO
NMATPOJVSSLXLNOFJAVFDSDEXBGTUUENVNHDXILUDYXGFOJRNCHRJO
TVOZLBMCKJBUIUJRPNRGOPBTTCXIETUPBCJBWEWOVUAMVDAGEIKFZL
LPSJVHPSMLQSSSLNHKMKZSMLYACCKGPUZAEWKBBLLSXIXTPGMTCSCC
XEFQEXIPPGKJFSBCLKPFSWBTDXYMRIVPPSACLVPFSVFSGLICHFPHPS
ACLXLKGBFOBRVVEGDEBTACLKPRKGPXVMVPZSCAGPCMQSCEEGUPEHUU
PVNJBUETZWZGEIUPRTWDPREEYUPVYGEWAEUSKXNHZFFOMHWPUOYKDA
MFWNAKUEKSKFETZRZQTNJUWYBHNXNATUWHNOCPKOTFMTUMENTNJURX
UGUISOPMMZHHENVSDHMLVVENVWFCJTUUYSKCEFOMQOEXAEZXMKLVNF
SEHXKMQRFTWOBSOLHBRJFOMTRXIJPWAESFIGNHUIIETKMLHKXEGSBO
OIJREYIETKCGPIASEPFJALQSESFLHPEGHYLVYRPNWGLCSXAETKMUCS
DEJSKFXTZGPKTTXBAUHTPFVGPSIGPSXEIMBCIZLTFXKOKFZBYSLXXC
TBXIVQYXDEOUKMDODIDPUZACLUNKEOUCKELHZKIOQURXROCLFRIFIK
VVWGSBDOQLVAPJFGZVVZLFYIRTKTWFHBLMVGUPXNAQEGETPPVXVVEN
VDVQNHHFRWRSTKMALPMTRSUDB
```

Unit 47
Quagmire 2 cipher

The *quagmire 2 cipher* (also called polyalphabetic *type 2*) uses a mixed and shifted alphabet for the ciphertext. The mixing is generated from a keyword. The shifts form another keyword, as they do in the Vigenère cipher.

Here is an example. The keywords are `QUAGMIRE` and `CIPHER`. First, look at the table of key alphabets. Notice that the shift keyword appears under the 'A' in the plaintext alphabet.

plaintext:	abcdefghijklmnopqrstuvwxyz
C	CDFHJKLNOPSTVWXYZQUAGMIREB
I	IREBCDFHJKLNOPSTVWXYZQUAGM
P	PSTVWXYZQUAGMIREBCDFHJKLNO
H	HJKLNOPSTVWXYZQUAGMIREBCDF
E	EBCDFHJKLNOPSTVWXYZQUAGMIR
R	REBCDFHJKLNOPSTVWXYZQUAGMI

Now we encipher a short message with this key table. The center row indexes the key alphabet.

```
THIS MESSAGE IS ENCRYPTED WITH A QUAGMIRE CIPHER
CIPH ERCIPHE RC IPHERCIPH ERCI P HERCIPHE RCIPHE
AHQM SDUXPPF KU CIKYMYYWL GKAH P AURLOQGF BOTZNY
```

There are some special cases of the quagmire 2 cipher:

- period = 1: monoalphabetic substitution (keyword cipher)
- mixed alphabet = regular alphabet: Vigenère cipher
- mixed alphabet generated as in affine cipher: periodic affine cipher

The quagmire 2 cipher can be factored into a Vigenère cipher followed by a monoalphabetic substitution cipher. The key for the monoalphabetic substitution is the same as the mixed alphabet generated by the alphabet keyword of the quagmire, but the Vigenère key is the shift keyword of the quagmire after it is encrypted by the inverse of the monoalphabetic substitution. Because the second factor of the quagmire 2 is a substitution cipher, it is resistant to the attack in the previous unit.

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; chapter XVIII.

American Cryptogram Association, www.cryptogram.org/downloads/aca.info/ciphers/QuagmireII.pdf

Fletcher Pratt, *Secret and Urgent*, New York: Bobbs-Merrill, 1939, 147.83.93.163/cops/fetch.php?data=4538&type=pdf&id=2942, chapter XI, section II.

Programming tasks

1. Write a function that takes the keywords for a quagmire 2 cipher and outputs the key alphabets for the periodic polyalphabetic substitution cipher.

2. Write a function or script to encipher a text with the quagmire 2 cipher and given keywords. You may use the function from Exercise 1, but there are other ways to accomplish this.

3. Write a function or script to decipher a text with the quagmire 2 cipher and given keywords. You may use the function from Exercise 1, but there are other ways to accomplish this.

4. Write a function or script to perform a dictionary attack on a ciphertext encrypted with a quagmire 2 cipher.

5. Write a function that takes the two quagmire 2 keywords and outputs the keys for the Vigenère and monoalphabetic substitution ciphers into which the quagmire can be factored.

Exercises

1. Factor the quagmire 2 cipher used in the example above, and find the keys of the two factor ciphers.

2. Encipher this text with the keywords `CHARLES` (alphabet) and `DICKENS` (shifts).

   ```
   IT WAS THE BEST OF TIMES IT WAS THE WORST OF TIMES IT
   WAS THE AGE OF WISDOM IT WAS THE AGE OF FOOLISHNESS IT
   WAS THE EPOCH OF BELIEF IT WAS THE EPOCH OF
   INCREDULITY IT WAS THE SEASON OF LIGHT IT WAS THE
   SEASON OF DARKNESS IT WAS THE SPRING OF HOPE IT WAS
   THE WINTER OF DESPAIR
   ```

3. Decipher this ciphertext with the keywords `PLANET` (alphabet) and `EARTH` (shifts).

```
FAFSTXGTYRDWTCRESHFKTAUMCEVWWNRCHIMYMXTNHGBRHTJWFYCOXY
RDYWXEDSTXPISROHVKBGEHBYTWEZOOSDURXXEJNZYVBYTACBVNMMJB
LNYMBWIIVPRQXFPXHXAVJNXATCMRCFSTGFNNYIMWYFXYBQSMJPSYOG
NFNIURRTXVWWWAGTXSGMPGSBATYWIVHOMTJIFQVBWRPMATFEPFBXME
QWDMWEWRKDTNJODCBWXWAFFNRAETGIMYOFSSPQSXGJFEHAHYRDPGYS
MJHISQIVJQIVROCEVUIMWAFFHSSWYEAMWTEITWT
```

4. Break this ciphertext with a dictionary attack. Both keywords are common five-letter English words.

```
CTXMSTARXSYRSWGOUKWBRPEHCNHCRNZDXJRDAAPSCUWGQTAPUJOLGX
DGPEJACUWJDZRIZPYOKHBCNSWBGDEPBCZTNCHDOMUKFRUOYODESYOI
EWZIGGBZHIICVIBGSJMTSSSLOHQGDTMSDAVFRZMMNNJNTNBTZPMIGD
TMXCQHGBNAZGSWTOOBHDEJZCNSMSWOBOBWAOBBMOGZSULMJTNHCKAV
BXOTVSRUECSDESQATKBTABPBGHQWZSZOOVVMTJAGOCKSCKRJAKPBJE
KSGUBGRMNGKDTMSCUWJGKVIBSWUUJAHRGCLVVDENGKTGEONVRIYRVG
RGLTDNUOUDLDHBOVGDOOQOBUGAZNEOTYOROVKTDWZSHIIILRIJKNHC
EIYTTNNYAPMSJITOCKSUJADACFQKTDENGUWGLCSTRNOSKFCNUWGWYB
TGQONVOVKZOMSGRHIVGUWGWYMOHNMAOHQGODENGAIZAOFTMBJAOMZA
DXMSJIHOWNHCXAVUWGAMRXHRVUWFRMVIZSYDXDORUVMIEUWRIYUWGJ
NHQEVOTOHWTRGNIUDXNQRNVEWSLCWAKRCDDHDPJDUFDJVOMONWDACF
QKTDEWZNEOVONVOVCUWGBVPGGRGHUPZSZVFHKTRGCNACJGKUWFRMUW
WCURDPAKCOQSTAWFBHOHHCKNOHPKTTGGZUEONQAUMITMTESSADNGKB
TGQEHDEVOTWFRJTPWAZMTOVKPGWPKOUWOYODESYRIZNZMHYVCTXHVO
MGFSZMXHSZMMISUUXCWGZKFCNMMGQREOOWVNDEWZHCFCCFODCNRDIA
OAJGBTAOHQGDTNEOSXOILMMRNGDTMJNHQEWGDXHYOTKFCNHIDSDAVF
RZUEDDGHIUNMRXHBGUWFBZABQWGVSGWEHGCRVLEHUKSTHQRSTFCGDE
PUNZTOWQNESRVXXNSYARGQADESCVRJJWJHD
```

Unit 48
Quagmire 3 cipher

The *quagmire 3 cipher* (also called polyalphabetic *type 3*) uses a mixed alphabet for both the plaintext and the ciphertext. The mixing is generated from a keyword. The ciphertext alphabets are shifted versions of the mixed alphabet, and the shifts form another keyword, as they do in the Vigenère cipher.

Here is an example. The keywords are `QUAGMIRE` and `CIPHER`. First, look at the table of key alphabets. Notice that the shift keyword appears under the first letter of the alphabet key.

plaintext:	`quagmirebcdfhjklnopstvwxyz`
`C`	`CDFHJKLNOPSTVWXYZQUAGMIREB`
`I`	`IREBCDFHJKLNOPSTVWXYZQUAGM`
`P`	`PSTVWXYZQUAGMIREBCDFHJKLNO`
`H`	`HJKLNOPSTVWXYZQUAGMIREBCDF`
`E`	`EBCDFHJKLNOPSTVWXYZQUAGMIR`
`R`	`REBCDFHJKLNOPSTVWXYZQUAGMI`

Now we encipher a short message with this key table. The center row indexes the key alphabet.

```
THIS MESSAGE IS ENCRYPTED WITH A QUAGMIRE CIPHER
CIPH ERCIPHE RC IPHERCIPH ERCI P HERCIPHE RCIPHE
GOXI FJAYTLK FA HBVJMUZZW GFGO T HBBHCXPK LKXMSJ
```

There are some special cases of the quagmire 3 cipher:

- period = 1: monoalphabetic substitution
- period = 1 and shift keyword = first letter of alphabet keyword: no encryption
- mixed alphabet = regular alphabet: Vigenère cipher
- mixed alphabet generated as in affine cipher: Vigenère cipher (whose key is a Caesar shift of the quagmire’s shift keyword)

The quagmire 3 cipher can be factored into a Vigenère cipher sandwiched between two monoalphabetic substitution ciphers. The keys for the monoalphabetic substitutions are inverses of each other. The final substitution cipher uses the same keyword as the quagmire’s alphabet keyword, while the first substitution is its inverse. The key of the Vigenère is the shift keyword of the quagmire encrypted by first of the substitution ciphers.

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; chapter XVIII.

American Cryptogram Association, www.cryptogram.org/downloads/aca.info/ciphers/QuagmireIII.pdf

Programming tasks

You may use the function from Exercise 1, but there are other ways to accomplish this.

1. Write a function that takes the keywords for a quagmire 3 cipher and generates the key alphabets for the periodic polyalphabetic substitution cipher.
2. Write a function or script to encipher a text with the quagmire 3 cipher and given keywords. You may use the function from Exercise 1, but there are other ways to accomplish this.
3. Write a function or script to decipher a text with the quagmire 3 cipher and given keywords. You may use the function from Exercise 1, but there are other ways to accomplish this.
4. Write a function or script to perform a dictionary attack on a ciphertext encrypted with a quagmire 3 cipher.
5. Write a function that takes the two quagmire 3 keywords and outputs the keys of the substitution ciphers and Vigenère cipher into which the quagmire can be factored.

Exercises

1. Factor the quagmire 3 cipher in the example above, and find the keys of the factor ciphers.
2. Encipher this text with the keywords `NURSERY` (alphabet) and `RHYME` (shifts).

   ```
   THE CAT AND HER KITTENS THEY PUT ON THEIR MITTENS, TO
   EAT A CHRISTMAS PIE. THE POOR LITTLE KITTENS THEY LOST
   THEIR MITTENS, AND THEN THEY BEGAN TO CRY.
   ```

3. Decipher this ciphertext with the keywords `DOLPHINS` (alphabet) and `FISHBOWL` (shifts).

   ```
   UXDAPTKWHERHJUJPNESLUUHKUCGFKOKNQFDBZFKNXSFQGUAUGVDAUO
   MBIPBBLUYGVMSEJOSHUVCJQVBNICDFLBWSYQRWUSKCRRBKQSWKJCDX
   UTSLGEJIZFASRMKGOBPSGEJIKUDGYCJARLRHTFOWVBNGMMEKEOOVNA
   KHXVPVQEUXGFHAZSRXCUDGJCUALBLLTZSHUSYDTGPJEFHUJPDAUGOA
   ```

```
XSIKKJJCSMSEGBZGTNTDCTWBTBKHCXSLGEDAUJYCTFDAUZPCQIJIEL
LKUESXGBLQT
```

4. Break this ciphertext with a dictionary attack. Both keywords are common five-letter English words.

```
OHSRHJIURBYNPSWGTIMBOHFCMYNHECDEBJVUBPAAUNEJIOHFOCJLIM
TVEFKJORPKWYROCJGNAACELEFXFECRHRLPKWYR
```

Unit 49
Quagmire 4 cipher

The *quagmire 4 cipher* (also called polyalphabetic *type 4*) uses mixed alphabets for both the plaintext and the ciphertext. The mixing is generated from two keywords. The ciphertext alphabets are shifted versions of the mixed alphabet, and the shifts form another keyword, as they do in the Vigenère cipher.

Here is an example. The keywords are `QUAGMIRE`, `KEYWORD`, and `CIPHER`. First, look at the table of key alphabets. Notice that the shift keyword appears under the first letter of the alphabet key.

plaintext:	`quagmirebcdfhjklnopstvwxyz`
C	`CFGHIJLMNPQSTUVXZKEYWORDAB`
I	`IJLMNPQSTUVXZKEYWORDABCFGH`
P	`PQSTUVXZKEYWORDABCFGHIJLMN`
H	`HIJLMNPQSTUVXZKEYWORDABCFG`
E	`EYWORDABCFGHIJLMNPQSTUVXZK`
R	`RDABCFGHIJLMNPQSTUVXZKEYWO`

Now we encipher a short message with this key table. The center row indexes the key alphabet.

```
THIS MESSAGE IS ENCRYPTED WITH A QUAGMIRE CIPHER
CIPH ERCIPHE RC IPHERCIPH ERCI P HERCIPHE RCIPHE
WZVR RHYDSLB FY SBTAWEAZU VFWZ S HYAHNVPB JJROQA
```

There are some special cases of the quagmire 4 cipher:

- period = 1: monoalphabetic substitution
- unmixed ciphertext alphabet: quagmire 1
- unmixed plaintext alphabet: quagmire 2
- same keyword for plaintext and ciphertext alphabets: quagmire 3
- both alphabets unmixed: Vigenère cipher

The quagmire 4 cipher can be factored into a Vigenère cipher sandwiched between two monoalphabetic substitution ciphers. The final substitution cipher uses the same keyword as the quagmire's ciphertext alphabet keyword, while the first substitution is the inverse of the key generated from the quagmire's plaintext alphabet keyword. The key of the Vigenère is the shift keyword of the quagmire encrypted by the inverse of the second substitution cipher.

Reading and references

Helen Fouché Gaines, *Cryptanalysis: a study of ciphers and their solution*, New York: Dover, 1956; previously titled *Elementary Cryptanalysis* and published by American Photographic in 1939; archive.org/details/cryptanalysis00gain; chapter XVIII.

American Cryptogram Association, www.cryptogram.org/downloads/aca.info/ciphers/QuagmireIV.pdf

Programming tasks

1. Write a function that takes the keywords for a quagmire 4 cipher and generates the key alphabets for the periodic polyalphabetic substitution cipher.
2. Write a function or script to encipher a text with the quagmire 4 cipher and given keywords. You may use the function from Exercise 1, but there are other ways to accomplish this.
3. Write a function or script to decipher a text with the quagmire 4 cipher and given keywords. You may use the function from Exercise 1, but there are other ways to accomplish this.
4. Write a function or script to perform a dictionary attack on a ciphertext encrypted with a quagmire 4 cipher.
5. Write a function that takes the three quagmire 4 keywords and outputs the keys of the substitution and Vigenère ciphers into which the quagmire can be factored.

Exercises

1. Factor the quagmire 4 cipher in the example above, and find the keys of the factor ciphers.
2. Encipher this text with keywords `FOUR` (plaintext alphabet), `LETTER` (ciphertext alphabet), and `WORD` (shifts).

   ```
   SEEMS LIKE ONLY YESTERDAY I LEFT MY MIND BEHIND DOWN
   IN THE GYPSY CAFE WITH A FRIEND OF A FRIEND OF MINE
   WHO SAT WITH A BABY HEAVY ON HER KNEE YET SPOKE OF
   LIFE MOST FREE FROM SLAVERY WITH EYES THAT SHOWED NO
   TRACE OF MISERY A PHRASE IN CONNECTION FIRST WITH SHE
   OCCURRED THAT LOVE IS JUST A FOUR LETTER WORD
   ```

3. Decipher this text with keywords `FOUR` (plaintext alphabet), `PIGMENT` (ciphertext alphabet), and `COLOR` (shifts).

   ```
   MYCSWLPRQAJWJIFMPXASGCXCFUQNQRCQQSMZXXAPZXXTHLQMQVHYVY
   CMQRCMZDZYPAYPTHQDRUYSSONWMIVQVRUNIFYCCPPLQMQVGASWPUCL
   ```

```
QUHXPNPZCPQBJEPTGFNSTFMPXZPZYLQVMPXSPOAPZAQTCIFDTCPPTW
ZSSYQBQAQCONPNQEYBSUDYHNQWIRCYRYBMFXSFUQNCSCDZUHRYNYWZ
XCPPHYLQVQPSFPCQQSVJDZYVMPSTHSSXYHAAXCSCWOESHSXWPPDOSF
DTXCULPFQAJTVQAPESIBQXCPWMCZYYDRMXBQFXSRUTWWTDDZYFMPSD
AJDVP
```

4. Break this ciphertext with a dictionary attack. All keywords are common five-letter English words.

```
XZPRVLJINCQXKYWVKXJJQXPRVCARJTCQSJDQCRECDGGIXTKKYWDUWJ
XBKTZCAKJAHRKMZFNMNZLBLRXVAIRZGQXGIXTKPTFNCQJUCIPCDQCI
ZSLJINCQXSJDQCRECDGRHKWMPAANXKBTCQHRQLUMHCAOVCWXZOAANI
HJVOSNWCVTPLJNCQJTQYIOHBMVWHRIIPMNAGLGOUXLGJUXJJDUKWJN
QIPVAOIOJQYIAANJYHKFXKAVCQOJGNAKWJNXKACGUPFCLCQLKQOMJD
NAKACCZRHCALRHKXZOCCLTKMDNQSZFVAKECDG
```

Unit 50
Hill-climbing attack on periodic polyalphabetic substitution ciphers

The main idea of this attack is an extension of the hill-climbing attack on the monoalphabetic substitution cipher. We find the period *m*, and so must work with *m* key alphabets. In the earlier attack, we swapped letters in the key alphabet and kept the new key only if the fitness of the deciphered text improved. Here, however, the key space has more dimensions, and it is easy to get trapped in a local maximum. To avoid this, we will take turns randomizing one of the key alphabets and climbing back up the hill.

Here is the algorithm:

1. find the period *m*
2. set big counter equal to 0
3. set best fitness equal to the fitness of the undecrypted ciphertext
4. set the parent key equal to a set of *m* key alphabets
   (best to choose key alphabets that maximize the monogram fitness of the plaintext)
5. while big counter is less than some large number
   a. for each *i* in 0, ..., m−1
      i. randomize the i^{th} key alphabet
      ii. find plaintext by deciphering the ciphertext with the parent key
      iii. set the parent's fitness as the fitness of the plaintext
      iv. set little counter to 0
      v. while little counter is less than about 1,000
         - copy the parent key into a child key
         - swap two randomly selected characters in the child's i^{th} key alphabet
         - find plaintext by deciphering the ciphertext with the child key
         - set the child's fitness as the fitness of the plaintext
         - if the child's fitness is greater than the parent's fitness
            · copy the child key into the parent key
            · copy the child's fitness into the parent's fitness
            · set little counter to 0
         - increment little counter

        - if the child’s fitness is greater than the best fitness
            · set the best fitness equal to the child’s fitness
            · copy the child key into the best key
            · set big counter to 0
        - increment big counter
6. output the best key

The limit on the big counter that we like is about 1,000,000 times the period squared.

Reading and references

Thomas Kaeding, “Slippery hill-climbing technique for ciphertext-only cryptanalysis of periodic polyalphabetic substitution ciphers,” *Cryptologia* 44:3 (2020) 205-222, DOI: 10.1080/01611194.2019.1655504

Programming tasks

1. Implement the attack. You will find that when written in Python, the attack takes a long time. You may want to write another version in a lower-level language.

Exercises

1. Break this ciphertext. The keys have a hidden message for you.

```
PWNYPCGUUMYYUIAGGUGFEMWLRNYAHOGGHBAEWCXWPAQSXPAGNITNIX
WBQFKMLVIFHDTLSMADKMWWXXNYCQYBVWIKHWYARVTKXXAVGBGLVITN
WVNMKMUWRYQWRNEHYBCHUTNYQKYHIXWFWTQSLVYKRKZUHXZKVMQPHC
YALCRWHDKGJGTNOKSHXPAPHAGWQVGUHCXYUIAMXPKLXQRWKWNBXVTM
EMGGIXWYOMADHKGFWQQYQKYAHMHXRBYLRNYAHWRYXMZKFBEIXXIKWY
JWUCCARGTKNMNMLBGEHCXECDTVHLKIKMWQKWYSREOYOTYAHWWMLNAU
XVGPHBYAHTGLVTGMYCYYNMYALCTISXWMYVKMCVTQXXMKLVIHYBYQRY
GHSTGLXXIWXPGKWTQHAUGIOMALHDTXLBXMWYTEROKTHDJWWBYBIWOM
AWXIOWOWGXSMKMQNQWWLYBHYFMHYUDTHVXTGWCCWKWNYQDKULYAMHK
YAWDEHYGTNOKAUKQGPHCUYFMYKWFGEPEOAHWWELMWMKQXLKXZEGMDI
OWKGXPGYSYAKHVYYJMTXXPGYUDKXWLYYQKYAHLWNGMSWVCTXXPGEWV
INWOGQUQYMHVZIRVKMLDJYVCZXIMWWGDJWLUUYFDTXFXZGXTGLVUKU
UXRWXMTKVDJYXQRIWLYMKMQNQWWLYBHYFMOHQDKGYWQECNZKXPGKPX
WWXPGFHCXYJMCKLDYWQXSBXGALFXRIRCGVEIRSSBGVHLGLVXWLAPTQ
HBGEWBIWOIKZQXWYQDTXCXZKOWSZYWIWLUZLXWQLRWUHOXIBDMHHUD
JWGMQYCQSHYBKGXBTVYLYBRVXBQLGBXCUEWLGFHVYQKQQWWDYAWDYB
PMCWKWNIOWSGHKYHLVKMLWYWFXSMWLYBPUGVLWYWOIZIRVEHYBHBUC
YPLCKMXXEHYBUEWVGMVCAMHTQBXMKMKWNLLVOWXPAMXQRWEMGGGMOB
GMNMKWYQHCJHYTNGRDWWZMAEREWLHTBWVESMLTEHYBXHFQGMCPAVUM
AUKMNSHDAZUMAMHBQWZMQHIDGUKVTEROKUWTXHSPKLXQOYXQTGKXCW
ZMWSREWVLCOHZMWSRNYAHWWMLNAUXWSVGMOBSPGKPMSMRNKMKWXIUM
```

OBSQYYXMNMKQXBQDWHGEOMLXSBQDJWXQRWVQSUHDJWWBYBIWOMVLWW
WDKHQWSVSTAUHUGGXXSMKMQNQWWLYBHYFMCWKWBWEMGGAXWDLVIMRP
GESITNWLJBHFGMKWYMHLJGRTTZLLAEVXUALCYBFWYBRVTXAPKUKQXI
HWPBQBGUHVYVHLAVHCEHYBXHFQGMCPALHHUWUQGGFMNYQMDIRVGGXQ
AEJBTQXPKGXPGYUMALRNMBRLJWPQXMUIRWGQOBQMGGJQSWHBKGJYJS
VQOLWVNURUUNXWYBRVNNHQSEWBIWSWWMXXYAHQSCHLYBRVTXLKGYVW
SVXMOAQQVNHCKGXXEHYBXULMSMLNKUOQYWUWYNUMMSREWFHUMWUCYA
HCGYUMJHAMBWUUGKHTEMKMMWJQSGLVIHIGJYXGGUWVNHIXWSREWLRL
KWXIYHAQYQHPAPHWQLRAGWQWQOHQYLOXCECDTAHTUSRECBXPYAHQRI
UXBWPMSMRNTMKMWXWLGMVXHSREWLRLKWXIKMLCSHWLOBGMSMXPAMWK
BYQLGLLVYWFPSHOXISWBGYFLTFSWSBHKMSWKBYQLGLLVXNUFGBOTAG
FMKMLCCBXPXNFPAVZWSUHCYAWDCWKXUWXXJWOYEHYDTJYMQEXPGUKW
TLWVNNQYWWGQOMWAKELDEMKWYIOWINHCEHYBWYFMXMWAKELDEYQKOH
QDWHOWWWHCXWQDKYONTKYCJWUQSZEXYAREWIHXUEHCKGXXAGHGAZHX
HURXUWUWYBRVAGGYWHVYGKLDEYVQYLWIXHQDJWWBYBIWOMWVNYVGGQ
LTQYOGASVCASXXEHYGGURUGBQYGYFMCWRNHWUUAGCOKXXCYHCXZBQB
GMYBSQHXSECWXDCXZKRAGVLMSUH