ZAKRES MATERIAŁU

Kryptoanaliza systemu Vigenere'a

I. Odgadywanie długości klucza
- **(A) Klasyczna metoda Kasiskiego**
- **(B) Statystyczna metoda Friedmana**

II. Ustalanie klucza przy znajomości jego długości

Kryptoanaliza systemu Vigenere'a

[Idea] (Kryptoanaliza szyfrów Vigenere'a)
- Okres *m* można określić stosując **metodę Kasiskiego** lub **statystyczną metodę Friedmana**.
- Znając długość klucza *m*, jego postać możemy odgadnąć stosując jednoczesną analizę *m* niezależnych szyfrów Cezara.

Powyższa metoda kryptoanalizy jest skuteczna dla polialfabetycznych szyfrów podstawieniowych.

I. Odgadywanie długości klucza

(A) Klasyczna metoda Kasiskiego

[Idea] (algorytmu odgadywania długości klucza - metoda Kasiskiego)
- W zaszyfrowanym tekście (kryptogramie) szukamy par identycznych fragmentów o długości co najmniej trzech znaków.
- Zapamiętujemy odległości między pierwszym znakiem pierwszego fragmentu i pierwszymi znakami kolejnych fragmentów. Oznaczamy te odległości odpowiednio przez d1, d2, ...
- Mając wartości d1, d2, ..., przypuszczamy, że m | NWD(d1, d2, ...) (m dzieli największy wspólny dzielnik tych liczb).

[Zadanie 01] Znajdź długość klucza zastosowanego w szyfrze Vigenere'a do utajnienia tekstu otwartego T (jest to tekst angielski). Skorzystaj z metody Kasiskiego. Do dyspozycji masz tylko szyfrogram.

```
chreevoahmaeratbiaxxwtnxbeeophbsbqmqeqerbwrvxuoakxaosxxweahbwgjmmqmnkg
rfvgxwtrzxwiaklxfpskautemndcmgtsxmxbtuiadngmgpsrelxnjelxvrvprtulhdnqwt
wdtygbphxtfaljhasvbfxngllchrzbwelekmsjiknbhwrjgnmgjsglxfeyphagnrbieqjt
amrvlcrremndglxrrimgnsnrwchrqhaeyevtaqebbipeewevkakoewadremxmtbhhchrtk
dnvrzchrclqohpwqaiiwxnrmgwoiifkee
```

Powtarzający się ciąg znaków:	„chr"
Pozycje pierwszego znaku:	1, 166, 236, 276, 286
Odległości od pierwszego wystąpienia:	165, 235, 275, 285
Powtarzający się ciąg znaków:	„xxw"
Pozycje pierwszego znaku:	19, 54
Odległości od pierwszego wystąpienia:	35

NWD(165, 235, 275, 285, 35) = 5
Istnieje duże prawdopodobieństwo, że klucz ma długość 5 (i tak akurat jest).

[Zadanie 02] Znajdź długość klucza zastosowanego w szyfrze Vigenere'a do utajnienia poniższego tekstu otwartego (jest to tekst polski). Skorzystaj z metody Kasiskiego. Do dyspozycji masz tylko szyfrogram. Tekst otwarty podano dla ilustracji.

```
wtekściezaszyfrowanymszukamyparidentycznychfragmentówodługościconajmni
ejtrzechznakówzapamiętujemyodległościmiędzypierwszymiznakamifragmentów
należącychdojednejparymającakieodległościdjedenddwazakładamyżemdzieli
największywspólnydzielniktychliczb

gtvdsmivsaczpyrywrgywsqnkkmpiabiuxndytsnicyyrkgdxndonhdvuxhsmithnkjdgi
ojkkzocysnkkfpzkprfiotlcewyfwlogchsmidbenzpiiornlzimzsnkkrfiprrzmonkhw
xacxzkcpvhnoaxdxeaiabydtjkcktksefwlogchsmiuceneewdgaqtkvautmizvfdjivei
xaapiokjsygsghlxyusiolebkdytalscqu
```

Powtarzający się ciąg znaków:	„smi"
Pozycje pierwszego znaku:	5, 60, 105, 180
Odległości od pierwszego wystąpienia:	55, 100, 175

NWD(55, 100, 175) = 5
Istnieje duże prawdopodobieństwo, że klucz ma długość 5 (i tak akurat jest, ponieważ kluczem jest słowo „karta").

(B) Statystyczna metoda Friedmana

[DEF] (indeks zgodności (współczynnik koincydencji) (ozn. $I_c(x)$))

Oznaczmy przez $x = x_1, x_2, ..., x_n$ ciąg *n* znaków alfabetu.
Imdeks zgodności - prawdopodobieństwo zdarzenia, że dwie losowo wybrane litery ciągu *x* są identyczne.
Jeśli oznaczymy symbolami f_0 , f_1 , ..., f_{25} częstości wystąpienia znaków (alfabetu angielskiego) a, b, c, ..., z, to

$$I_c(x) = \frac{\sum_{i=0}^{25} f_i(f_i - 1)}{n(n - 1)}$$

[Uzasadnienie]
- dwa wyrazy z ciągu *x* można wybrać na $\binom{n}{2} = \frac{n!}{(n-2)!2!} = \frac{n(n-1)}{2}$ sposoby,
- dla każdego i $0 \le i \le 25$ mamy $\binom{f_i}{2}$ sposobów wybrania dwóch elementów *i*.

Niech *x* będzie ciągiem znaków w języku angielskim.
Symbolami p_0 , p_1 , ..., p_{25} oznaczmy oczekiwane prawdopodobieństwa wystąpienia liter a, b, ..., z (tabela 01).
Ponieważ prawdopodobieństwo, że dwa razy wybierzemy literę *a* wynosi $(p_0)^2$, dla *b* $(p_1)^2$, ..., dla *z* $(p_{25})^2$, więc

$$I_c(x) \approx \sum_{i=0}^{25} (p_i)^2 = 0{,}065$$

Stąd, w metodzie Friedmana szukamy takiego *m*, dla którego indeksy zgodności są w przybliżeniu równe 0,065.

A	B	C	D	E	F	G	H	I	J	K	L	M
0,082	0,015	0,028	0,043	0,127	0,022	0,020	0,061	0,070	0,002	0,008	0,040	0,024
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0,067	0,075	0,019	0,001	0,060	0,063	0,091	0,028	0,010	0,023	0,001	0,020	0,001

Tab. 01. Prawdopodobieństwa wystąpienia 26 liter alfabetu języka angielskiego

---

[Idea] (metody koincydencji Friedmana)

- Niech y będzie szyfrogramem uzyskanym z tekstu otwartego przez zastosowanie szyfru Vigenere'a.
- Utwórzmy m podciągów y_1, y_2, ..., y_m tego ciągu w ten sposób, że w skład y_1 wchodzą litery ciągu y, których pozycje modulo m są równe 0, w skład y_2 wchodzą litery ciągu y, których pozycje modulo m są równe 1, itd.
- Jeśli m jest długością klucza, to każda z liczb $I_c(y_i)$ $(1 \le i \le m)$ (dla tekstów angielskich) powinna być w przybliżeniu równa 0.065.
- Jeśli m nie jest długością klucza, to podciągi y_i będą miały charakter losowy, ponieważ powstały z zastosowania różnych kluczy.
- Dla całkowicie losowo wybranego ciągu

$$I_c(x) \approx 26 \cdot \left(\frac{1}{26}\right)^2 = 0.038$$

ponieważ każda litera z 26-znakowego alfabetu ma tę samą częstość występowania równą 1/26.

- Należy więc policzyć współczynniki koincydencji dla podciągów y_1, y_2, ..., y_m dla kolejnych wartości m i wybrać to m, dla którego współczynniki $I_c(y_i)$ $(1 \le i \le m)$ będą najbardziej zbliżone do wartości 0.065.

[Zadanie 03] Potwierdź (metodą Friedmana) hipotezę postawioną w zadaniu 01, że długość klucza użytego do zaszyfrowania tekstu otwartego jest równa 5.

```
thealmondtreewasintentativeblossomthedayswerelongeroftenendingwithmagn
ificenteveningsofcorrugatedpinkskiesthehuntingseasonwasoverwithhoundsa
ndgunsputawayforsixmonthsthevineyardswerebusyagainasthewellorganizedfa
rmerstreatedtheirvinesandthemorelackadaisicalneighborshurriedtodothepr
uningtheyshouldhavedoneinnovember

chreevoahmaeratbiaxxwtnxbeeophbsbqmqeqerbwrvxuoakxaosxxweahbwgjmmqmnkg
rfvgxwtrzxwiaklxfpskautemndcmgtsxmxbtuiadngmgpsrelxnjelxvrvprtulhdnqwt
wdtygbphxtfaljhasvbfxngllchrzbwelekmsjiknbhwrjgnmgjsglxfeyphagnrbieqjt
amrvlcrremndglxrrimgnsnrwchrqhaeyevtaqebbipeewevkakoewadremxmtbhhchrtk
dnvrzchrclqohpwqaiiwxnrmgwoiifkee
```

```
m=1
y1="chreevoahmaeratbiaxxwtnxbeeophbsbqmqeqerbw      // y1=y;
    rvxuoakxaosxxweahbwgjmmqmnkgrfvgxwtrzxwiak
    lxfpskautemndcmgtsxmxbtuiadngmgpsrelxnjelx
    vrvprtulhdnqwtwdtygbphxtfaljhasvbfxngllchr
    zbwelekmsjiknbhwrjgnmgjsglxfeyphagnrbieqjt
    amrvlcrremndglxrrimgnsnrwchrqhaeyevtaqebbi
    peewevkakoewadremxmtbhhchrtkdnvrzchrclqohp
    wqaiiwxnrmgwoiifkee"
```

tab. częstości wystąpienia 26 liter alfabetu języka angielskiego w tekście y1

a	b	c	d	e	f	g	h	i	j	k	l	m
19	15	8	7	26	6	15	17	11	7	10	12	17
n	o	p	q	r	s	t	u	v	w	x	y	z
15	7	8	10	24	9	14	4	10	16	20	3	3

```
Ic=0.045015155238797414
---
m=2
y1="creohartixwnbepbbmeebrxokasxehwjmmkrvxtzwa      // for(int i=0; i<y.length(); i++)
    lfsatmdmtxxtidggsexjlvvruhnwwtgpxflhsbxglh      //   if(i%2==0)y1+=y.charAt(i);
    zwlksinhrgmjgxepanbejarlrengxrmnnwhqayvaeb      // y1 jest typu String, więc znaki numerujemy od 0
    peekkearmmbhhtdvzhcqhwaixrgoike"
```

tab. częstości wystąpienia 26 liter alfabetu języka angielskiego w tekście y1

a	b	c	d	e	f	g	h	i	j	k	l	m
10	8	2	3	14	2	8	11	5	4	6	6	9
n	o	p	q	r	s	t	u	v	w	x	y	z
7	3	4	2	11	5	7	1	5	8	12	1	3

```
Ic=0.04556589906908378
y2="hevameabaxtxeohsqqqrwvuaxoxwabgmqngfgwrxik      // for(int i=0; i<y.length(); i++)
    xpkuencgsmbuanmprlnexrptldqtdybhtajavfnlcr      //   if(i%2==1)y2+=y.charAt(i);
    beemjkbwjngslfyhgriqtmvcrmdlrigsrcrheetqbi
    ewvaowdexthcrknrcrlopqiwnmwife"
```

tab. częstości wystąpienia 26 liter alfabetu języka angielskiego w tekście y2

a	b	c	d	e	f	g	h	i	j	k	l	m
9	7	6	4	12	4	7	6	6	3	4	6	8
n	o	p	q	r	s	t	u	v	w	x	y	z
8	4	4	8	13	4	7	3	5	8	8	2	0

```
Ic=0.041025641025641026
---
m=3
y1="chreevoahmaeratbiaxxwtnxbeeophbsbqmqeqerbw      // for(int i=0; i<y.length(); i++)
    rvxuoakxaosxxweahbwgjmmqmnkgrfvgxwtrzxwiak      //   if(i%3==0)y1+=y.charAt(i);
    lxfpskautemndcmgtsxmxbtuiadngmgpsrelxnjelx      // tylko czerwone znaki
    vrvprtulhdnqwtwdtygbphxtfaljhasvbfxngllchr      // y1 jest typu String, więc znaki numerujemy od 0
    zbwelekmsjiknbhwrjgnmgjsglxfeyphagnrbieqjt
    amrvlcrremndglxrrimgnsnrwchrqhaeyevtaqebbi
    peewevkakoewadremxmtbhhchrtkdnvrzchrclqohp
    wqaiiwxnrmgwoiifkee"
```

tab. częstości wystąpienia 26 liter alfabetu języka angielskiego w tekście y1

a	b	c	d	e	f	g	h	i	j	k	l	m
6	6	4	3	8	4	10	3	3	2	4	1	7
n	o	p	q	r	s	t	u	v	w	x	y	z
2	6	4	2	7	2	1	2	4	7	5	0	2

```
Ic=0.043223443223443223
y2=...                                              // if(i%3==1)y2+=y.charAt(i);
```

tab. częstości wystąpienia 26 liter alfabetu języka angielskiego w tekście y2

a	b	c	d	e	f	g	h	i	j	k	l	m
7	5	2	2	8	0	5	8	5	2	1	8	7
n	o	p	q	r	s	t	u	v	w	x	y	z
9	1	2	4	8	3	6	1	3	1	6	0	0

```
Ic=0.049663928304705
y3=...                                              // if(i%3==2)y3+=y.charAt(i);
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

tab. częstości wystąpienia 26 liter alfabetu języka angielskiego w tekście y3

a	b	c	d	e	f	g	h	i	j	k	l	m
6	4	2	2	10	2	0	6	3	3	5	3	3
n	o	p	q	r	s	t	u	v	w	x	y	z
4	0	2	4	9	4	7	1	3	8	9	3	1

```
Ic=0.04705003734129948
---
m=4
y1=...                                  // if(i%4==0)y1+=y.charAt(i);
Ic=0.04219409282700422
y2=...                                  // if(i%4==1)y2+=y.charAt(i);
Ic=0.03896103896103896
y3=...                                  // if(i%4==2)y3+=y.charAt(i);
Ic=0.04528804528804529
y4=...                                  // if(i%4==3)y4+=y.charAt(i);
Ic=0.040293040293040296
---
m=5
y1=...                                  // if(i%5==0)y1+=y.charAt(i);
```

tab. częstości wystąpienia 26 liter alfabetu języka angielskiego w tekście y1

a	b	c	d	e	f	g	h	i	j	k	l	m
7	6	6	4	1	2	0	0	1	2	2	0	2
n	o	p	q	r	s	t	u	v	w	x	y	z
4	0	1	4	3	0	2	1	1	9	5	0	0

```
Ic=0.0629800307219662
y2=...                                  // if(i%5==1)y2+=y.charAt(i);
```

tab. częstości wystąpienia 26 liter alfabetu języka angielskiego w tekście y2

a	b	c	d	e	f	g	h	i	j	k	l	m
3	1	0	3	10	2	3	5	3	0	0	0	2
n	o	p	q	r	s	t	u	v	w	x	y	z
6	6	1	0	3	7	5	1	1	1	0	0	0

```
Ic=0.06810035842293907
y3=...                                  // if(i%5==2)y3+=y.charAt(i);
Ic=0.06861239119303636
y4=...                                  // if(i%5==3)y4+=y.charAt(i);
Ic=0.060814383923849816
y5=...                                  // if(i%5==4)y5+=y.charAt(i);
Ic=0.07244843997884717
---
...
---
```

Sprawdźmy dla jakiego m (długość klucza) indeksy zgodności są najbardziej zbliżone do wartości 0.065.

Długość klucza	Średnia z indeksów zgodności	Długość klucza	Średnia z indeksów zgodności	Długość klucza	Średnia z indeksów zgodności	Długość klucza	Średnia z indeksów zgodności
1	0.0450151552387974	8	0.04203778677462889	15	**0.0693233082706767**	22	0.03886113886113885
2	0.0432957700473624	9	0.04553659847777495	16	0.0392543859649123	23	0.04809683070552635
3	0.0466458029564826	10	**0.06626344086021504**	17	0.0434043586474838	24	0.04036935286935287
4	0.0416840543422822	11	0.04331492262526744	18	0.0408950617283951	25	**0.07090909090909088**
5	**0.0665911208481277**	12	0.04217473884140551	19	0.0426470588235294		
6	0.0438283199104509	13	0.04241917502787069	20	**0.0604166666666667**		
7	0.0424544447800262	14	0.03823505686859725	21	0.0427350427350427		

Z tabeli wynika, że najbardziej prawdopodobne jest użycie klucza o długości 10, ale przyjmiemy, że klucz ma długość 5 (to jest również bardzo prawdopodobna możliwość), co ustaliliśmy wcześniej metodą Kasiskiego. Pomyśl jakie rachunki należy przeprowadzić, aby uzyskać bardziej precyzyjne wyniki niż przy obliczaniu średniej z indeksów zgodności (np. możemy wyznaczyć (dla każdego m) średnią z wartości |0.065 - Ic| i wybrać to m, dla którego średnia jest najbardziej bliska zeru).

[Zadanie 04] Potwierdź (metodą Friedmana) hipotezę postawioną w zadaniu 02, że długość klucza użytego do zaszyfrowania tekstu otwartego jest równa 5. Wyznacz teoretycznie wartość $I_C(x)$ dla alfabetu polskiego. Prawdopodobieństwa wystąpienia poszczególnych liter w tekstach polskich znajdują się w materiałach do ćwiczeń 05

II. Ustalanie klucza przy znajomości jego długości

[DEF] (wzajemny indeks zgodności)

Niech $x = x_1x_2...x_n$ i $y = y_1y_2...y_{n'}$ będą ciągami złożonymi z n i n' znaków. **Wzajemny indeks zgodności** (ozn. $MI_C(x,y)$) to prawdopodobieństwo tego, że losowo wybrany element ciągu x będzie równy losowo wybranemu elementowi z y.

Jeśli częstości występowania elementów a, b, c, ..., z w x i y oznaczymy przez f_0, f_1, ..., f_{25} i f_0', f_1', ..., f_{25}', to $MI_C(x,y)$ można wyrazić wzorem

$$MI_C(x,y) = \frac{\sum_{i=0}^{25} f_i f_i'}{nn'}$$

[Algorytm] (wyznaczania względnego przesunięcia)

- Podciągi y_i powstały z szyfrowania tekstu jawnego z przesunięciem.
- Przyjmijmy, że znana jest długość klucza m i niech $K=(k_1,k_2,...,k_m)$ będzie szukanym kluczem.
- Oszacujemy wartość $MI_C(y_i, y_j)$.
- Wybieramy losowo po jednym znaku z y_i i z y_j.
- Prawdopodobieństwo otrzymania dwa razy tego samego znaku jest równe $p_{(h-k_i)\%26}p_{(h-k_j)\%26}$, gdzie h=0 dla a, h=1 dla b, h=2 dla c itd.
- $MI_C(y_i, y_j) \approx \sum_{h=0}^{25} p_{h-k_i}p_{h-k_j} = \sum_{h=0}^{25} p_h p_{h+k_i-k_j}$
- Wartość tego oszacowania zależy od różnicy $(k_i - k_j)\%26$, (tj. od tzw. **względnego przesunięcia** y_i i y_j).
- Jeśli względne przesunięcie nie jest zerowe, to wartość oczekiwana $MI_C(y_i, y_j)$ oscyluje między 0.031 i 0.045
- Dla zerowego względnego przesunięcia mamy $MI_C(y_i, y_j) \approx 0.065$.
- Niech y_j będzie przesunięty względem y_i o k_1, gdzie $k_1 = 0,...,25$.
- k_1 znajdziemy wyznaczając wszystkie wartości $MI_C(y_i, y_j^g) = \frac{\sum_{k=0}^{25} f_k f_{k-g}'}{nn'}$ i wybierając tę najbliższą 0.065
- k_1 dla którego wartość $MI_C(y_i, y_j^g)$ jest najbliższa liczbie 0.065 jest względnym przesunięciem ciągu y_j względem y_i.

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

Wyznaczamy **względne przesunięcie**

Przyjęliśmy hipotezę, że klucz ma długość 5. Obliczamy możliwe wartości $MI_C(x_i, y_i^g) = \frac{\sum_{i=0}^{25} f_i f'_{i-g}}{nn'}$

I	j	Wartości $MI_C(x_i, y_i^g)$
1	2	0.02897, 0.02771, 0.02822, 0.03401, 0.03956, 0.03754, 0.02646, 0.02570, 0.05215, **0.06853**, 0.04485, 0.02646, 0.03779, 0.04334, 0.03779, 0.04384, 0.03779, 0.02872, 0.04182, 0.04107, 0.03401, 0.03729, 0.05165, 0.04560, 0.04233, 0.03679,
1	3	0.03956, 0.03376, 0.04006, 0.03427, 0.02847, 0.05392, 0.04888, 0.03376, 0.02973, 0.05644, 0.05089, 0.04560, 0.03981, 0.04056, 0.03603, 0.03779, 0.03275, 0.02746, 0.03779, 0.03653, 0.03175, 0.03729, 0.05518, 0.02973, 0.02494, 0.03704,
1	4	0.03405, 0.04301, 0.02586, 0.02714, 0.03866, 0.04967, 0.04045, 0.03251, 0.02995, 0.03405, 0.03943, 0.04480, 0.04403, 0.03431, 0.03917, 0.04557, 0.04455, 0.03789, 0.05556, 0.04711, 0.03251, 0.02714, 0.03968, 0.03789, 0.03994, 0.03507,
1	5	0.04301, 0.03303, 0.02842, 0.04608, 0.04301, 0.04455, 0.03917, 0.03175, 0.02663, 0.03072, 0.03610, 0.04071, 0.04173, 0.02432, 0.01971, 0.04839, **0.07040**, 0.04403, 0.02867, 0.03866, 0.04403, 0.04301, 0.04736, 0.03379, 0.02637, 0.04634,
2	3	0.04611, 0.04888, 0.04132, 0.03225, 0.03603, 0.03527, 0.03679, 0.03023, 0.02469, 0.03981, 0.03452, 0.02973, 0.04056, **0.06778**, 0.04107, 0.03301, 0.03754, 0.04510, 0.03301, 0.03326, 0.02797, 0.03376, 0.04586, 0.05291, 0.04208, 0.03049,
2	4	0.04608, 0.03482, 0.04378, 0.04455, 0.03405, 0.03149, 0.04071, 0.04583, 0.04045, 0.04813, 0.04480, 0.03379, 0.02407, 0.02842, 0.04250, 0.03968, 0.02688, 0.03482, 0.05018, 0.03507, 0.03251, 0.04019, **0.05684**, 0.04327, 0.02867, 0.02842,
2	5	0.03328, 0.03328, 0.03687, 0.04685, 0.02688, 0.01818, 0.04378, **0.08090**, 0.05069, 0.02995, 0.03123, 0.04506, 0.03943, 0.03738, 0.02765, 0.02688, 0.03149, 0.03994, 0.04045, 0.03763, 0.04147, 0.04608, 0.04531, 0.04301, 0.03533, 0.03098,
3	4	0.03866, 0.03610, 0.04071, 0.03354, 0.03687, 0.06068, 0.03533, 0.04122, 0.02970, 0.05863, 0.03507, 0.03584, 0.03431, 0.05376, 0.03098, 0.03251, 0.03559, 0.03687, 0.03610, 0.02867, 0.04608, 0.03251, 0.05172, 0.03277, 0.03482, 0.03098,
3	5	0.03559, 0.03405, 0.03405, 0.03635, 0.03021, 0.04378, 0.04378, 0.05018, 0.02560, 0.04122, 0.05197, 0.05069, 0.03584, 0.03200, 0.03328, 0.03379, 0.05223, 0.03175, 0.02714, 0.03098, **0.07220**, 0.03584, 0.03456, 0.03251, 0.04327, 0.02714,
4	5	0.05203, 0.03876, 0.03330, 0.03850, 0.04162, 0.04318, 0.03746, 0.04865, 0.02888, 0.02836, 0.03694, **0.06139**, 0.03330, 0.03382, 0.03226, 0.05281, 0.03434, 0.02732, 0.03928, 0.04344, 0.03356, 0.02732, 0.03044, 0.03902, 0.04839, 0.03564,

Tab. Rzeczywiste wzajemne indeksy zgodności.

Wartości podkreślone wskazują, że:
względne przesunięcie y1 i y2 jest równe 9,
względne przesunięcie y1 i y5 jest równe 16,
względne przesunięcie y2 i y3 jest równe 13,
względne przesunięcie y2 i y4 jest równe 22,
względne przesunięcie y2 i y5 jest równe 7,
względne przesunięcie y3 i y5 jest równe 20,
względne przesunięcie y4 i y5 jest równe 11.

Na tej podstawie otrzymujemy układ równań
k1 – k2 = 9,
k1 – k5 = 16,
k2 – k3 = 13,
k2 – k4 = 22
k2 – k5 = 7,
k3 – k5 = 20,
k4 – k5 = 11
Stąd

k2 = k1 – 9
k5 = k1 – 16
k3 = k2 – 13 = k1 – 9 – 13 = k1 – 22
k4 = k5 + 11 = k1 – 16 + 11 = k1 – 5

układ równań rozwiązujemy w arytmetyce mod 26, więc
k2 = k1 + 17
k3 = k1 + 4
k4 = k1 + 21
k5 = k1 + 10

Ostatecznie z dużym prawdopodobieństwem stwierdzamy, że klucz jest postaci (k1, k1+17, k1+4, k1+21, k1+10) dla pewnego $k_1 \in Z_{26}$.

```
---
class Vigenere_krypt{
...
//metoda Friedmana
public double[] Friedman(String s){
int[] Tfreq = new int[26];
double[] TIc = new double[26];
int sum, N, d; double Ic, SIc; String s1="";
for(int k=1; k<26; k++){
  SIc=0;                                          //System.out.println(""+k);
  for(int k1=0; k1<k; k1++){
    for(int i=0; i<26; i++)Tfreq[i]=0;
    s1=""; sum=0; N=0;
    for(int i=0; i<s.length(); i++)
      if(i%k==k1){
        Tfreq[(int)s.charAt(i)-'a']++;
        N++;}
    for(int i=0; i<26; i++)s1+=Tfreq[i]+",";    //System.out.println(s1);
    for(int i=0; i<26; i++)sum+=(Tfreq[i]*(Tfreq[i]-1));
    d = N*(N-1);
    Ic = (double)sum/d;                         //System.out.println("Ic="+Ic);
    SIc+=Ic;}
  TIc[k]=(double)SIc/k;}
return TIc;}

//wzajemne indeksy zgodności
public void wzInZg(String s, int m){
int[][] Tfreq = new int[m][26];
int[] TN = new int[m];
for(int i=0; i<m; i++) Tfreq[i] = new int[26];

for(int k=0; k<m; k++)
  for(int i=0; i<s.length(); i++){
    Tfreq[i%m][(int)s.charAt(i)-'a']++;
    TN[i%m]++;}

int sum; double MIc=0; String sij="";
for(int i=0; i<m; i++)
  for(int j=i+1; j<m; j++){sij="";
    for(int g=0; g<26; g++){
      sum=0;
      for(int k=0; k<26; k++)
        sum+=Tfreq[i][k]*Tfreq[j][(k-g+26)%26];
      MIc=(double)sum/(TN[i]*TN[j]);
      sij+=MIc+", ";}
    System.out.println(sij);}}
...
}
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego