Gimnazija Bežigrad, Ljubljana

RAZBIJANJE SUBSTITUCIJSKE ŠIFRE

RAZISKOVALNA NALOGA

RAČUNALNIŠTVO IN INFORMATIKA

JAŠA KNAP

3. letnik

Mentor: dr. Aleksandar JURIŠIĆ

Somentorica: Jasna KOS

Ljubljana, marec 2020

Zahvaljujem se svojemu mentorju Aleksandru Jurišiću za usmerjanje pri raziskovanju in somentorici Jasni Kos za pomoč in svetovanje pri pisanju naloge.

Vsebina

Povzetek

Naloga opisuje učinkovito strategijo in računalniški program za razbijanje enoabecedne substitucijske šifre. Program z uporabo grobe sile ugotovi, kako so zašifrirane črke A, E, I, O, N, R, T in S, ki v slovenščini predstavljajo več kot 60% vseh črk, kar razbije šifro. Ko bo program optimiziran, bo vgrajen v Kriprogram – javno dostopno spletno stran za popularizacijo kriptografije v Sloveniji.

KLJUČNE BESEDE: kriptografija, računalniški program, razbijanje substitucijske šifre, matrike, Kriptogram

KEY WORDS: cryptography, computer program, attack on substitution cipher, matrices, Kriptogram

1. Uvod

Kriptografija je zaradi svoje pomembnosti v vsakdanjem življenju in zaradi vseh zgodb, ki jo spremljajo, vsakomur zanimiva. Ko sem našel razpis za raziskovalno nalogo, ki je bil objavljen v okviru projekta SKOZ, ki ga financira Ministrstvo za izobraževanje, znanost in šport, sem se odločil, da svoje znanje o kriptografiji nadgradim. Najprej sem začel prebirati poljudno literaturo, potem pa sem osnovno znanje nadgrajeval z branjem strokovne literature in se nato lotil raziskovalnega dela.

Cilj naloge je bil najti učinkovito strategijo za reševanje substitucijske šifre, nato pa napisati program, ki bi nam čimbolj olajšal njeno reševanje pri razmeroma kratkih šifriranih besedilih.

Po predstavitvi osnovnih šifer in matematičnih vsebin, ki jih potrebujemo v nadaljevanju, so predstavljene strategije za ročno reševanje substitucijske šifre. Te so osnova za sistematično razreševanje kompleksnejšega problema. V prvem koraku analiziramo pogostost pojavljanja posameznih črk in črkovnih zvez v slovenskem jeziku. Uvedemo postopke za iskanje razdalj med samoglasniki. Podatke, zapisane s številskimi vrednostmi, vnašamo v matrike. Ko matrike med sabo primerjamo z ustreznimi kriteriji, ugotovimo, katere črkovne zveze so bolj verjetne.

Program, ki sem ga sestavil v Pythonu, z uporabo grobe sile ugotovi, kako so zašifrirane črke A, E, I, O, N, R, T in S, ki v slovenščini predstavljajo več kot 60% vseh črk, kar razbije šifro.

Naloge sem se lotil, ker se mi je zdela zanimiv izziv, pri katerem sem lahko izboljšal svoje znanje matematike in programiranja.

2. Kriptografija

Kriptografija - sestavljena iz grških besed kryptos (skrit) in graphein (pisati) - je veda o matematičnih tehnikah za dosego informacijske varnosti, kot so zaupnost, celovitost podatkov, overjanje identitete. Ukvarja se s študijem in razvojem metod za šifriranje, kjer se običajno uporabljajo skrivni ključi, s katerimi je mogoče dešifrirati šifrirano sporočilo ali informacijo. Uporablja se za preprečevanje in odkrivanje zlorab in ostalih zlonamernih dejanj.

Šifra ali šifriranje je v kriptografiji algoritem za prirejanje tajnopisa glede na šifrirni ključ. Pri ne-tehnični uporabi beseda »koda« po navadi pomeni »šifro«. Znotraj tehničnih diskusij pa se besedi »šifra« in »koda« nanašata na dva različna koncepta. Pri kodah so besede ali fraze preoblikovane v neko drugo obliko, kar po navadi skrajša sporočilo – torej delujejo na stopnji pomenov. Na drugi strani pa šifre delujejo na nižji stopnji, na stopnji individualnih črk in imajo več lastnosti teksta[1]. Med najbolj osnovne šifre spadajo premešalka (transpozicijska šifra), zamenjalka (substitucijska šifra), Viegenerjeva šifra in metoda s kodnimi knjigami.

Osnovni namen kriptografije je omogočiti sporočevalcu in naslovniku, da se sporazumevata preko nezaščitenega kanala, tako da nasprotnik, ki želi izvedeti vsebino njunega pogovora, tega ne more razumeti. Takšna zveza je lahko na primer telefonska linija ali računalniška mreža. Sporočilo, ki ga želi sporočevalec posredovati naslovniku, imenujemo čistopis, ki je običajno besedilo ali številski podatki. Sporočevalec s pomočjo vnaprej določenega ključa čistopis zašifrira in dobljeni tajnopis pošlje po kanalu. Nasprotnik, ki prebere tajnopis, ne more določiti čistopisa, medtem ko naslovnik, ki pozna šifrirni ključ, lahko dešifrira tajnopis in rekonstruira čistopis. To lahko opišemo bolj formalno v matematičnem jeziku.

Simetrični kriptosistem je peterica $(\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{K}, \boldsymbol{\varepsilon}, \boldsymbol{D})$, za katero veljajo naslednje 4 lastnosti:

1. $\boldsymbol{P}$ je končna množica možnih čistopisov
2. $\boldsymbol{C}$ je končna množica možnih tajnopisov
3. $\boldsymbol{K}$ je končna množica možnih ključev

Najpomembnejša je četrta lastnost, ki nam pove, da če čistopis zašifriramo in dobljeni tajnopis dešifriramo, dobimo začetni čistopis.[2]

4. Za vsak ključ $\mathrm{K} \in \boldsymbol{K}$ se da učinkovito priti do šifrirnega postopka $e_K \in \boldsymbol{\varepsilon}$ in dešifrirnega postopka $d_K \in \boldsymbol{D}$. $e_K : \boldsymbol{P} \to \boldsymbol{C}$ in $d_K : \boldsymbol{C} \to \boldsymbol{P}$ sta taki funkciji, da velja $d_K(e_K(x)) = x$ za vsak čistopis $x \in \boldsymbol{P}$.

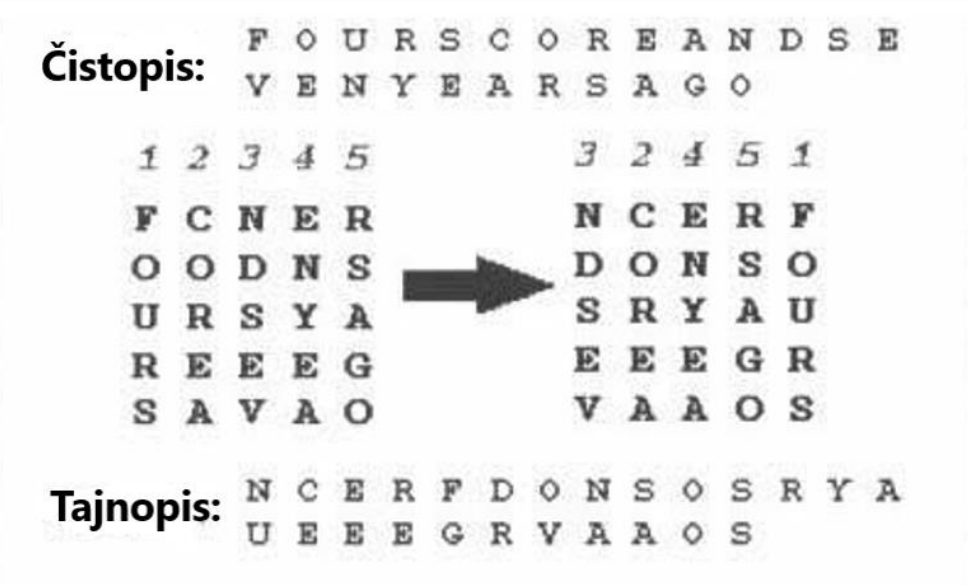
Pri moderni kriptografiji ne gre za to, da bi napadalec moral poiskati algoritem s katerim je bila informacija zašifrirana, pač pa je predpostavka, da tega običajno pozna, manjka mu pa ključ oziroma geslo, s katerim lahko informacijo dešifrira.[2]

2.1 Transpozicijska šifra

Pri preprosti transpozicijski šifri najprej razdelimo čistopis na enako dolge odseke. Nato v vsakemu odseku zamenjamo vrstni red znakov glede na permutacijo P, ki je naš šifrirni ključ. Zato velja, da ima tajnopis vse znake, ki so bili v čistopisu.

Dolžina permutacije P je perioda transpozicijske šifre. Če je naš ključ $P = \{4, 2, 5, 3, 6, 1\}$, je perioda takšne šifre 6. V tem primeru 4. znak vsakega odseka premaknemo na prvo mesto, 2. znak ostane na drugem mestu, 5. znak premaknemo na tretje mesto, itd.[3]

Transpozicijo lahko izvedemo na različne načine, med njimi so zelo pogosti grafični. Eden izmed njih je prikazan na spodnji sliki. Znake posameznega odseka dolžine n^2 po vrsti zapišemo v stolpce. Nato poljubno zamenjamo vrstni red stolpcev. Šifrirano besedilo dobimo tako, da zaporedoma zapišemo znake po vrsticah, od zgornje do spodnje.



Slika 1: Primer transpozicijske šifre s periodo 25[13]

2.2 Substitucijska šifra

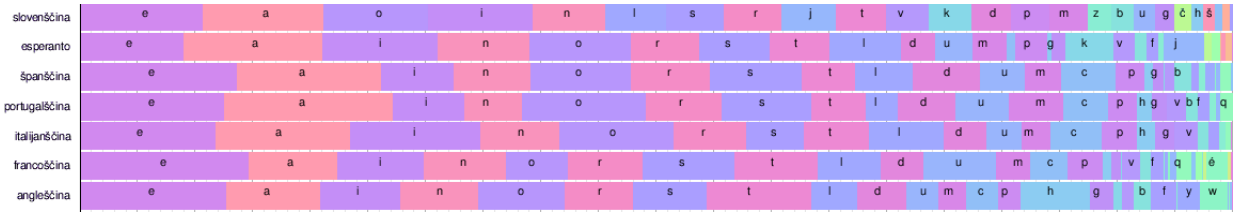
Najpreprostejša vrsta substitucijske šifre je enoabecedna oziroma monoalfabetna, kjer je šifrirni ključ poljubna permutacija znakov abecede, v kateri je zapisano sporočilo. Vsak znak sporočila glede na ključ preslikamo v nek drug znak. Če je v abecedi n znakov, je različnih možnih ključev $n!$. V slovenščini je to $25! \approx 1{,}55 \cdot 10^{25}$ možnih ključev.[4]

A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž
Š	A	J	Ž	M	D	C	V	Z	O	T	U	R	P	G	K	N	F	L	E	Č	I	B	H	S

Slika 2: Primer naključne permutacije črk slovenske abecede

Največja pomanjkljivost substitucijskih šifer je ohranjanje deleža črk, ki se v govorjenih jezikih ne pojavljajo enako pogosto. V slovenščini črka E na primer predstavlja okoli 10% vseh črk. Približno enako velja za A, sledita O in I s približno 9%. Najredkejše črke pa so C (≈ 0,66%), Ž (≈ 0,64%) ter F (≈ 0,11%). Če bi bil zašifriran tudi presledek, bi to hitro opazili, saj bi predstavljal okoli 17% vseh znakov.

Samoglasnikov je v praktično kateremkoli besedilu med 35-40%, saj bi ga bilo v nasprotnem primeru težko izgovarjati. Poleg tega so razdalje med samoglasniki razmeroma kratke, med dvema redko stojijo več kot trije soglasniki. Hkrati pa samoglasniki v slovenščini zelo redko stojijo eden zraven drugega. Zaradi vseh teh lastnosti je substitucijsko šifro mogoče enostavno razbiti.



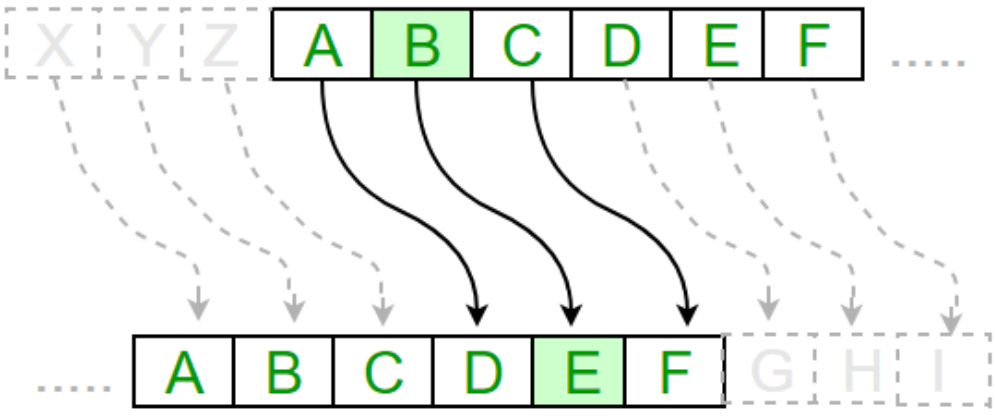
Slika 3: Relativna frekvenca črk v različnih jezikih[14]

Nekoliko varnejše od enoabecednih so večabecedne oziroma polialfabetne substitucijske šifre. Pri njih čistopis zašifriramo z uporabo več substitucijskih šifer. Lahko bi se na primer odločili, da bi vsako tretjo črko zašifrirali z drugačnim ključem kot ostale. Delež črk je v večabecednih substitucijskih šifrah bolj izravnan kot pri enoabecednih, zato jih je tudi težje razbiti.[5]

Če substitucijo izvedemo nad večjim številom črk (nad pari, trojicami, raznimi kombinacijami…) govorimo o večgrafski substitucijski šifri.[4]

2.2.1 Cezarjeva šifra

Cezarjeva šifra je posebna oblika substitucijske šifre, kjer vsak znak zamenjamo z n-tim naslednjim po abecedi. Poimenovana je po Juliju Cezarju, ki je v svojih sporočilih uporabljal modul 3. Velja za izredno šibko šifro, saj za razbijanje ustreza pregled n-zamikov, kjer je n število znakov v abecedi čistopisa. Zato praktično ni uporabna, razen kot sestavni del kompleksnejših šifer.



Slika 4: Šifra, ki jo je uporabljal Julij Cezar[15]

2.3 Viegenerjeva šifra

Pri šifriranju z Viegenerjevo šifro je izvedemo več zaporednih Cezarjevih šifriranj, pri čemer se zamik spreminja. Izberemo si neko besedo, ki bo predstavljala naš ključ in določala zamik. Naj bo to recimo AVTO. Radi bi zašifrirali besedilo POZDRAVLJENSVET. Pogledamo v Viegenerjev kvadrat: prva črka tajnopisa bo presečišče P in A, torej P. Druga bo presečišče O in V, torej J, tretja presečišče Z in T itd.[6]

Čistopis: **P**OZDRAVLJENSVET

Ključ: **A**VTOAVTOAVTOAVT

Tajnopis: **P**LSŠRVRCJCIHVCO

	A	**B**	**C**	**Č**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**R**	**S**	**Š**	**T**	**U**	**V**	**Z**	**Ž**
A	A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž
B	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A
C	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B
Č	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C
D	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č
E	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D
F	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E
G	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F
H	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G
I	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H
J	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I
K	K	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J
L	L	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K
M	M	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L
N	N	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L	M
O	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N
P	P	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O
R	R	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P
S	S	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R
Š	Š	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S
T	T	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š
U	U	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T
V	V	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U
Z	Z	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V
Ž	Ž	A	B	C	Č	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	Š	T	U	V	Z

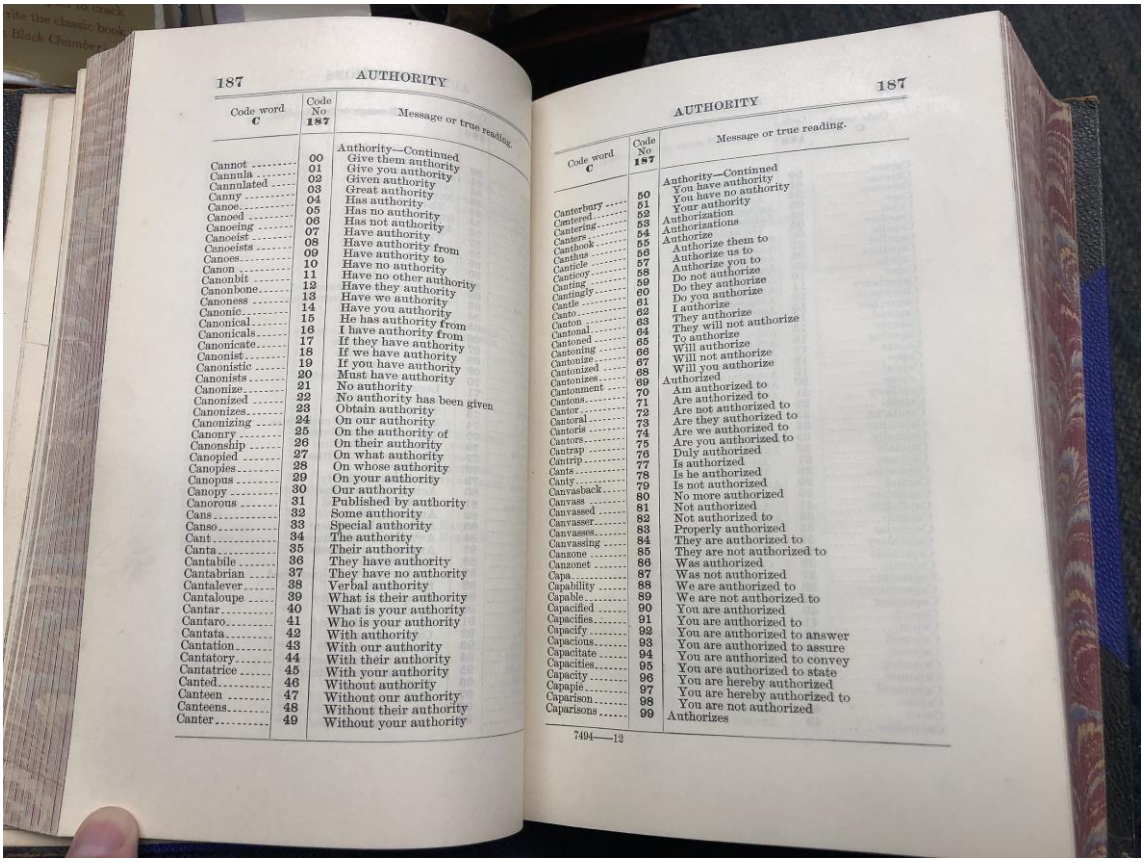
Slika 5: Viegenerjev kvadrat

Viegenerjev kvadrat lahko za abecedo z n-znaki konstruiramo tako, da narišemo prazno $(n+1) \times (n+1)$ tabelo. Nato v najvišjo vrstico in najbolj lev stolpec po vrsti zapišemo števila $0, 1, \ldots, n-1$, tako da ostane zgornji levi kvadrat prazen, zraven njega pa sta 0. Vrednost vsakega izmed ostalih n^2 polj izračunamo tako, da seštejemo vrednosti najvišje vrstice in najbolj levega stolpca ter najdemo celoštevilski ostanek pri deljenju z n. Nato vsa števila nadomestimo z znaki, kot si sledijo po abecedi.

2.4 Metoda s kodnimi knjigami

Kodne knjige so knjige, s pomočjo katerih lahko zakodiramo besedilo. Na začetku so jih uporabljali predvsem za zgoščevanje sporočil, ki so jih pošiljali s telegrafi. Pogoste besedne zveze so zamenjali s tri do pet črkovnimi kodami, kar je olajšalo pošiljanje in ga pocenilo.

Med drugo svetovno vojno je bila uporaba kodnih knjig zelo pogosta. Razbijanje kode je zahtevno, vendar ni nemogoče - če razbijalec kodno knjigo najde, sporočilo ni več skrivno. Pri kodiranju s kodnimi knjigami se torej zanašamo na to, da kode ne bodo razkrite, kar pa je na dolgi rok malo verjetno, zato strokovnjaki in organizacije takšno kodiranje odsvetujejo.[7]



187 AUTHORITY

Code word C	Code No. 187	Message or true reading.
		Authority—Continued
Cannot	00	Give them authority
Cannula	01	Give you authority
Cannulated	02	Given authority
Canny	03	Great authority
Canoe	04	Has authority
Canoed	05	Has no authority
Canoeing	06	Has not authority
Canoeist	07	Have authority
Canoeists	08	Have authority from
Canoes	09	Have authority to
Canon	10	Have no authority
Canonbit	11	Have no other authority
Canonbone	12	Have they authority
Canoness	13	Have we authority
Canonic	14	Have you authority
Canonical	15	He has authority from
Canonicals	16	I have authority from
Canonicate	17	If they have authority
Canonist	18	If we have authority
Canonistic	19	If you have authority
Canonists	20	Must have authority
Canonize	21	No authority
Canonized	22	No authority has been given
Canonizes	23	Obtain authority
Canonizing	24	On our authority
Canonry	25	On the authority of
Canonship	26	On their authority
Canopied	27	On what authority
Canopies	28	On whose authority
Canopus	29	On your authority
Canopy	30	Our authority
Canorous	31	Published by authority
Cans	32	Some authority
Canso	33	Special authority
Cant	34	The authority
Canta	35	Their authority
Cantabile	36	They have authority
Cantabrian	37	They have no authority
Cantalever	38	Verbal authority
Cantaloupe	39	What is their authority
Cantar	40	What is your authority
Cantaro	41	Who is your authority
Cantata	42	With authority
Cantation	43	With our authority
Cantatory	44	With their authority
Cantatrice	45	With your authority
Canted	46	Without authority
Canteen	47	Without our authority
Canteens	48	Without their authority
Canter	49	Without your authority

AUTHORITY 187

Code word C	Code No. 187	Message or true reading.
		Authority—Continued
Canterbury	50	You have authority
Cantered	51	You have no authority
Cantering	52	Your authority
Canters	53	Authorization
Canthook	54	Authorizations
Canthus	55	Authorize
Canticle	56	Authorize them to
Cantoxy	57	Authorize us to
Canting	58	Authorize you to
Cantingly	59	Do not authorize
Cantle	60	Do they authorize
Canto	61	Do you authorize
Canton	62	I authorize
Cantonal	63	They authorize
Cantoned	64	They will not authorize
Cantoning	65	To authorize
Cantonize	66	Will authorize
Cantonized	67	Will not authorize
Cantonizes	68	Will you authorize
Cantonment	69	Authorized
Cantons	70	Am authorized to
Cantor	71	Are authorized to
Cantoral	72	Are not authorized to
Cantoris	73	Are they authorized to
Cantors	74	Are we authorized to
Cantrap	75	Are you authorized to
Cantrip	76	Duly authorized
Cants	77	Is authorized
Canty	78	Is he authorized
Canvasback	79	Is not authorized
Canvass	80	No more authorized
Canvassed	81	Not authorized
Canvasser	82	Not authorized to
Canvasses	83	Properly authorized
Canvassing	84	They are authorized to
Canzone	85	They are not authorized to
Canzonet	86	Was authorized
Capa	87	Was not authorized
Capability	88	We are authorized to
Capable	89	We are not authorized to
Capacified	90	You are authorized
Capacifies	91	You are authorized to
Capacify	92	You are authorized to answer
Capacious	93	You are authorized to assure
Capacitate	94	You are authorized to convey
Capacities	95	You are authorized to state
Capacity	96	You are hereby authorized
Capapie	97	You are hereby authorized to
Caparison	98	You are not authorized
Caparisons	99	Authorizes

7494—12

Slika 6: Kodna knjiga. Na levi polovici strani so zapisane kode za besedne zveze na desni polovici[16]

3. Matematično ozadje

3.1 Matrike

Pravokotno shemo $m \times n$ števil, razporejenih v m vrstic in n stolpcev imenujemo matrika dimenzije $m \times n$. Števila v shemi imenujemo elementi matrike. Element $a_{i,j}$ leži v i-ti vrstici in j-tem stolpcu[8].

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

Matrike običajno označujemo z velikimi tiskanimi črkami. Krajše jih zapišemo v obliki $A = [a_{i,j}];\ i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n.$

Ničelna matrika je takšna matrika, ki ima vse elemente enake 0.

Matrike istih dimenzij lahko seštevamo ali odštevamo. To naredimo tako, da seštejemo oziroma odštejemo istoležne elemente:

$$[a_{i,j}] \pm [b_{i,j}] = [a_{i,j} \pm b_{i,j}];\ i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n.$$

Ker drugih računskih operacij ne bomo uporabljali, jih ne omenjam.

3.2 Metrični prostor

Metrični prostor je neprazna množica M, opremljena s preslikavo $d : M\ x\ M \rightarrow [0, \infty)$, ki ima naslednje lastnosti za vse $x, y, z \in M$:

- $d(x, y) \geq 0;\ d(x, y) = 0 \Leftrightarrow x = y$ (nenegativnost)
- $d(x, y) = d(y, x)$ (simetričnost)
- $d(x, y) \leq d(x, z) + d(z, y)$ (trikotniška neenakost)

Preslikavi *d* pravimo metrika ali razdalja.[4] Razdaljo bomo uporabljali pri primerjanju matrik deleža posameznih parov črk v besedilu. Večja, kot je razdalja, bolj se besedili med seboj razlikujeta. Zato lahko ugotovimo, katere permutacije črk so manj verjetne od drugih. Med najbolj osnovne sodita evklidska in Manhattanova razdalja. Za matriki velikosti $n \times m$ z ju izračunamo po naslednjih formulah:

Evklidska razdalja:

$$\mathrm{d(x,y)} = \sqrt{\sum_{m,n} \left(x_{m,n} - y_{m,n}\right)^2}$$

Manhattanova razdalja:

$$\mathrm{d(x,y)} = \sum_{m,n} \left| x_{m,n} - y_{m,n} \right|$$

kjer so $x_{1,1}, \ldots, x_{m,n}$ elementi prve matrike, $y_{1,1}, \ldots, y_{m,n}$ pa elementi druge.[2]

Ker koren pri evklidski razdalji zgolj zmanjša končno vrednost razdalje, ga bomo izpustili in uporabili

$$\mathrm{d}(\mathrm{x},\mathrm{y}) = \sum_{m,n} \left(x_{m,n} - y_{m,n}\right)^2.$$

3.3 Standardni odklon

Standardni odklon σ (ali standardna deviacija) je odklon podatkov od aritmetične sredine in ga izračunamo po formuli:

$$\sigma = \sqrt{\frac{(x_1-\bar{x})^2+(x_2-\bar{x})^2+\cdots+(x_N-\bar{x})^2}{N}};$$

kjer so $x_1, x_2, \ldots, x_N$ vrednosti podatkov, $\bar{x}$ pa njihova aritmetična sredina.[8]

Večji kot je standardni odklon, bolj so podatki razpršeni okoli aritmetične sredine.

3.4 Permutacije, variacije in kombinacije[9]

Permutacije so razporeditve danih n elementov na n prostih mest.

Če so vsi elementi med seboj različni, so to permutacije brez ponavljanja.

Število permutacij n elementov brez ponavljanja izračunamo po formuli:

$$P_n = n\,(n-1)\cdots 3 \cdot 2 \cdot 1 = n!$$

Zaradi mnogih razlogov se definira $0! = 1$. Intuitivna razlaga bi lahko bilo dejstvo, da lahko 3 različne predmete na mizi razvrstimo na $3! = 6$ načinov, 2 predmeta na $2! = 2$ načina, 1 predmet pa le na $1! = 1$ način. Enako lahko naredimo z 0 predmeti: pustimo jih tako kot so, torej imamo $0! = 1$ način.

Variacije brez ponavljanja so razporeditve n različnih elementov na r prostih mest. Pri tem je $r < n$, zato ostane nekaj elementov nerazporejenih.

Število variacij brez ponavljanja izračunamo po formuli:

$$V_n^r = \frac{n!}{(n-r)!}$$

Če pri variacijah zanemarimo vrstni red in opazujemo samo, kateri elementi so izbrani, dobimo kombinacije.

Kombinacije brez ponavljanja so izbire r (različnih) elementov izmed n različnih elementov, ki so na voljo.

Število kombinacij brez ponavljanja izračunamo po formuli:

$$C_n^r = \frac{n!}{r!\,(n-r)!}$$

Desno stran enačbe lahko zapišemo krajše z binomskim simbolom:

$$C_n^r = \frac{n!}{r!\,(n-r)!} = \binom{n}{r}$$

Poznamo tudi permutacije, variacije in kombinacije s ponavljanjem, kjer se isti elementi lahko pojavijo večkrat. Ker jih v raziskovalni nalogi nisem uporabljal, jih ne bom podrobneje opisal.

4. Razbijanje substitucijske šifre

Če hočemo razbiti neznano šifro, moramo najprej ugotoviti, za katero vrsto šifre gre. Zato moramo preučiti lastnosti tajnopisa:

- če je relativna frekvenca črk v tajnopisu praktično enaka relativni frekvenci črk v slovenščini, gre zelo verjetno za transpozicijsko šifro
- če je relativna frekvenca črk v tajnopisu različna relativni frekvenci črk v slovenščini, znaki pa se v tajnopisu pojavljajo različno pogosto, gre verjetno za enoabecedno substitucijsko šifro
- če je relativna frekvenca črk v tajnopisu različna relativni frekvenci črk v slovenščini, vsi znaki v tajnopisu pa se pojavljajo približno enako pogosto, gre morda za večabecedno substitucijsko ali Viegenerjevo šifro, lahko pa bi šlo tudi za kakšno kompleksnejšo

Pogosto se splača tudi preveriti, ali je sporočilo šifrirano s Cezarjevo šifro, ki je kljub svoji šibkosti med laiki še vedno priljubljena, morda prav zaradi svoje preprostosti. Seveda pa ne pride v poštev za kakršno koli strokovno rabo.

Če bi bil v enoabecedni substitucijski šifri zašifriran tudi presledek, bi ga zlahka prepoznali, saj predstavlja približno eno šestino vseh znakov v slovenščini, kar je veliko več od vseh ostalih znakov.

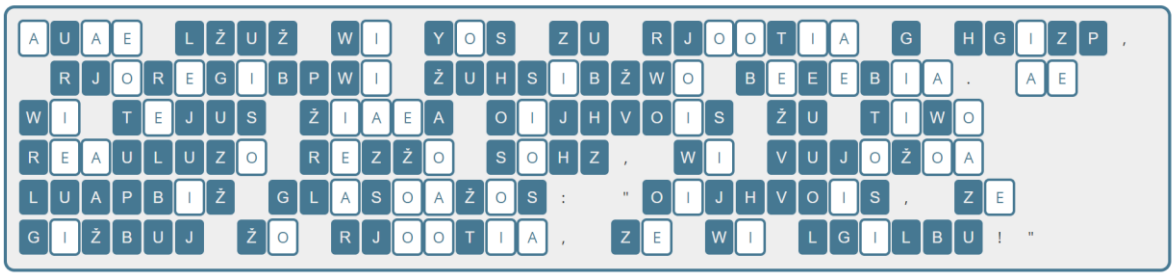
Lahko bi se zgodilo, da bi bile zašifrirane tudi tuje črke (X, Y, Q, W,…), številke in nasploh katerikoli znaki (»/, &, # ,…«). V večini primerov verjetno niti ne bi močno ovirale, morda bi celo pomagale z namigovanjem na vsebino. Poleg tega so taki znaki v slovenščini redki, zato med razbijanjem šifre nimajo velikega vpliva, saj nam posredujejo malo novih informacij čistopisa.

Za lažje delo bom zato v nalogi privzel, da kot vhodni podatek program prejme tajnopis, ki je zašifriran z enoabecedno substitucijsko šifro. Znaki tega čistopisa so črke slovenske abecede, presledki pa niso zašifrirani.

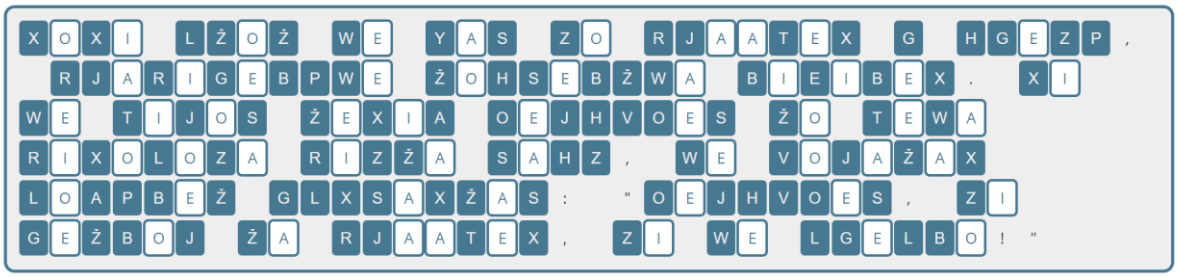
4.1 Iskanje strategije za ročno reševanje substitucijske šifre

Reševanja problema sem se lotil tako, da sem se poskusil izuriti pri reševanju substitucijskih šifer. Primere sem našel na spletni strani Kriptogram[20]. S tem, ko sem jih rešil večino, sem prišel do naslednjih ugotovitev:

1. Najprej si je treba zabeležiti frekvence črk, če to še ni bilo storjeno. Tako lahko približno vidimo, v kaj bi lahko bila zašifrirana posamezna črka tajnopisa.
2. Nato je najbolje ugotoviti, katere črke so zašifrirani samoglasniki, saj tako lažje prepoznamo katero od besed v tajnopisu. Poleg tega lahko pravilno kombinacijo 4 črk uganemo; če bi izbrali napačno, bi v tajnopisu takoj opazili, da bi se pojavljali pari samoglasnikov, kar je v slovenščini zelo redko. Hkrati pa poskušamo najti pravo kombinacijo samoglasnikov.



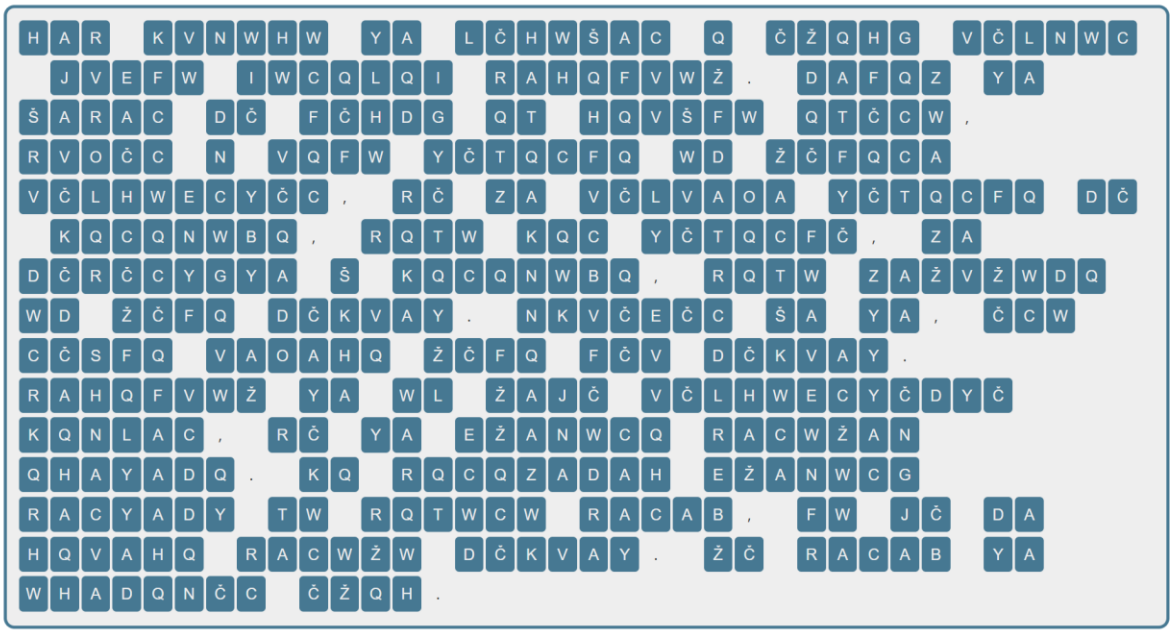
Slika 7: Primer napačno postavljenih samoglasnikov v substitucijski šifri, ki vidno stojijo preveč skupaj



Slika 8: Primer pravilno postavljenih samoglasnikov v substitucijski šifri, saj praktično ne stojijo v parih

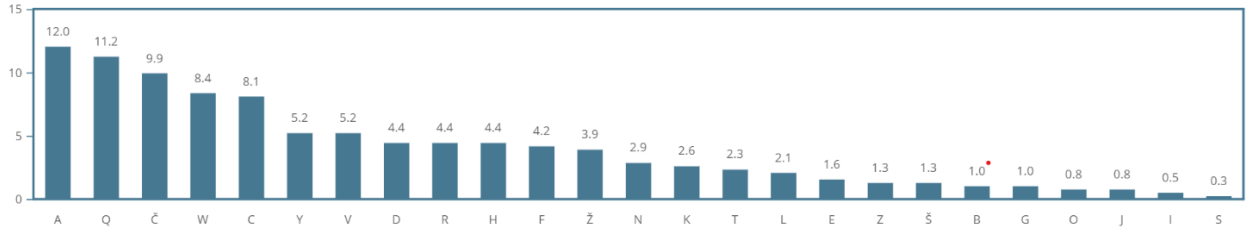
3. Takoj, ko imamo samoglasnike, se je pametno lotiti iskanja črk R in N in J: R spada med najpogostejše črke, njegova posebnost pa je dejstvo, da ga pogosto izgovarjamo kot polglasnik. Zato je verjetnost, da stoji v »luknjah« med samoglasniki, velika. N je trenutno 5. najpogostejša črka v slovenščini, takoj za samoglasniki AEIO. Poleg tega se zelo pogosto pojavlja v parih »AN«, »NA«, »IN«, ki so v slovenščini med najpogostejšimi. J pa je večinoma lahko najti zaradi dejstva, da je par črk »JE« v slovenščini daleč najpogostejši (predstavlja okoli 2,5% vseh parov), uporabljamo ga kot veznik, pojavlja pa se tudi v končnicah besed. Če se v besedilu nek par črk izrazito pogosto ponovi, je velika verjetnost, da gre za »JE«. To je tudi dober način za določanje E-ja, posredno pa tako pogosto lahko točno določimo A, saj A in E po frekvenci običajno rahlo odstopata od I-ja in O-ja.
4. Naslednji korak je iskanje podvojenih črk. Če imamo podvojeni črki znotraj neke besede, nam preostane le nekaj možnosti: D (»oddati«), Z (»izzivati«), O (»poosebljati«), U (»vakuum«), itd. V nasprotnem primeru pa gre verjetno za ime, kar bi nam lahko razkrilo kaj o vsebini tajnopisa.
5. Če imamo enočrkovno besedo, ki je enaka prvi črki za sledečim presledkom, gre običajno za S (»s sorodniki«), Z (»z znanostjo«), V (»v vesolju«), O (»o osebah«) …
6. Poskušamo uganiti kakšno besedo. V pomoč nam je, če okvirno poznamo vsebino tajnopisa: če gre za dnevnik, bi lahko iskali besede »danes«, »popoldne«, »dopoldne«, ali morda besedne zveze »v šoli«, »na treningu« itd. Že če nam uspe uganiti eno samo besedo, postane problem veliko lažji, saj lahko natančneje določimo ostale črke ter morda lažje uganemo vsebino sporočila.
7. Če ne gre, poskušamo uganiti prave kombinacije črk, dokler ne dobimo smiselnih rezultatov.

Poglejmo si na naslednjem primeru:



Slika 9: Tajnopis, ki ga želimo razšifrirati

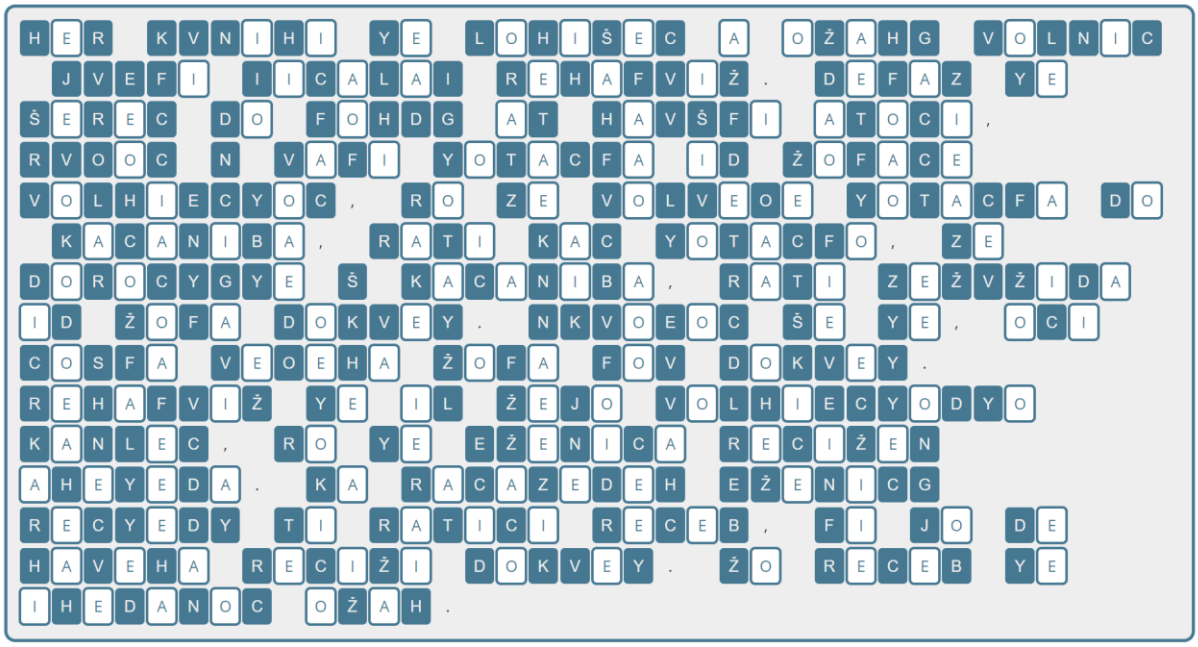
1. Najprej naredimo frekvenčno analizo:



Slika 10: Delež črk v tajnopisu

2. Poiščemo samoglasnike v sporočilu:

Vidimo, da po pogostosti izstopata A in Q, zato gre verjetno za E in A. Poleg tega lahko opazimo, da se v besedilu zelo pogosto pojavlja par »YA«, zato bi lahko šlo za »JE«. Vstavimo A namesto E in Q namesto A. Zdaj moramo uganiti še O in I. S poskušanjem vidimo, da se zdi smiselno Č namesto O in W namesto I.

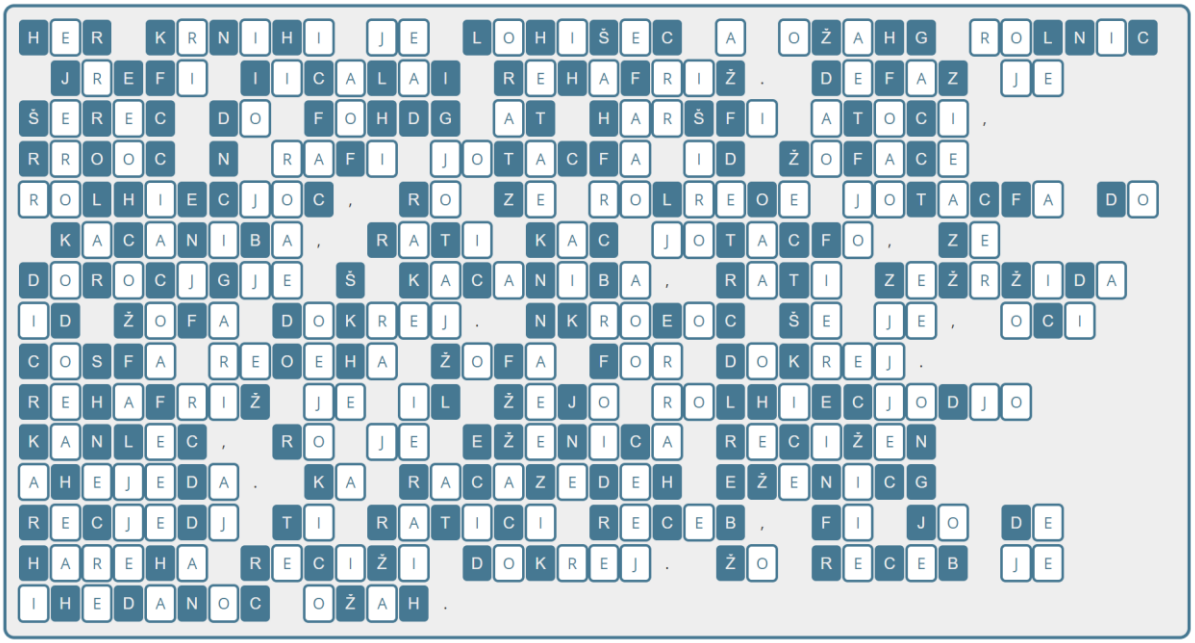


Slika 11: Zelo verjetni samoglasniki

Ker vidimo, da med vstavljenimi črkami praktično ni samoglasniških parov, hkrati pa so razdalje od enega samoglasnika do drugega razmeroma kratke, vemo, da smo skoraj gotovo izbrali pravilne samoglasnike.

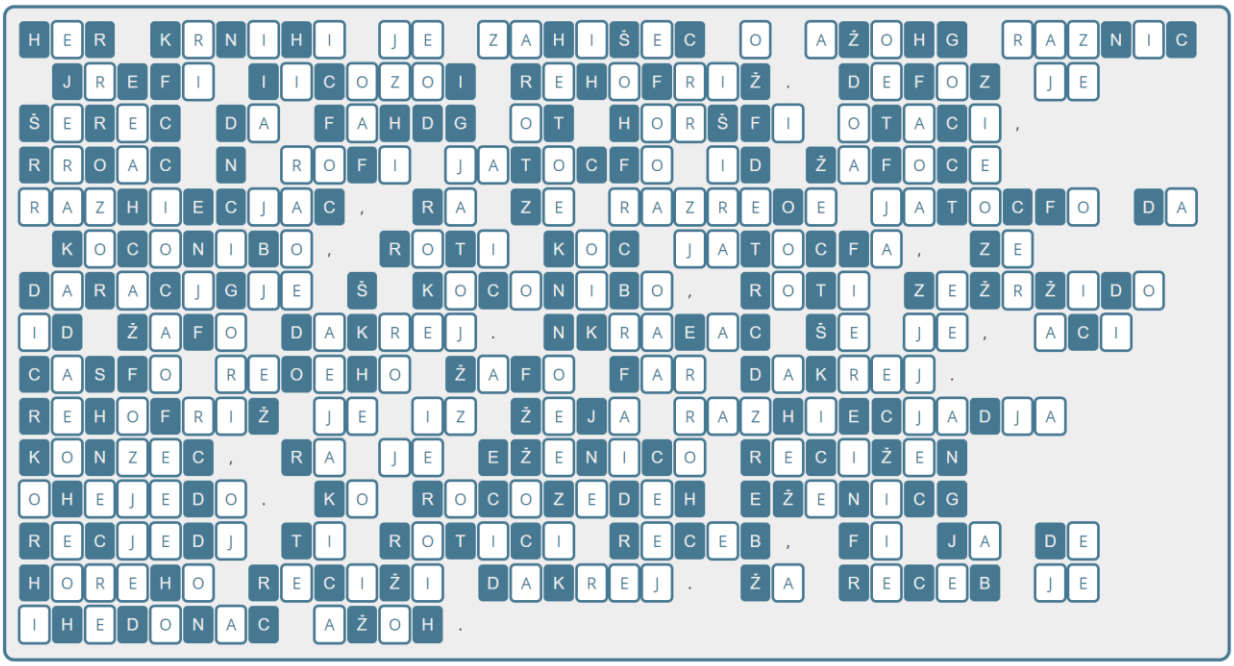
3. Iščemo N, R in J

Ker je »YA« najpogostejši par tajnopisa, predpostavimo, da gre za »JE«. Vstavimo Y namesto J. Zdaj poskusimo poiskati R. Ker ga pogosto izgovarjamo kot polglasnik, je velika verjetnost, da se bo nahajal v eni izmed »lukenj« med samoglasniki. Po premisleku se zdi zelo verjetno, da bi V lahko bil R. Vstavimo R namesto V.



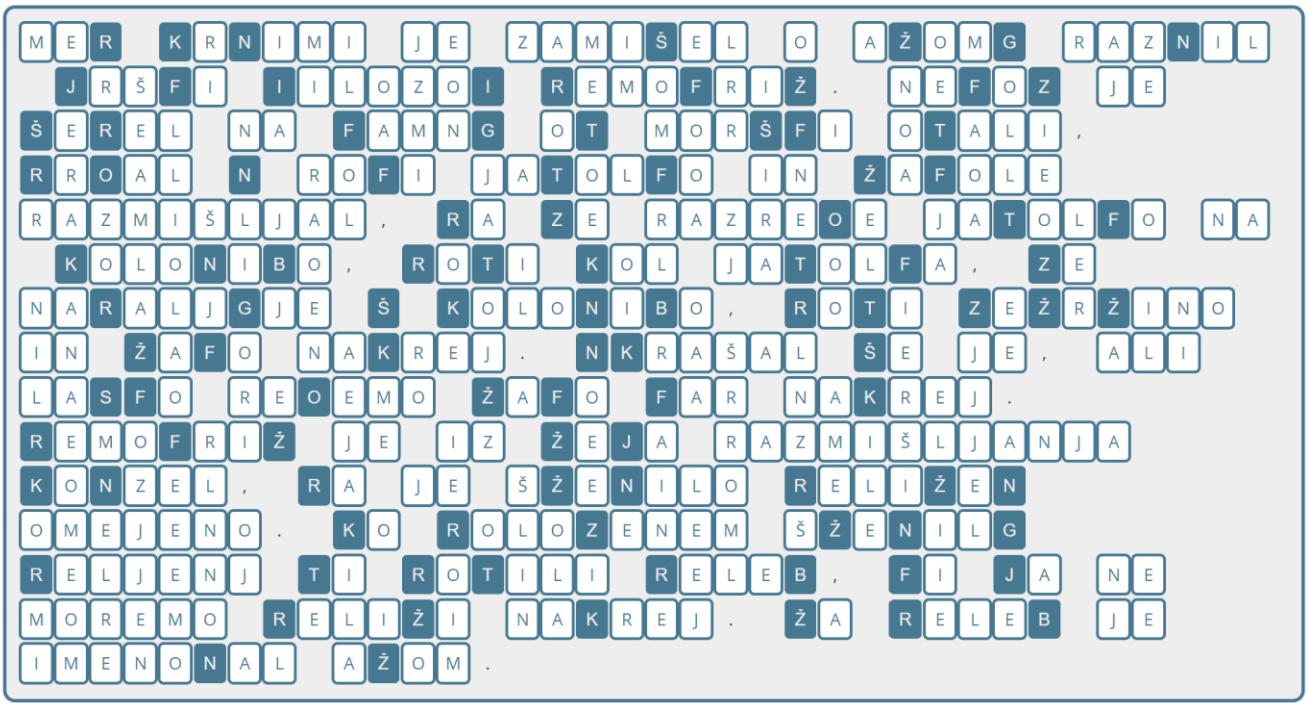
Slika 12: Stanje po vstavljanju črk J in R

Črka A v slovenščini sicer lahko predstavlja besedo, vendar se zdi verjetneje da bi šlo za O. Poskusimo zamenjati A in O. Hkrati poskusimo vstaviti L namesto Z, saj ima L v tajnopisu praktično enako frekvenco, kot Z v slovenščini, hkrati pa v 10. vrstici tvorimo besedo »IZ«.



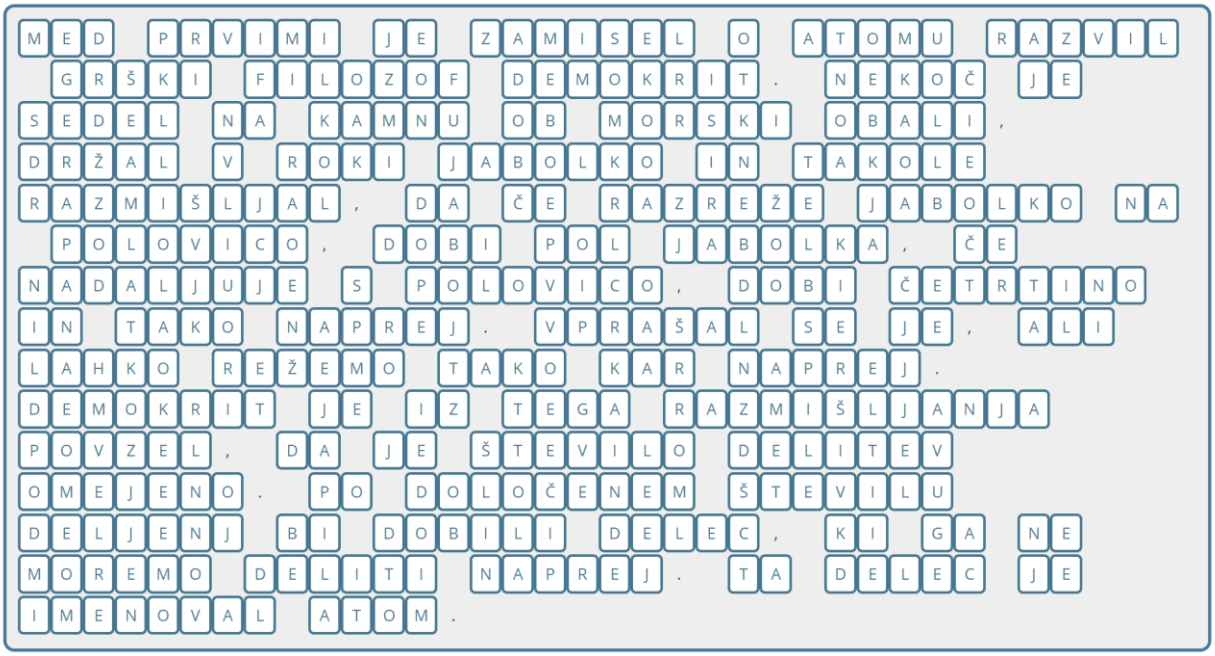
Slika 13: Stanje po zamenjavi A z O ter vstavitvi črke Z

Čeprav se nam sprva ne zdijo vse besede smiselne poskusimo katero uganiti. Opazimo, da bi zadnja beseda 10. vrstice lahko bila »RAZMIŠLJANJA«. Poskusimo:



Slika 14: Po vstavitvi besede "RAZMIŠLJANJA"

Vidimo, da se je sestavilo cel kup besed, kar pomeni, da smo skoraj že pri cilju. Od te točke dalje moramo samo še vstavljati črke tako, da dobimo smiselne besede. Če pri tem ugotovimo, da smo se pri kakšni izmed črk, ki smo jih vstavili s predpostavkami, zmotili, jih ustrezno popravimo. Vstavimo preostale črke:



Slika 15: Po vstavljanju preostalih črk je sporočilo dešifrirano

Kot smo videli, imajo izredno pomembno vlogo pri razbijanju substitucijske šifre predvsem kratke in dolge besede. S pomočjo kratkih lahko hitreje najdemo pravilne kombinacije črk. Dolge besede pa so koristne za ugibanje; nemalokrat se zgodi, da glede na vzorec črk lahko pravilno sklepamo, za katero besedo gre. Potem, ko pravilno uganemo daljo besedo, je šifra že praktično razbita.

4.2 Pisanje računalniškega programa

Program sem pisal v programskem jeziku Python. Njegova glavna prednost je velika izbira modulov, ki nam olajšajo delo. Največ sem uporabljal modul numpy, ki omogoča učinkovito delo z matrikami, še posebej zato, ker je razmeroma dobro optimiziran za izvrševanje programa.

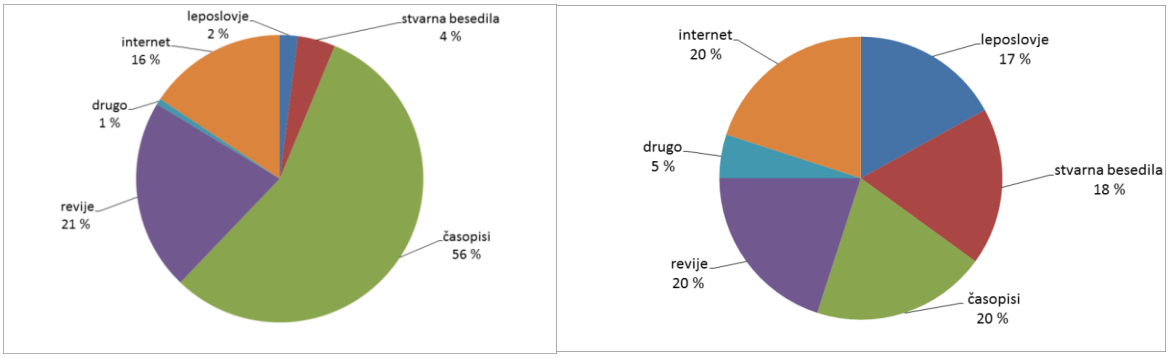
Ko bom program dodatno izboljšal in optimiziral, ga bom prevedel v JavaScript, nato ga bom vgradil v spletno stran Kriptogram. Ker sem večinoma delal z matrikami, ki so v JavaScriptu dobro podprte, pri tem ne pričakujem večjih zapletov.

4.3 Izdelava slovarja

Glavna šibka točka substitucijske šifre je ohranjanje deleža črk v besedilu. Zato moramo v prvi fazi analizirati lastnosti slovenščine, kot so pogostost pojavljanja črk in besed, povprečje razdalj med samoglasniki, najpogostejše dvočrkovne besede,…

Na podlagi besedil moramo izdelati slovar oziroma korpus, ki bo služil kot naš reprezentativni vzorec jezika. Na začetku sem si za izdelavo korpusa izbral Cankarjevo delo Hlapci, ker je v celoti objavljeno na spletu. Kasneje se je izkazalo, da bi bilo bolje imeti kakovostnejši korpus, ki bi bil tudi bolj reprezentativen.

Zato sem uporabil nabor približno 10 milijonov besed, ki so jih zbrali za izdelavo javno dostopnega korpusa Kres - uravnoteženega podkorpusa Gigafide, najobsežnejšega slovenskega korpusa[21].



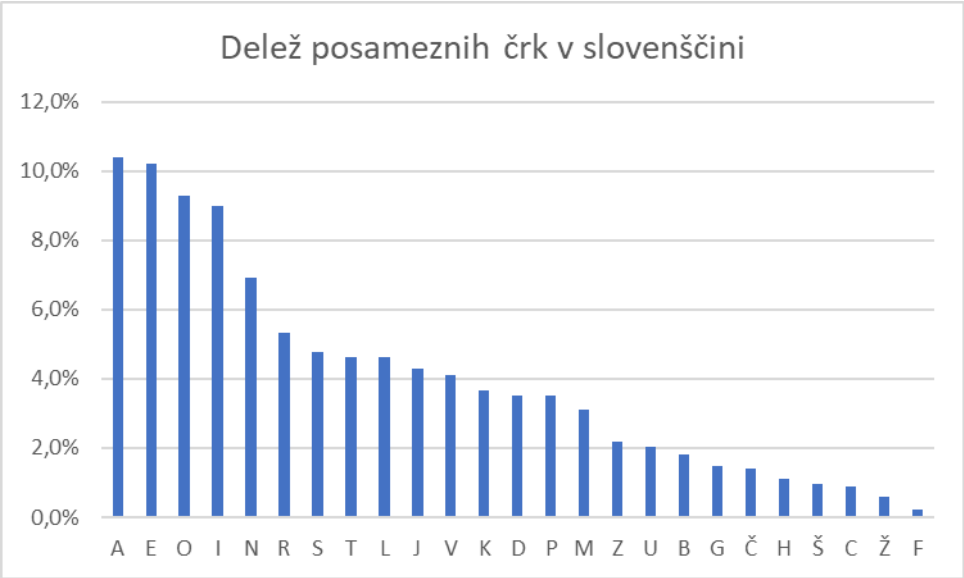
Slika 16: Delež zvrsti besedil, uporabljenih za izdelavo korpusov Gigafide (levo) in Kresa (desno)[17,18]

Na podlagi analize kvalitetnejše izdelanega korpusa, sem prišel do naslednjih ugotovitev:

Pogostost črk v slovenščini:

A	E	O	I	N	R	S	T	L	J	V	K	
10,4%	10,2%	9,27%	9,00%	6,91%	5,33%	4,77%	4,61%	4,61%	4,30%	4,12%	3,66%	
D	**P**	**M**	**Z**	**U**	**B**	**G**	**Č**	**H**	**Š**	**C**	**Ž**	**F**
3,50%	3,50%	3,11%	2,19%	2,02%	1,82%	1,49%	1,40%	1,10%	0,96%	0,89%	0,61%	0,24%

Slika 17: Delež posameznih črk v slovenščini, računano glede na lastni korpus



Slika 18: Grafični prikaz deleža črk v slovenščini, računano glede na lastni korpus

Deset najpogostejših parov črk v slovenščini:

JE	NA	NI	IN	EN	RA	ST	AN	PR	PO
1,69%	1,41%	1,28%	1,26%	1,25%	1,24%	1,22%	1,21%	1,19%	1,19%

Slika 19: Delež posameznih parov črk v slovenščini, računano glede na lastni korpus

Zanimivo je, da se rezultati precej razlikujejo od tistih, ki so jih pridobili raziskovalci z drugačnimi korpusi. Možna razlaga je dejstvo, da so različni ustvarjalci korpusov uporabili različne vire za nabor besed. To razlago podpira naslednja tabela, ki je izračunana glede na Gigafido:

celoten korpus		leposlovje		internet	
črka	frekvenca	črka	frekvenca	črka	frekvenca
a	10,13 %	a	10,83 %	a	10,52 %
e	9,8 %	e	10,71 %	e	9,94 %
o	9,07 %	i	8,85 %	o	9,15 %
i	8,78 %	o	8,83 %	i	8,66 %
n	6,75 %	n	6,32 %	n	6,78 %
r	5,34 %	l	5,35 %	r	5,26 %
s	4,59 %	s	5,1 %	s	4,63 %
t	4,53 %	r	4,87 %	t	4,58 %
l	4,44 %	j	4,55 %	l	4,37 %
v	4,15 %	t	4,22 %	v	4,16 %

Slika 20: Delež posameznih črk je v različnih zvrsteh besedil drugačen[19]

Zanimivo je, da so deleži črk, ki sem jih izračunal glede na svoj korpus, najbolj podobni tistim iz interneta. Glavni razlog za to je verjetno živost jezika, ki se s časom spreminja, ter velik vpliv interneta.

Daleč najpogostejši par črk v slovenščini je »JE«. Sledijo mu »NA«, »NI«, »EN«,… Vrednosti sem vnesel v matriko tako, da so za par znakov »z_1z_2« shranjene na mesto $a_{i,j}$, kjer sta *i* in *j* zaporedni števili znakov z_1 in z_2 po slovenski abecedi, če bi začeli šteti z A = 0. Par »DA« je torej na primer shranjen na mestu $a_{4,0}$, »JE« na mestu $a_{10,5}$, »EN« pa na mestu $a_{5,14}$.

V poglavju 4.1 so elementi matrike označeni z indeksi, ki so naravna števila. V programiranju navadno uporabljamo tudi ničlo. Črke slovenske abecede zato oštevilčimo tako, da črki A pripada število 0 in tako naprej. To pomeni, da so indeksi elementov matrike lahko tudi 0.

4.4 Iskanje samoglasnikov

Samoglasniki se od soglasnikov razlikujejo po tem, da jih lahko izgovarjamo brez ostalih glasov, zato so za govorjenje nujno potrebni. Kot sem že omenil v poglavju 3.2, je število soglasnikov med vsakim posameznim samoglasnikom v besedilu razmeroma majhno in se giblje okoli 1,5 (če pri tem ne štejemo presledkov).

To dejstvo bomo uporabili, da ugotovimo, katere 4 črke predstavljajo samoglasnike A, E, I, O. Njihov delež se v besedilih giblje med 7-12%, medtem ko je U bližje 1-2%, zato niti ni tako pomemben kot ostali samoglasniki.

Ker se morajo vsi štirje samoglasniki A, E, I in O skoraj gotovo nahajati med prvih 6-7 najpogostejših črk v tajnopisu, bomo skušali najti takšno kombinacijo 4 črk, pri kateri so vse med njimi zašifrirani samoglasniki. Da bomo res prepričani, da smo zajeli vse samoglasnike, jih bomo iskali med prvimi 9 najpogostejšimi črkami v tajnopisu. Tako bomo preverili $\binom{9}{4} = 126$ kombinacij. Za vsako izmed njih bomo izračunali matriko *M,* ki nam bo povedala povprečno dolžino razdalj med dvema črkama v tajnopisu. To strogo opišemo po naslednjem postopku (računalniški program je priložen v prilogi):

1. Naj bo *A* kombinacija 4 črk, za katere hočemo izračunati matriko razdalj, matriki *B* in *C* ničelni matriki velikosti 4×4, i in j pa sta zaporedni števili črk Z_0 in Z_1 v *A*, če začnemo šteti z 0 (če imamo kombinacijo *A* = (»A«, »E«, »I«, »O«) in če je »A« = Z_0 in »E« = Z_1, potem je *i* = 0 in *j* = 1)
2. Za vsako črko v besedilu, ki se nahaja v *A (razen zadnje)*, po vrsti preštejemo število črk, ki ji sledijo in niso v *A*, dokler ne pridemo do črke, ki je v *A* . Presledkov in črk v *A* ne štejemo. Naj bo črka, pri kateri smo začeli šteti Z_1 in črka, pri kateri smo končali Z_2 (Z_0 in Z_1 sta elementa A). Dobljeno število prištejemo v polje $b_{i,j}$ matrike *B* in v matriki *C* polju $c_{i,j}$ prištejemo 1
3. *M* pridobimo tako, da polju $m_{i,j}$ za $0 \leq i,j \leq 3$ priredimo vrednost $\frac{b_{i,j}}{c_{i,j}}$, s čimer izračunamo aritmetično sredino posameznih dolžin razdalj med samoglasniki.

Da je postopek bolj razumljiv, si ga oglejmo na spodnjem primeru:

Recimo, da hočemo narediti matriko razdalj za kombinacijo črk *A* = (»A«, »E«, »I«, »O«), glede na besedilo »MED PRVIMI JE ZAMISEL«.

Slika 21: Primer besedila, glede na katerega bomo pridobili matriko razdalj

Najprej zanemarimo presledke in jih za lažjo predstavo označimo z oranžno:

Slika 22: Presledke v besedilu zanemarimo in jih označimo oranžno

Nato črke iz kombinacije *A* podčrtajmo modro:

Slika 23: Črke, ki se nahajajo v A, označimo z modro

Zdaj moramo za vsako črko kombinacije *A* po vrsti prešteti črke, dokler ne pridemo do naslednje črke iz kombinacije *A*. Začnemo pri E-ju v besedi »MED«. Sledijo mu črke D, P, R, V. Nehamo šteti, saj smo prišli do I-ja, ki se nahaja v *A*.

Slika 24: Prikaz računanja razdalje

Ker so med E-jem in I-jem štiri črke, ima ta razdalja vrednost 4. Velja Z_0 = »E« in Z_1 = »I«, posledično pa *i* = 1 in *j* = 2. Zato polju $b_{1,2}$ matrike B prištejemo 4, $c_{1,2}$ pa 1.

Naslednjo razdaljo začnemo šteti tam, kjer smo nehali šteti prejšnjo (v tem primeru pri prvem I-ju v besedi »PRVIMI«).

Ta postopek ponavljamo, dokler ne pridemo do zadnje črke v *A*.

Slika 25: Ko pridemo do zadnje črke iz A, je postopek končan. Rezultate označimo v matriko

Nato vsakemu polju $m_{i,j}$ matrike M za $0 \leq i, j \leq 3$ priredimo vrednost $\frac{b_{i,j}}{c_{i,j}}$. Tako bi dobili matriko:

$$M = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Matrika *M* nam pove, kakšna je povprečna razdalja med posameznimi samoglasniki v besedilu. V tem primeru se je izšlo, da so vsi elementi *M* naravna števila, pri večjih besedilih pa pridejo racionalna. *M* skoraj nikoli ni simetrična, saj razdalje vedno štejemo od leve proti desni.

Spodaj je navedena matrika razdalj M za kombinacijo A = (»A«, »E«, »I«, »O«), pridobljeno glede na Hlapce (vrednosti so zaradi preglednosti zaokrožene na 3 decimalna mesta):

$$M = \begin{bmatrix} 1{,}720 & 1{,}628 & 1{,}480 & 1{,}574 \\ 1{,}599 & 1{,}614 & 1{,}445 & 1{,}478 \\ 1{,}572 & 1{,}618 & 1{,}687 & 1{,}520 \\ 1{,}630 & 1{,}542 & 1{,}532 & 1{,}589 \end{bmatrix}$$

Slika 26: Matrika razdalj med samoglasniki AEIO, izračunano glede na Hlapce

Ko imamo definiran postopek, izračunamo matriko razdalj *M* za vseh 126 možnih kombinacij črk, med katerimi bi vse štiri lahko bile zašifrirani samoglasniki. Pomislimo, kaj bi se zgodilo z vrednostmi matrike, če ena ali več črk v kombinaciji *A* ne bi bila samoglasnik.

Vsota elementov matrike *M* bi bila približno enaka, saj je ta odvisna le od deleža črk v kombinaciji *A* – štejemo namreč črke, ki niso v njej in jih nato delimo s številom vseh črk.

Zato pomislimo drugače. Omenili smo že, da je število soglasnikov med samoglasniki razmeroma konstantno in se večinoma giblje med 1 in 3. Čim pa v kombinaciji *A* ne bi bile vse štiri črke samoglasniki, bi se pojavila razmeroma dolga zaporedja črk, med katerimi ne bi bilo nobene črke iz *A* (kot je lepo vidno na sliki 7). Zato moramo vrednotiti razpršenost elementov matrike *M*, kar storimo s standardnim odklonom. Manjši kot bo standardni odklon, večja je verjetnost, da so vse štiri črke kombinacije *A* zašifrirani samoglasniki. Za matriko na sliki 24, je $\sigma = 0{,}0728$, vrednost matrike prave kombinacije pa bi ji morala biti podobna.

Izkaže se, da se standardni odklon zelo dobro obnese, tudi pri kratkih tajnopisih. Pri nekaterih je bilo dovolj že 50 znakov, da sem lahko našel pravilno kombinacijo črk, pri skoraj vseh pa je zadostovalo 100. Tako sem izmed 126 kombinacij uspel najti učinkovito najti tisto, za katero vem, da so vse črke v njej samoglasniki. Naslednji korak je bil ugotoviti, za katere samoglasnike gre.

4.5 Določanje samoglasnikov

Do zdaj nam je uspelo ugotoviti, katere 4 črke so samoglasniki, vendar ne vemo točno, kateri izmed njih so A, E, I in O. Vseh možnih permutacij 4 črk je 4! = 24, zato sem poskusil oceniti vsako izmed njih in ugotoviti, katera je najverjetneje pravilna. Permutacija črk P = (»Z_0«, »Z_1«, »Z_2«, »Z_3«) bi bila pravilna natanko tedaj, ko bi preslikava čistopisa v tajnopis vsebovala preslikave črk »A« → »Z_0«, »E« → »Z_1«, »O« → »Z_2« in »I« → »Z_3«. Takšen pogoj sem določil, ker si črke A, E, I in O sledijo po pogostosti v istem zaporedju (slike 17, 18, 20), kar nekoliko olajša interpretacijo končnih rezultatov.

Na začetku je vse kazalo, da se črki A in E po lastnostih precej razlikujeta od I in O. Primerjal sem jih po različnih kriterijih:

1. Najprej sem za vsako izmed 24 permutacij P_0, P_1,…, P_{23} priredil matrike deležev parov $M_{d(0)}$, $M_{d(1)}$, … , $M_{d(23)}$ po naslednjem postopku:
   -Matrike M_d naj bodo matrike velikosti 4×4, elemente permutacije P_n označimo s (»$Z_{n,0}$«, »$Z_{n,1}$«, »$Z_{n,2}$«, »$Z_{n,3}$«).
   -Z $d_{n(a,b)}$ označimo delež parov črk »$Z_{n,a}Z_{n,b}$« v tajnopisu z odstranjenimi presledki (za $0 \leq a, b \leq 3$)..
   -Poljem $m_{a,b}$ matrike $M_{d(n)}$ za $0 \leq a, b \leq 3$ priredimo vrednosti:

$$m_{a,b} = d_{a,b}$$

To pomeni, da bomo permutaciji P_0 priredili matriko M_{d0}, katere elementi predstavljajo deležev posameznih parov črk. Poglejmo si vizualno:

$$M_{d(0)} = \begin{bmatrix} d_{0(0,0)} & d_{0(0,1)} & d_{0(0,2)} & d_{0(0,3)} \\ d_{0(1,0)} & d_{0(1,1)} & d_{0(1,2)} & d_{0(1,3)} \\ d_{0(2,0)} & d_{0(2,1)} & d_{0(2,2)} & d_{0(2,3)} \\ d_{0(3,0)} & d_{0(3,1)} & d_{0(3,2)} & d_{0(3,3)} \end{bmatrix} \begin{matrix} Z_{0,0} \\ Z_{0,1} \\ Z_{0,2} \\ Z_{0,3} \end{matrix}$$

$$Z_{0,0} \quad Z_{0,1} \quad Z_{0,2} \quad Z_{0,3}$$

Če bi hoteli izvedeti delež para črk »$Z_{0,2}Z_{0,3}$«, bi podatek našli v presečišču četrte vrstice in tretjega stolpca matrike $M_{d(0)}$. Če bi hoteli izvedeti delež para črk »$Z_{0,3}Z_{0,2}$«, pa bi pogledali v presečišče tretje vrstice in četrtega stolpca iste matrike.

Ko sem takšne matrike oblikoval za vse permutacije P_0, P_1,…, P_{23}, sem jih lahko z metričnimi normami primerjal z matriko deležev parov $M_{d(samoglasniki)}$, ki sem jo izračunal s kombinacijo (»A«, »E«, »O«, »I«) glede na slovar po zgoraj navedenem postopku. Enako kot pri poglavju 5.4 bi bilo za pričakovati, da manjša kot je razdalja med matrikama $M_{d(samoglasniki)}$ in $M_{d(n)}$, večja je verjetnost, da je permutacija P_n pravilna.

Tako sem matrike $M_{d(0)}$, $M_{d(1)}$, … , $M_{d(23)}$ primerjal z $M_{d(samoglasniki)}$ z evklidsko in Manhattanovo razdaljo. Rezultati so bili slabši, kot sem pričakoval. Normi sta se izkazali za praktično enako neučinkoviti, saj sta število kandidatskih permutacij zožili le na približno 12. Resda sem se zavedal, da je samoglasniških parov malo, vendar sem se zanašal na dejstvo, da presledkov pri tej metodi ne upoštevamo in bi se morda lahko nekateri pari črk vseeno večkrat pojavljali. Ker je bila ta metoda neuspešna, sem poskusil z drugo.

```
207 def evklidska_razdalja(matrika1, matrika2):
208     v = 0
209     for i in range(len(matrika1)):
210         for j in range(len(matrika2)):
211             v+=(matrika1[i][j]-matrika2[i][j])**2
212             return(v)
```

Slika 27: Definicija evklidske razdalje v programu Python

```
220 def manhatanova_razdalja(matrika1, matrika2):
221     v = 0
222     for i in range(len(matrika1)):
223         for j in range(len(matrika2)):
224             v+=abs(matrika1[i][j]-matrika2[i][j])
225             return(v)
```

Slika 28: Definicija Manhattanove razdalje v programu Python

2. Druga ideja je do neke mere podobna prvi. Izhaja iz dejstva, da so nekateri pari črk v slovenščini bolj podobni od drugih (tabela se nahaja v poglavju 4.3). Za *tajnopis* in *čistopis* sem ustvaril matriki $M_{d(tajnopis)}$ in $M_{d(slovar)}$velikosti *25 x 25* po naslednjem postopku:

$$M_d = \begin{bmatrix} d_{0,0} & \cdots & d_{0,24} \\ \vdots & \ddots & \vdots \\ d_{24,0} & \cdots & d_{24,24} \end{bmatrix} \begin{matrix} Z_0 \\ \vdots \\ Z_{24} \end{matrix}$$
$$Z_0 \quad \cdots \quad Z_{24}$$

Znaki Z_0,…, Z_{24} tokrat predstavljajo slovenske črke po abecednem redu, $d_{slovar(a,b)}$ delež parov »Z_aZ_b« *v slovarju*, $d_{tajnopis(a,b)}$ pa delež parov »Z_aZ_b« *v tajnopisu* (za $0 \leq a, b \leq 24$). Polja $M_{d(tajnopis)}$ izračunamo po predpisu:

$$m_{a,b} = d_{tajnopis(a,b)},$$

polja $M_{d(slovar)}$ pa po:

$$m_{a,b} = d_{slovar(a,b)}.$$

Naj bodo vektorji $V_{tajnopis(0)}, V_{tajnopis(1)},\ldots, V_{tajnopis(24)}$ zaporedne vrstice matrike $M_{d(tajnopis)}$, vektorji $V_{slovar(0)}, V_{slovar(1)},\ldots, V_{slovar(24)}$ pa zaporedne vrstice matrike $M_{d(slovar)}$. Naj $\sigma(V)$ predstavlja standardni odklon elementov vektorja V.

Ker je pogostost posameznih parov v besedilu različna, bi bilo smiselno pričakovati, da bodo standardni odkloni elementov vektorjev $V_{slovar(0)}, V_{slovar(1)},\ldots, V_{slovar(24)}$ med seboj različni. To je najlažje razložiti tako, da si predstavljamo $V_{slovar(10)}$ kot vektor, ki po vrsti vsebuje podatke o pogostosti parov »JA«, »JB«,…,»JE«,…, »JŽ« v slovenščini. Ker pa je »JE« najpogostejši par črk v slovenščini, lahko sklepamo, da bodo imeli elementi vektorja $V_{čistopis(10)}$ razmeroma visok standardni odklon.

Ta predpostavka se je izkazala za resnično. Za lažje razumevanje recimo, naj bo standardni odklon črke »Z_n« enak $\sigma(V_{slovar(n)})$, standardni odklon črke »Z_n« v tajnopisu pa enak $\sigma(V_{tajnopis(n)})$. Črke, ki se v slovenščini pogosteje pojavljajo v najpogostejših parih, imajo višji standardni odklon od tistih, ki se redkeje. Spodnja tabela prikazuje standardne odklone črk v promilih, ki sem jih izračunal glede na matriko $M_{d(slovar)}$:

A	E	O	I	N	R	S	T	L	J	V	K	
3,8 ‰	3,4 ‰	2,7 ‰	3,2 ‰	4,0 ‰	2,8 ‰	3,3 ‰	2,6 ‰	3,0 ‰	3,9 ‰	1,9 ‰	3,0 ‰	
D	**P**	**M**	**Z**	**U**	**B**	**G**	**Č**	**H**	**Š**	**C**	**Ž**	**F**
2,5 ‰	2,9 ‰	2,3 ‰	1,5 ‰	0,7 ‰	1,7 ‰	1,2 ‰	0,9 ‰	0,3 ‰	0,7 ‰	0,5 ‰	0,6 ‰	0,1 ‰

Iz tabele lahko razberemo kar nekaj zanimivih dejstev. Poleg črke J imajo izrazito visok standardni odklon predvsem črke N, A in E. Prav tako je lepo vidna korelacija med pogostostjo črk in njihovimi standardnimi odkloni.

Zanimivo pa je še neko dejstvo. Vidimo, da imata samoglasnika A in E precej večja standardna odklona od O in I.

Iz poglavja 4.4 vemo, katere črke v tajnopisu so v resnici zašifrirani samoglasniki. Zato sem vsaki izmed njih izračunal standardni odklon v tajnopisu. Zaradi velike razlike med samoglasnikoma A in E ter samoglasnikoma O in I, sem predpostavil, da lahko 4 kandidate ločim na dve skupini: na kandidata za A in E ter na kandidata za O in I.

Tako sem izmed 24 možnih permutacij prešel na 4: AEIO, AEOI, EAIO in EAOI. Torej se je 2. metoda obnesla veliko bolje od prve. Žal sem ugotovil, da število možnosti težko reduciram na manj kot 4, saj imata lahko črki A in E v besedilu zelo podobne ali zelo različne lastnosti. Podobno velja tudi za črki O in I.

Problem metode je sledeč: včasih se zgodi, da sta zašifrirani črki I in O v besedilu mnogo pogostejši, kot predvideva povprečje. V tem primeru imata večji standardni odklon in posledično program narobe izbere kandidate.

Rezultat natančnega določanja samoglasnikov z uporabo statistike je bil slabši, kot bi si želel. Ne glede na to, kako sem primerjal podatke, vedno so se našli primeri tajnopisov, ki so izstopali iz povprečja in ovirali delovanje programa.

4.6 Določanje črk N, R, S, T z uporabo grobe sile

Ker se je uporaba statističnih metod izkazala za nezanesljivo, sem poskusil substitucijsko šifro razbiti z grobo silo (angleško »brute-force«). To pomeni, da hočemo s preizkušanjem vseh možnosti uganiti pravo. Naš cilj je bil določiti prvih 8 najpogostejših črk v slovenščini: A, E, O, I, N, R, S in T, ki skupaj predstavljajo okoli 60% vseh slovenskih črk. Če bi za nek tajnopis točno vedeli, katere črke so zašifrirani A, E, O, I, N, R, S, T, bi ga zelo hitro razbili.

Vemo, da je možnih permutacij samoglasnikov 4! = 24. Poglejmo, koliko je permutacij črk v tajnopisu, ki bi lahko bile zašifrirane črke N, R, S, T. S preučevanjem kratkih besedil sem ugotovil, da lahko črke, ki niso samoglasniki, močno odstopajo od njihovega pričakovanega mesta po pogostosti. T, ki je 8. najpogostejša črka v slovenščini, bi se lahko nahajal na primer na 13. mestu po pogostosti. S poskušanjem sem ugotovil, da se skoraj vedno vse nahajajo med

15 najpogostejših črk v tajnopisu. To pomeni, da jih v resnici iščemo med 11 kandidati, saj za 4 od 15 vemo, da so samoglasniki, ki ne morejo biti N, R, S ali T.

Najprej sem iz pomožnega seznama najpogostejših črk v tajnopisu odstranil zašifrirane črke A, E, O, I. Nato sem izmed 11 naslednjih po istem seznamu izpisal vse variacije črk dolžine 4, ki jih je $\frac{11!}{(11-4)!} = 7920$. Za lažje razumevanje naslednjega koraka, bom označil variacije kandidatov za črke N, R, S, T z $V_1, V_2, \dots, V_{7920}$, možne permutacije kandidatov za A, E, O, I pa $P_1, P_2, \dots, P_{24}$.

```
435 najpogostejsih_11_preostalih_znakov_v_sifri=znaki_v_besedilu_po_pogostosti[:11]
436
437 permutacije = []
438 p = []
439
440 p = list(itertools.combinations(najpogostejsih_11_preostalih_znakov_v_sifri, 4))
441 pomozni_p = []
442 for i in p:
443     pomozni_p += itertools.permutations(i)
444
445 for moznost in najverjetnejsi_samoglasniki:
446     for j in pomozni_p:
447         permutacije.append(moznost + j)
```

Slika 29: Prirejanje osmeric, ki jih bomo pregledali z uporabo grobe sile

Iz vseh možnih permutacij kandidatov za A, E, O, I in variacij kandidatov za N, R, S, T sem ustvaril osmerice možnih kandidatov za A, E, O, I, N, R, S, T tako, da sem elementom množic $P_1, P_2, \dots, P_{24}$ prištel elemente $V_1, V_2, \dots, V_{7920}$, pri čemer nisem spremenil njihovega vrstnega reda. Vseh možnih osmeric je torej $24 \cdot 7920 = 190.080$. Nato sem poiskal najverjetnejše osmerice O_n po naslednjem postopku:

Naj bodo $o_{n1}, o_{n2}, \dots, o_{n8}$ elementi osmerice O_n, pri čemer je $1 \leq n \leq 190080$. Najprej iz tajnopisa odstranimo vse znake, ki niso črke, razen presledkov. Nato iz tajnopisa naredimo seznam besed, dolgih vsaj 8 črk, ki ga imenujemo S_n. Tako zmanjšamo število elementov, ki jih bo program moral pregledovati, kar močno poveča njegovo učinkovitost. Nazadnje ustvarjeni seznam S_n preslikamo tako, da izvedemo preslikave črk $o_{n1} \to A, o_{n2} \to E, o_{n3} \to O, o_{n4} \to I, o_{n5} \to N, o_{n6} \to R, o_{n7} \to S$ ter $o_{n8} \to T$. Vse ostale črke besed v seznamu zamenjamo z znakom $*$.

```
41 def zamenjaj_znake(tekst, znaki_za_zamenjati, znaki_za_nadomestiti):
42     for i in znaki_za_zamenjati:
43         tekst = tekst.replace(i, str(znaki_za_zamenjati.index(i)))
44     for i in range(len(znaki_za_nadomestiti)):
45         tekst = tekst.replace(str(i), znaki_za_nadomestiti[i])
46     for i in se_eni_pomozni_znaki_v_besedilu:
47         if i not in znaki_za_nadomestiti:
48             tekst = tekst.replace(i, "*")
49     return(tekst)
```

Slika 30: Definicija za "substitucijo" znakov

Nato iz slovarja poiščemo vse besede, ki vsebujejo vsaj 8 črk in iz njih ustvarimo seznam S_{slovar}. Na enak način kot v prejšnjem odstavku izvedemo preslikave besed seznama S_{slovar}. Ker vse besede seznamov S_n in S_{slovar}, vsebujejo le črke A, E, O, I, N, R, S, T ali znak *, jih imenujmo *substituirane besede*.

Program je smiselnost osmeric ocenjeval tako, da je za vsako ujemanje substituiranih besed iz seznamov S_{slovar} in S_n ničelnemu vektorju V dolžine 190.080 na n-to mesto prištel kvadrat dolžine ujemajoče substituirane besede. Če bi se v seznamu S_{slovar} na primer nahajala substituirana beseda »*or*i*s*i«, hkrati pa bi se pojavila tudi v seznamu S_1, bi prvemu mestu ničelnega vektorja V prištel $9^2 = 81$.

```
497 for i in permutacije: # osmerice
498     pomozna_sifra = zamenjaj_znake(sifra2, i, aeoinrst)
499     for i in besede_za_pregledovanje: # 8 ali več črkovne substituirane besede iz korpusa
500         if i in pomozna_sifra: # tajnopis
501             v_poskus += len(i)**2
502     matrika_v.append(v_poskus)
503     v_poskus=0
```

Slika 31: Program, ki preveri vse osmerice

Kmalu se je izkazalo, da slovar, ki sem ga ustvaril iz Hlapcev, ne bo uporaben, saj je vseboval manj kot 1000 besed z osmimi ali več črkami. Zato sem ustvaril veliko večji slovar, ki sem ga opisal v poglavju 4.3. Z njim sem lahko ustvaril več kot 77000 različnih substituiranih besed, kar je bilo dovolj, da so se v vektorju V začele kazati razlike.

4.6.1 Rezultati

Analiza korpusa se je izkazala za uporabno, saj sem lahko ustvaril natančne matrike razdalj in pogostosti parov črk. Poleg tega je pripomogla k razumevanju problema, saj se je večkrat zgodilo, da je novo odkritje o lastnostih matrik spodbudilo napredek na ostalih področjih.

Metrične norme niso dale pričakovanih rezultatov. Mislil sem, da bi morali biti evklidska in Manhattanova razdalja natančna kriterija, za primerjavo matrik, vendar sem se motil. Ko sem raziskoval, zakaj so se obnesle slabo, sem ugotovil, da je bilo veliko matrik simetričnih. Zaradi simetričnosti metričnega prostora, ki sem jo navedel v poglavju 3.2, je bilo veliko vrednosti evklidskih in Manhattanovih razdalj med seboj parno enakih, kar je dodatno otežilo delo.

Zelo dobre rezultate je prinesla uporaba standardnega odklona. Nikjer v literaturi nisem zasledil pristopa z uporabo standardnega odklona, zato ta ideja predstavlja najbolj izviren del naloge. Najprej sem s standardnim odklonom našel način, kako se hitro in zanesljivo najde črke, ki so v poljubnem tajnopisu zašifrirani samoglasniki. Program je bil tako natančen, da je že pri tajnopisih dolžine med 50 in 100 črk lahko konsistentno pravilno določil kombinacije črk, ki so bile samoglasniki A, E, I, O.

Zaradi presenetljive učinkovitosti standardnega odklona sem ga preizkušal na raznolike načine. Med njimi se je razmeroma dobro izkazalo računanje standardnega odklona posameznih vrstic matrik pogostosti parov. Tako smo ugotovili, da imajo nekatere črke višji

standardni odklon kot druge, kar sem natančno prikazal v poglavju 4.5. Pri dovolj dolgih besedilih bi se dalo črke najti že na podlagi te metode.

Vendar je bil naš cilj zahtevnejši; hoteli smo napisati program, ki bi deloval tudi pri krajših tajnopisih. Pri teh samo statistično primerjanje matrik ni bilo dovolj.

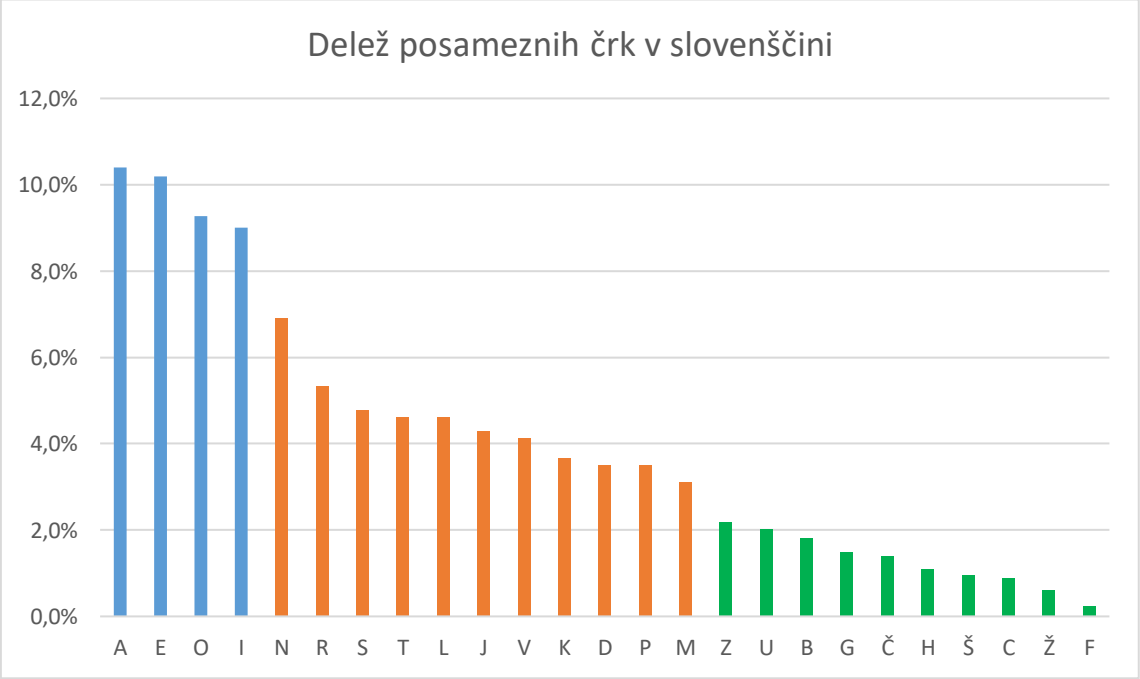
Zato sem se odločil poizkusiti z uporabo grobe sile. Rezultati so bili takšni, kot sem predvideval; izkazala se je za uporabno in natančno pri iskanju osmerice, ki pravilno prikazuje, kako je zašifriranih osem najpogostejših črk v slovenskem jeziku. Njena edina pomanjkljivost je bilo dolgo izvajanje programa zaradi izvrševanja velikega števila operacij.

5. Zaključek

Raziskovalna naloga, ki sem si jo zastavil, je zanimiva predvsem zaradi števila različnih načinov, s katerimi bi lahko prišli do rešitve, zato še zdaleč ni zaključena; predstavlja mi izziv, saj sem dobil občutek, da sem se z vsako novo idejo, ki mi jo je uspelo spisati v program, naučil nekaj povsem novega. Spoznal sem mnoge vidike kriptografije, sočasno pa izboljšal svoje znanje programiranja in matematike.

Zato bom raziskoval, dokler ne bom s svojo rešitvijo popolnoma zadovoljen. Takrat bom program prevedel v JavaScript in ga objavil na spletni strani Kriptogram.

Trenutno se mi najbolj zanimiva zdi ideja o združevanju črk slovenske abecede v tri skupine glede na pogostost pojavljanja. Vidimo, da so med mejami posameznih skupin veliki preskoki. Črke v besedah bi lahko pobarvali, tako kot so na sliki 32 in primerjali barvne vzorce.



Slika 32: Črke po pogostosti, razporejene v tri skupine

6. Viri

[1] *Šifra*. [internet]. [citirano 24. 2. 2020]. Dostopno na naslovu: https://sl.wikipedia.org/wiki/%C5%A0ifra

[2] *Kriptografija.* [internet]. [citirano 24. 2. 2020]. Dostopno na naslovu: https://sl.wikipedia.org/wiki/Kriptografija

[3] A. Dimovski, D. Gligorski. *Attacks on the Transposition Ciphers Using Optimization Heuristics.* 2003- [internet]. [citirano 23. 2. 2020]. Dostopno na naslovu: https://pdfs.semanticscholar.org/3866/77ae1b8ea4cd086c419a1ba0306f6971f88b.pdf

[4] Kodrič, S. *Orodja za razbijanje substitucijske šifre*. 2013. [internet]. [citirano 24. 2. 2020]. Dostopno na naslovu: http://eprints.fri.uni-lj.si/1976/1/Kodri%C4%8D_S-1.pdf

[5] Jackobsen, T. *A Fast Method for the Cryptanalysis of Substitution Ciphers*. 1995. [internet]. [citirano 23. 2. 2020]. Dostopno na naslovu: https://www.researchgate.net/profile/Thomas_Jakobsen4/publication/266714630_A_fast_method_for_cryptanalysis_of_substitution_ciphers/links/56ebe4fe08aefd0fc1c718ef.pdf

[6] *Vigenere Cipher.* [internet]. [citirano 24. 2. 2020]. Dostopno na naslovu: https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher

[7] Crypto Museum. *Codebooks.* [internet]. [citirano 24. 2. 2020]. Dostopno na naslovu: https://www.cryptomuseum.com/crypto/codebook/index.htm

[8] Mohorčič, A., Pustavrh. S, idr. *Vega 1: i-učbenik za matematiko v 1. letniku gimnazije*. 2014. [internet]. [citirano 24. 2. 2020]. Dostopno na naslovu: https://eucbeniki.sio.si/vega1/3306/index.html

[9] Pavletič, M. *Kombinatorika.* [internet]. [citirano 24. 2. 2020]. Dostopno na naslovu: http://www2.arnes.si/~mpavle1/mp/kombi.html

[10] Logar, N., Erjavec, T., Krek, S., Grčar, M., in Holozan, P. *Written corpus ccKres 1.0, Slovenian language resource repository CLARIN.SI*. 2013. [internet]. [citirano 24. 2. 2020]. Dostopno na naslovu: http://hdl.handle.net/11356/1034

[11] Jurišić, A. *Tečaj iz kriptografije in teorije kodiranja*. 2009. [internet]. [citirano 24. 2. 2020]. Dostopno na naslovu: http://lkrv.fri.uni-lj.si/~ajurisic/kitk2-09/folije/p02.pdf

[12] Ključevšek, A. *Statistična analiza slovenskih jezikovnih korpusov*. 2016. [internet]. [citirano 24. 2. 2020]. Dostopno na naslovu: http://eprints.fri.uni-lj.si/3570/1/63040071-ALEKSANDER_KLJU%C4%8CEV%C5%A0EK-Statisti%C4%8Dna_analiza_slovenskih_jezikovnih_korpusov.pdf

[20] Kriptogram. [internet]. [citirano 3. 3. 2020]. Dostopno na naslovu: http://lkrv.fri.uni-lj.si/crypto-portal/

[21] Gigafida. [internet]. [citirano 3. 3. 2020]. Dostopno na naslovu: http://www.gigafida.net/Support/About

6.1 Slikovno gradivo

[13] *Transposition Cipher*. [internet.] [citirano 24. 2. 2020]. Dostopno na naslovu: https://www.wattpad.com/532975681-codes-ciphers-transposition-cipher

[14] *Pogostost pojavljanja črk v besedilu*. [internet.] [citirano 24. 2. 2020]. Dostopno na naslovu: https://upload.wikimedia.org/wikipedia/sl/timeline/0767e6e3fba035c565d2958d65927d2f.png

[15] *Cezarjeva šifra*. [internet.] [citirano 24. 2. 2020]. Dostopno na naslovu: https://media.geeksforgeeks.org/wp-content/uploads/ceaserCipher.png

[16] Reinhold, A. *Page 187 of State Department 1899 Code Book*. 2019. [internet.] [citirano 24. 2. 2020]. Dostopno na naslovu: https://en.wikipedia.org/wiki/Codebook#/media/File:State_Department_code_book_1899,_code_page_187.agr.jpg

[17] *Zvrsti besedil v korpusu Gigafida*. [internet.] [citirano 29. 2. 2020]. Dostopno na naslovu: http://www.gigafida.net/Content/Images/About/zvrsti.png

[18] *Zvrsti besedil v korpusu Kres*. [internet.] [citirano 29. 2. 2020]. Dostopno na naslovu: http://www.korpus-kres.net/Content/Images/About/zvrsti.png

[19] Ključevšek, A. *Najpogostejše črke v korpusu Gigafida*. 2016. [internet]. [citirano 24. 2. 2020]. Dostopno na naslovu: http://eprints.fri.uni-lj.si/3570/1/63040071-ALEKSANDER_KLJU%C4%8CEV%C5%A0EK-Statisti%C4%8Dna_analiza_slovenskih_jezikovnih_korpusov.pdf

7. Kazalo slik

8. Priloge

Definicija računanja matrike razdalj med samoglasniki:

```
def najkrajsa_razdalja(potencialni_samoglasniki, mnozica, dolzina_stranice_matrike):
    pomozna_matrika = [[0]*dolzina_stranice_matrike for i in range(dolzina_stranice_matrike)]
    matrika_razdalj = [[0]*dolzina_stranice_matrike for i in range(dolzina_stranice_matrike)]
    rzd = 0
    mesto = 0
    for i in mnozica:
        if mnozica[mesto] not in potencialni_samoglasniki:
            mesto+=1
        else:
            s1 = mnozica[mesto]
            break

    while mesto < len(mnozica):
      while ((mesto < len(mnozica) and mnozica[mesto] not in potencialni_samoglasniki)):
          mesto+=1
          rzd +=1
      if mesto < len(mnozica):
          s2 = mnozica[mesto]
          matrika_razdalj[potencialni_samoglasniki.index(s1)][potencialni_samoglasniki.index(s2)]+=rzd
          pomozna_matrika[potencialni_samoglasniki.index(s1)][potencialni_samoglasniki.index(s2)]+=1
          rzd=0
          s1 = s2
      mesto+=1
    v = 0
    for i in range(dolzina_stranice_matrike):
        for j in range(dolzina_stranice_matrike):
            if pomozna_matrika[i][j] != 0:
                matrika_razdalj[i][j]=matrika_razdalj[i][j]/pomozna_matrika[i][j]
                v += matrika_razdalj[i][j]
            else:
                matrika_razdalj[i][j]=0
    return(matrika_razdalj)

```

Slika 33: Definicija za računanje matrike razdalj med samoglasniki

Definicija računanja matrike deleža parov v besedilu:

```
def pridobi_matriko_frekvenc_parov(mnozica_crk, matrika, seznam_znakov, dolzina_stranice_matrike):
    m = [[0]*dolzina_stranice_matrike for i in range(dolzina_stranice_matrike)]
    for i in range(dolzina_stranice_matrike):
        for j in range(dolzina_stranice_matrike):
            m[i][j]=matrika[seznam_znakov.index(mnozica_crk[i])][seznam_znakov.index(mnozica_crk[j])]
    return(m)
```

Slika 34: Definicija za računanje matrike deleža parov v besedilu

Opomba: definicija je bila za vsak slučaj napisana zelo splošno. Argument funkcije seznam_znakov določa, katero abecedo znakov uporabimo za izdelavo matrike. Na koncu se je izkazalo, da je bilo najbolje uporabljati le slovensko abecedo

Definicija štetja frekvence elementov v vektorju:

```
def pregled(i, mnozica):
    if(i in mnozica):
        return(int(mnozica.count(i)))
    else:
        return(0)
```

Slika 35: Preprosta definicija, ki se je izkazala za zelo praktično