Лабораторна робота 1 з Симетричної Криптографії

Команда: Бондар, Дигас Група: ФІ-03

Підготовка даних

1. В якості середньостатистичного тексту російською мовою ми взяли дописи з телеграм-каналу терориста і військового злочинця ігоря гіркіна
2. Повний вхідний текст можна знайти у відповідному файлі
3. Обробка тексту і підготовка до аналізу відбувається у розділі "Text reading and preprocessing"

Програмна частина

Text reading and preprocessing

In [ ]:
```
filename = "girkin_crying.txt"
```

In [ ]:
```
def get_text(_filename):
    f = open(_filename, "r", encoding='utf-8')
    text = f.read()
    f.close()
    return text

def transform_symbol(_c):
    if 'а' <= _c and _c <= 'я':
        return _c
    elif _c <= 'Я' and _c >= 'А':
        return _c.lower()
    elif _c == 'Ё' or _c == 'ё':
        return 'е'
    else:
        return ' '

def preprocess_text(_text):
    _text = get_text(filename)
    text_formatted = ""
    # Change symbols according to requirements
    for c in _text:
        text_formatted += transform_symbol(c)

    # Remove consequtive spaces
    text_formatted = ' '.join(text_formatted.split())
    return text_formatted
```

In [ ]:
```
text = preprocess_text(get_text(filename))
```

Text processing (singular char count and bigram count)

In [ ]:
```
def count_chars(_text):
    c_count = {}
    for c in _text:
        if c not in c_count:
            c_count[c] = 1
        else:
```

```
            c_count[c] = c_count[c] + 1

    return dict(sorted(c_count.items()))

# Bigrams with intersection (ex: [1, 2], [2, 3], [3, 4])
def count_bigrams_w_i(_text):
    b_count = {}
    prev_char = _text[0]
    for c in _text[1:]:
        bg = prev_char + c
        prev_char = c
        if bg not in b_count:
            b_count[bg] = 1
        else:
            b_count[bg] = b_count[bg] + 1

    return dict(sorted(b_count.items()))

# Bigrams without intersection (ex: [1, 2], [3, 4])
def count_bigrams_wo_i(_text):
    b_count = {}
    i = 1
    while i < len(_text):
        bg = _text[i - 1] + _text[i]
        if bg not in b_count:
            b_count[bg] = 1
        else:
            b_count[bg] = b_count[bg] + 1
        i = i + 2

    return dict(sorted(b_count.items()))
```

In [ ]:
```
chars_freq_wspaces = count_chars(text)
chars_freq_wospaces = chars_freq_wspaces.copy()
del chars_freq_wospaces[' ']

bigrams_freq_w_intersect = count_bigrams_w_i(text)
bigrams_freq_wo_intersect = count_bigrams_wo_i(text)
```

Show symbol frequencies

In [ ]:
```
for k, v in chars_freq_wspaces.items():
    print(f"{k} : {v}")
```

```
  : 86793
а : 37904
б : 8174
в : 22994
г : 7676
д : 14226
е : 43426
ж : 4720
з : 7408
и : 37250
й : 6380
к : 16426
л : 18174
м : 15420
н : 36030
о : 56742
п : 15740
р : 24918
с : 26566
т : 31194
у : 12712
ф : 1612
х : 5356
ц : 2422
ч : 6828
ш : 3240
щ : 1846
ъ : 164
ы : 8952
ь : 7650
э : 1236
ю : 3406
я : 9048
```

Show bigram frequencies

In [ ]:

```python
alph = list(chars_freq_wspaces.keys())
print("   ", end='|')
for l_c in alph:
    print(f" '{l_c}' ", end='|')
for l_c in alph:
    print(f"\n'{l_c}'", end='|')
    for r_c in alph:
        k = l_c + r_c
        if k in bigrams_freq_w_intersect:
            print(f"{bigrams_freq_w_intersect[k]:5d}", end='|')
        else:
            print(f"{0:5d}", end='|')
```

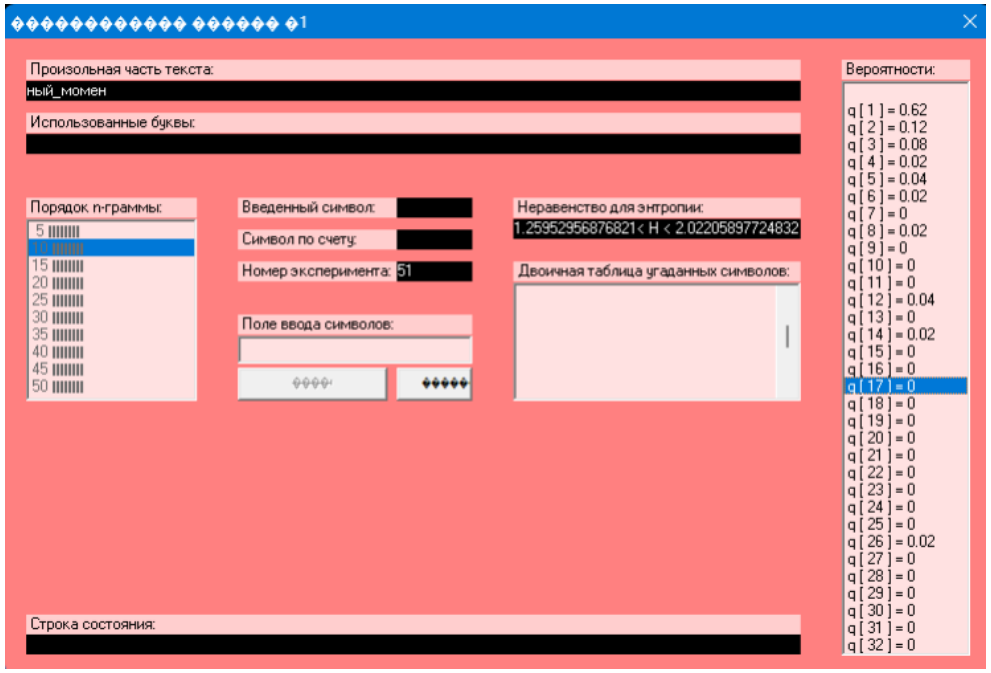
	' '	'а'	'б'	'в'	'г'	'д'	'е'	'ж'	'з'	'и'	'й'	'к'	'л'	'м'	'н'	'о'	'п'	'р'	'с'	'т'	'у'	'ф'	'х'	'ц'	'ч'	'ш'	'щ'	'ъ'	'ы'	'ь'	'э'	'ю'	'я'
' '	0	1790	3314	9052	1754	3626	1146	514	2046	5760	54	4112	1376	3208	9268	5460	10444	3268	7726	4094	2800	728	749	320	2080	382	6	0	0	0	1104	72	540
'а'	7586	10	488	1636	560	1036	954	856	1386	298	468	2572	2688	1642	3256	54	594	1914	2356	2938	48	48	672	804	690	664	122	0	0	0	4	762	798
'б'	152	500	2	12	0	20	1048	36	6	566	0	32	494	74	264	1704	20	582	192	6	666	0	106	0	0	4	294	140	1070	32	0	16	136
'в'	4476	2924	8	50	8	94	2190	2	216	1430	0	372	798	108	1220	4218	168	598	1558	196	304	0	2	28	66	302	12	0	1386	64	0	0	196
'г'	256	678	0	28	4	358	288	0	0	508	0	42	472	2	186	3774	0	656	26	14	366	0	0	0	6	10	0	0	0	0	0	2	0
'д'	670	2872	10	278	78	70	2566	18	14	1236	0	160	532	26	984	2134	122	506	412	38	754	0	14	12	68	46	0	6	296	134	10	8	152
'е'	10900	116	340	1060	1234	1878	810	338	972	132	1462	756	2488	2242	5884	246	606	3992	2790	2974	74	52	498	264	428	352	286	0	0	0	14	164	74
'ж'	70	558	30	0	6	372	2300	4	0	558	0	54	0	4	608	10	2	2	36	0	66	2	0	0	12	0	0	0	0	26	0	0	0
'з'	784	2248	102	458	110	472	226	32	6	496	0	86	78	336	546	496	60	146	18	2	268	0	2	16	2	0	0	8	154	52	4	58	142
'и'	10652	278	252	1570	278	592	1938	162	1438	1206	1036	1560	1846	1914	1914	282	176	872	1306	2506	34	46	1334	388	1140	162	86	0	0	0	0	488	1794
'й'	4376	0	0	0	38	68	4	0	4	0	0	54	0	42	330	222	10	4	798	82	0	2	2	44	162	138	0	0	0	0	0	0	0
'к'	2094	3156	8	76	2	2	378	124	2	2264	0	16	280	12	54	4862	12	1302	184	814	728	0	16	30	0	10	0	0	0	0	0	0	0
'л'	984	1970	38	4	102	58	3106	230	0	3374	0	216	248	0	430	2002	20	0	160	16	690	0	0	0	60	6	0	0	248	2662	0	540	1010
'м'	4472	1430	54	6	2	6	2322	4	10	2022	0	52	128	72	568	2250	136	18	96	0	944	20	0	10	10	6	0	0	590	18	0	0	174
'н'	1070	6156	50	14	98	338	5634	4	20	5800	0	352	0	18	2452	6742	8	130	608	1052	902	114	2	202	62	14	10	0	3244	336	0	38	560
'о'	11081	6	2788	4806	2624	3082	1624	1222	860	444	2662	952	3306	2980	3056	362	1142	3440	4006	3878	46	142	202	106	936	286	128	0	0	0	78	84	412
'п'	214	1004	0	36	20	2	1618	0	2	460	0	20	720	0	118	6148	126	4460	22	40	394	0	0	10	4	2	0	0	230	2	2	0	86
'р'	628	5210	56	302	194	122	3738	214	38	2414	0	108	24	546	554	5806	94	130	418	516	1468	336	58	16	20	132	8	0	880	284	10	32	562
'с'	1684	774	62	758	20	160	1846	8	0	1212	0	2840	1436	270	564	1878	1008	328	766	6960	842	18	118	26	160	92	0	6	138	742	0	38	1812
'т'	3914	3734	18	1694	16	52	2902	0	4	3088	0	402	68	38	836	6562	20	1550	1180	20	940	2	32	0	20	0	2	0	622	3074	8	16	380
'у'	2916	136	296	188	392	1192	176	784	80	8	36	730	318	432	182	4	822	388	828	878	0	0	94	22	658	130	254	0	2	0	2	736	28
'ф'	356	122	0	0	0	0	198	0	0	160	0	0	60	2	10	220	0	392	10	8	22	46	0	0	2	0	0	0	4	0	0	0	0
'х'	3228	274	2	124	0	2	176	0	0	80	0	0	18	92	116	1020	0	70	44	30	76	0	0	0	0	0	0	4	0	0	0	0	0
'ц'	92	188	0	16	0	62	558	0	0	1188	0	102	2	0	12	60	12	0	2	2	38	0	0	0	0	0	0	0	88	0	0	0	0
'ч'	102	1304	0	20	0	0	2062	0	0	976	0	102	4	2	694	12	0	20	0	1242	116	2	0	0	0	54	0	0	0	116	0	0	0
'ш'	62	282	0	0	0	0	1020	0	0	920	0	82	144	4	220	138	2	2	0	204	84	0	0	0	0	0	0	0	0	76	0	0	0
'щ'	4	180	0	0	0	0	958	0	0	586	0	0	0	0	48	0	0	2	0	0	36	0	0	0	0	0	0	0	0	32	0	0	0
'ъ'	4	0	0	0	0	0	86	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74
'ы'	2588	0	50	548	58	70	1222	2	26	24	638	32	462	878	76	0	120	48	276	372	4	0	1254	0	46	122	0	0	0	0	0	0	36
'ь'	4180	0	40	10	8	16	158	0	84	40	0	504	0	56	1260	72	2	0	476	90	0	4	0	56	12	306	6	0	0	0	0	218	52
'э'	26	0	2	22	0	2	0	0	0	0	4	54	54	14	16	0	4	18	18	930	2	50	0	4	0	12	0	0	0	0	0	0	4
'ю'	1598	4	124	0	26	194	4	36	16	0	2	8	4	58	6	0	0	40	88	556	0	0	0	4	120	2	496	0	0	0	0	20	0
'я'	5574	0	40	226	44	280	170	130	178	0	18	54	126	348	298	4	10	40	166	736	0	0	200	60	64	6	136	0	0	0	0	114	26

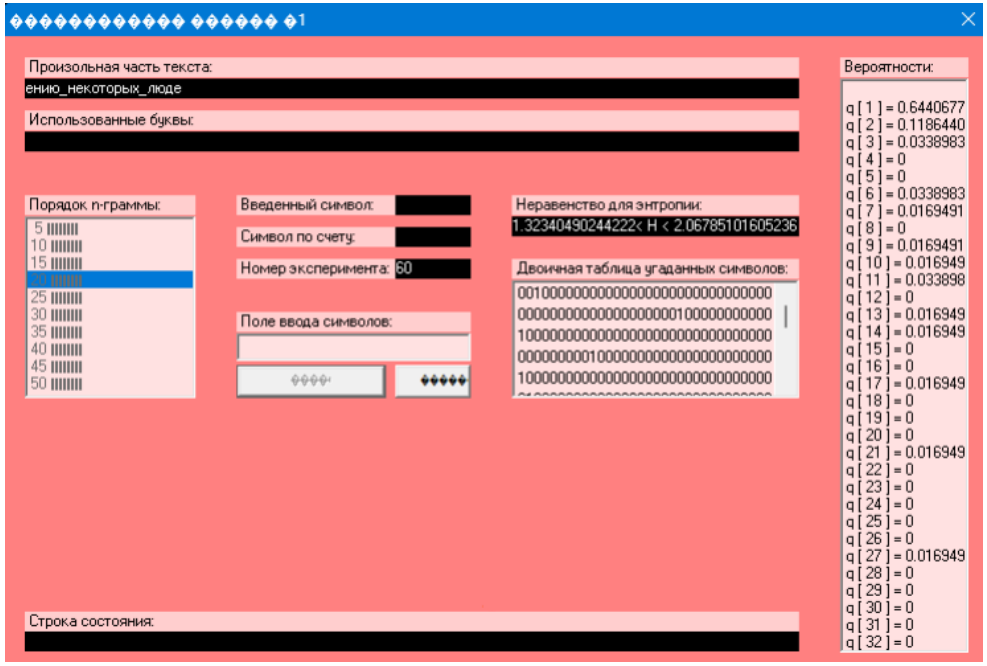
Calculate H_1 and H_2

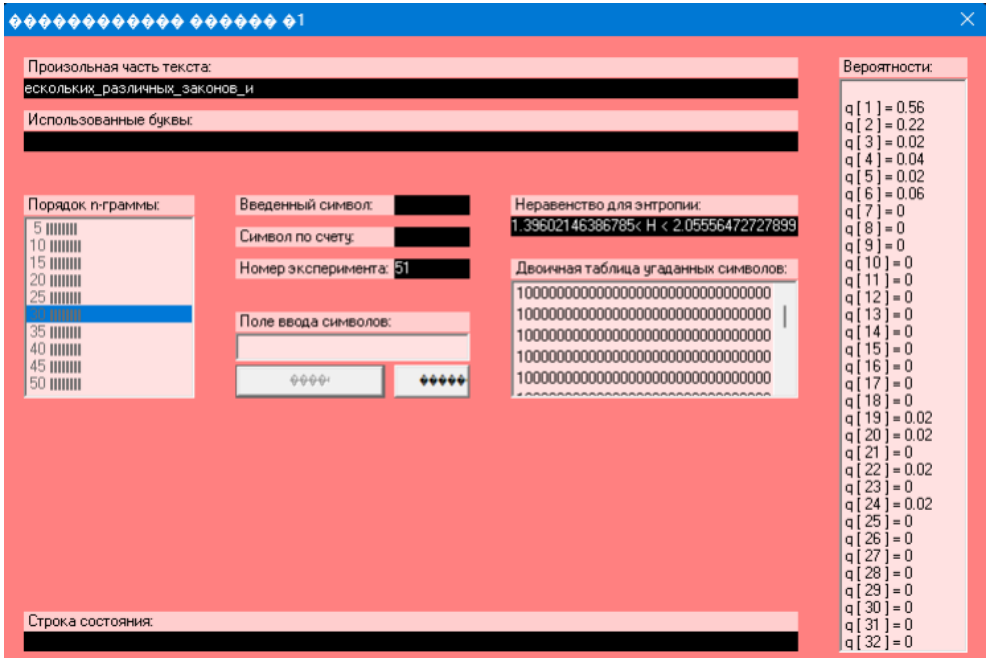
In [ ]:
```python
import math

char_amount = sum(chars_freq_wspaces.values())
t1 = [chars_freq_wspaces[k] / char_amount for k in chars_freq_wspaces.keys()]
H1 = -sum(a * math.log2(a) for a in t1)

bg_amount = sum(bigrams_freq_w_intersect.values())
t2 = [bigrams_freq_w_intersect[k] / bg_amount for k in bigrams_freq_w_intersect.keys()]
H2 = -sum(a * math.log2(a) for a in t2) / 2
print(f"H1 = {H1}")
print(f"H2 = {H2}")
```

```
H1 = 4.385129362944809
H2 = 3.9881215496418245
```

Обчислення $H^{(10)}$, $H^{(20)}$, $H^{(30)}$ за допомогою Cool Pink Program


Произольная часть текста:
ный_момен
Использованные буквы:
Порядок n-граммы:
5
10
15
20
25
30
35
40
45
50
Введенный символ:
Символ по счету:
Номер эксперимента: 51
Поле ввода символов:
Неравенство для энтропии:
1.25952956876821< H < 2.02205897724832
Двоичная таблица угаданных символов:
Вероятности:
q [1] = 0.62
q [2] = 0.12
q [3] = 0.08
q [4] = 0.02
q [5] = 0.04
q [6] = 0.02
q [7] = 0
q [8] = 0.02
q [9] = 0
q [10] = 0
q [11] = 0
q [12] = 0.04
q [13] = 0
q [14] = 0.02
q [15] = 0
q [16] = 0
q [17] = 0
q [18] = 0
q [19] = 0
q [20] = 0
q [21] = 0
q [22] = 0
q [23] = 0
q [24] = 0
q [25] = 0
q [26] = 0.02
q [27] = 0
q [28] = 0
q [29] = 0
q [30] = 0
q [31] = 0
q [32] = 0
Строка состояния:


Произольная часть текста:
ению_некоторых_люде
Использованные буквы:
Порядок n-граммы:
5
10
15
20
25
30
35
40
45
50
Введенный символ:
Символ по счету:
Номер эксперимента: 60
Поле ввода символов:
Неравенство для энтропии:
1.32340490244222< H < 2.06785101605236
Двоичная таблица угаданных символов:
Вероятности:
q [1] = 0.6440677
q [2] = 0.1186440
q [3] = 0.0338983
q [4] = 0
q [5] = 0
q [6] = 0.0338983
q [7] = 0.0169491
q [8] = 0
q [9] = 0.0169491
q [10] = 0.016949
q [11] = 0.033898
q [12] = 0
q [13] = 0.016949
q [14] = 0.016949
q [15] = 0
q [16] = 0
q [17] = 0.016949
q [18] = 0
q [19] = 0
q [20] = 0
q [21] = 0.016949
q [22] = 0
q [23] = 0
q [24] = 0
q [25] = 0
q [26] = 0
q [27] = 0.016949
q [28] = 0
q [29] = 0
q [30] = 0
q [31] = 0
q [32] = 0
Строка состояния:



Отримані результати:

$$1.2595 \leq H^{(10)} \leq 2.0221 \tag{1}$$
$$1.3234 \leq H^{(20)} \leq 2.0679 \tag{2}$$
$$1.3960 \leq H^{(30)} \leq 2.0556 \tag{3}$$

Обчислимо надлишковість:

Так як $H_0 = \log_2(32) = 5$, то за формулою надлишковості $R = 1 - \frac{H_\infty}{H_0}$:

Розглядаємо $H^{(i)}$ як наближення H_∞.

$$\begin{aligned} H^{(10)}: &\quad 0.404 \leq R \leq 0.748 \\ H^{(20)}: &\quad 0.464 \leq R \leq 0.735 \\ H^{(30)}: &\quad 0.588 \leq R \leq 0.721 \end{aligned} \tag{4}$$