

Лабораторна робота 2 з Симетричної Криптографії

Варіант: 2

Команда: Бондар, Кістаєв

Група: ФІ-03

Підготовча частина: оголошення констант, функція для зчитування і передобробки тексту

```
In [ ]: # Constants
ALPH = "абвгдежзийклмнопрстуфхцщъыьэюя"
PROBS = [0.0792, 0.0171, 0.0433, 0.0174, 0.0305, 0.0841, 0.0105, 0.0175,
          0.0683, 0.0112, 0.0336, 0.0500, 0.0326, 0.0672, 0.1108, 0.0281,
          0.0445, 0.0533, 0.0618, 0.0280, 0.0019, 0.0089, 0.0036, 0.0147,
          0.0081, 0.0037, 0.0002, 0.0196, 0.0192, 0.0038, 0.0061, 0.0213]
ALPH_SIZE = len(ALPH)
RING = {}
for char in ALPH:
    pos = ALPH.index(char)
    RING[char] = pos
EXPECTED_I = sum([(p ** 2) for p in PROBS])
I0 = 1.0 / ALPH_SIZE
FILE_NAME = "text.txt"
FILE_TO_ENCRYPT = "text_to_encrypt.txt"
```

```
In [ ]: # Text preprocessing
def transform_symbol(_c):
    if 'a' <= _c and _c <= 'я':
        return _c
    elif _c <= 'Я' and _c >= 'А':
        return _c.lower()
    elif _c == 'Ё' or _c == 'ё':
        return 'e'
    else:
        return ""

def preprocess_text(_text):
    text_formatted = ""
    # Change symbols according to requirements
    for c in _text:
        text_formatted += transform_symbol(c)
    # Remove consecutive spaces
    text_formatted = ' '.join(text_formatted.split())
    return text_formatted

def read_text(filename):
    f = open(filename, "r", encoding='utf-8')
    text = f.read()
    f.close()
    return preprocess_text(text)
```

Частина 1: Застосування шифру Віженера, обчислення індексів відповідності

Для зашифрування ми обрали уривок з тексту для лабораторної роботи 1.

```
In [ ]: # Function to evaluate index of coincidence
def coincidence_index(text: list) -> int:
    sum = 0
    for c in ALPH:
        occurances = text.count(c)
        sum += occurances * (occurances - 1)
    return sum / (len(text) * (len(text) - 1))
```

```
In [ ]: def encrypt_text(text: str, key: str):
    res = ""
    r = len(key)
    for i in range(0, len(text), r):
        for j in range(r):
            if i + j == len(text):
                break
            id = (RING[text[i + j]] + RING[key[j]]) % ALPH_SIZE
            res += ALPH[id]

    return res
```

```
In [ ]: key_2 = "не"
key_3 = "пон"
key_4 = "зроз"
key_5 = "катка"
key_n = "фанаткаспартак"

decrypted = read_text(FILE_TO_ENCRYPT)
encrypted_2 = encrypt_text(decrypted, key_2)
encrypted_3 = encrypt_text(decrypted, key_3)
encrypted_4 = encrypt_text(decrypted, key_4)
encrypted_5 = encrypt_text(decrypted, key_5)
encrypted_n = encrypt_text(decrypted, key_n)

print("Indicies of coincidence:")
print(f"I_0 theoretical: {I0}")
print(f"I_M theoretical: {EXPECTED_I}")
print(f"I for Message: {coincidence_index(decrypted)}")
print(f"I_{len(key_2)}: {coincidence_index(encrypted_2)}")
print(f"I_{len(key_3)}: {coincidence_index(encrypted_3)}")
print(f"I_{len(key_4)}: {coincidence_index(encrypted_4)}")
print(f"I_{len(key_5)}: {coincidence_index(encrypted_5)}")
print(f"I_{len(key_n)}: {coincidence_index(encrypted_n)}")
```

```
Indicies of coincidence:
I_0 theoretical: 0.03125
I_M theoretical: 0.055300129999999996
I for Message: 0.057782276937813626
I_2: 0.04393647476297772
I_3: 0.043993077098912585
I_4: 0.038089018058322176
I_5: 0.038977239329915425
I_15: 0.036002351173954214
```

Як власне можна побачити, зі збільшенням довжини ключа, значення індексу відповідності для тексту все більше відрізняється від значення для початкового тексту і все більше прямує до значення I_0 . Це відбувається, так як, при шифруванні тексту більшим ключем, він все більше втрачає статистичні властивості мови.

Частина 2: Розшифровка тексту відповідно до варіанту (2)

```
In [ ]: # Separate cyphertext according to key Lenght
def divide_into_blocks(text: list, r: int):
    res = []
    for i in range(r):
        block = text[i::r]
        res.append(block)
    return res

def coinc_idx_for_blocks(text: list, r: int):
    idxs = []
    blocks = divide_into_blocks(text, r)
    for i in range(r):
        idxs.append(coincidence_index(blocks[i]))
    return idxs

def summary_coinc_idx_for_diff_r(text: list):
    idx_table = {}
    # if we assume that len(KEY) divides len(text) -> itr through divisors
    # also we decided to make unified solution (check also if key length > 30)
    itr_range = range(2, int(len(text) / 2))
    for r in itr_range:
        idx_table[r] = coinc_idx_for_blocks(text, r)
    return idx_table
```

```
In [ ]: from numpy.linalg import norm
# Instead of comparing average idx diff, we measure vector distance and compare
def compare_idx_vectors(left: list, right: list) -> bool:
    left_dist = norm([EXPECTED_I - idx for idx in left])
    right_dist = norm([EXPECTED_I - idx for idx in right])
    return left_dist <= right_dist

def get_best_r(text: list):
    best_r = 2
    idx_map = summary_coinc_idx_for_diff_r(text)
    for (r, idx) in idx_map.items():
        if compare_idx_vectors(idx, idx_map[best_r]):
            best_r = r
            print(f"log I_{r}: {sum(idx) / len(idx)}")
    return best_r
```

Метод частот

```
In [ ]: # Basic frequency method
def crack_key_freq(text, r) -> str:
    key = ""
    blocks = divide_into_blocks(text, r)
    for i in range(r):
        y = RING[max(blocks[i], key = blocks[i].count)]
        x = PROBS.index(max(PROBS))
        k_i = (y - x) % ALPH_SIZE
        key += ALPH[k_i]
    return key

In [ ]: # Modified frequency method which allow leave correct key parts
# Execution without optional parameters is same as 'crack_key_freq()'
def crack_key_freq_mod(text, r, found_pos=[], iteration=0, prev_key="") -> str:
    key = ""
    blocks = divide_into_blocks(text, r)
```

```

for i in range(r):
    if i not in found_pos:
        y = RING[max(blocks[i], key = blocks[i].count)]
        x = PROBS.index(sorted(PROBS)[-(iteration + 1)])
        k_i = (y - x) % ALPH_SIZE
        key += ALPH[k_i]
    else:
        key += prev_key[i]
return key

```

Метод $M_i(g)$

```

In [ ]: def crack_key_Mi(text, r) -> str:
        key = ""
        blocks = divide_into_blocks(text, r)
        for i in range(r):
            k_i = 0
            M_max = 0
            for g in range(ALPH_SIZE):
                try_M = 0
                for t in range(ALPH_SIZE):
                    try_M += PROBS[t] * blocks[i].count(ALPH[(t + g) % ALPH_SIZE])
                if try_M > M_max:
                    M_max = try_M
                    k_i = g
            key += ALPH[k_i]
        return key

```

Знаходження довжини ключа

```

In [ ]: text = read_text(FILE_NAME)
        # Best key length
        r = get_best_r(text)
        print(f"Best key length: {r}")

```

```

log I_2: 0.03626820548217928
log I_14: 0.05528168514213951
Best key length: 14

```

Пошук ключа

```

In [ ]: # Get indices with correct key parts
        def success_idxes(right_key, to_improve, r):
            res = []
            for i in range(r):
                if right_key[i] == to_improve[i]:
                    res.append(i)
            return res

        # This is to complete frequency method
        # Kind of simulation of checking artifacts in text manually :)
        def crack_freq_iterable(text: str, r: int, expected_key: str):
            key = crack_key_freq_mod(text, r)
            for i in range(1, ALPH_SIZE):
                print(f"Freq key (itr {i - 1}): {key}")
                if key != expected_key:
                    found = success_idxes(expected_key, key, r)
                    key = crack_key_freq_mod(text, r, found_pos=found, iteration=i, prev_key=key)
                else:
                    break
            return key

```

```
In [ ]: keyMi = crack_key_Mi(text, r)
print(f"Mi key: {keyMi}\n")
keyFreq = crack_freq_iterable(text, r, keyMi)
print(f"Freq key result: {keyFreq}")
```

Mi key: последнийдозор

Freq key (itr 0): жосвеыдиадозор

Freq key (itr 1): последнийдозор

Freq key result: последнийдозор

Дешифрування тексту

Так як в обох методах ми отримали однакові ключі, то використаємо тільки один з них.

```
In [ ]: # Function to decrypt texts with key
def decrypt_text(text, key):
    res = ""
    r = len(key)
    for i in range(0, len(text), r):
        for j in range(r):
            if i + j == len(text):
                break
            id = (RING[text[i + j]] - RING[key[j]]) % ALPH_SIZE
            res += ALPH[id]

    return res
```

```
In [ ]: decrypt_text(text, keyMi)
```

Out[]: 'какаясмогэтосделатьспросилгесерипочемуэтогонесмогсделатывыстоялипосредибескрайнейсеройравнинывзгляднефиксироваляркихкрасоквцелойкартиненостоиловсмотретьсявотдельнуюпесчинкуитавспыхивалазолотомбагрянцемлазурьюзеленынадголовойзастылобелоес розовымбудтомолочнуюрекуперемешалискисельнымиберегамидаивплеснуливнебесаещедуветерибылохолодномневсегдахолодноначетвертомслоесумраканоэтоиндивидуальнаяреакциягесерунапротивбыложарколицораскраснелосьполбустекаликапелькипотамненехватаетсилысказалаялицогесерасовсемпобагровелоответнеправильныйтывысшиймагтакполучилосьслучайнотывысшийпочемувысшихмаговтакжезываютмагамивнекатегорийпотомучтооразницавсилеждуниминастольконезначительначтонеможетбытьисчисленаиневажноопределитьктыосильнееактослабеепробормоталяборисигнатъевичяпонимаюмненехватаетсилыянемогупротинаптыйслойгесерпосмотрелсебеподноиподделноскомботинкапесокподбросилввоздухагнулвпередисчезэточтосоветяподбросилпередсобойпесокшагнулвпередтщепотытаясьпойматьсвоютеньтенинебылоничегонеизменилосьяпопрежнемуоставалсяначетвертомслоеистановилосьвсехолоднеепаротмоегодыханияуже нерассеивалсябелымоблачкомakoлючимиигламиосыпалсянапесокразвернувшисьэтовсегдапрощесихологическиискатьвыходпозадиясделалшагивышелнатретийуровеньсумракавбесцветныйлабиринтизъеденныхвременемкаменныхплитнадкоторымисерелонизкоезастывшеенебокоегдепокамнустелилисьвысохшиеестеблипохожиенаприбитыйморозомвьюнокпереростокещашагвторойслойсумракакаменныйлабиринтнакрылипереpletенныеветвиивещепервыйслойуженекаменьужестеныиокназнакомыестенымосковскогоофисаночногодозораветгосумеречномобличьепоследнимусилиемывывалилсяизсумракавреальныймирпрямокабинетгесераразумеетсяшефужесиделвкреслеаяпошатываясьстоялпереднимнукакакконмогменяопередитьведьонпошелнапятьтислойаяначалвыходитьизсумракаогдаяувиделчтоутебяничегонеполучаетсясказалгесердаже неглядянаменятывышлиизсумраканепрямуюизпятагословнастоящиймирянемогскрытьудивлениядачтотебяудивляетяпожалпечаминичегонеудивляетеслигесерзахочетпреподнести мне сюрпризугеобудетогромныйвыборяоченьмногогоне знаиэтообидносказалгесерсядьгородецкийселнапротивгесерасложилрукинаколенихдажеголовуопустилбудтовчемточувствовалсвоювинуантонхорошиймагвсегдадостигаетсвоегомогуществавнужноевремясказалшефпоканестанешьмудреене станешь сильнеепоканестанешь сильнее не овладеешьвысшеймагиейпокане овладеешьвысшеймагиейневлезешьвопасныеместаутебя ситуация уникальнаятыпопалподопморщилсязаклятиешфурантысталвысшиммагомнебудучикэтогоготовымдаутебяестьсиладатыумеешьуправлятьточтотыструдомделалраньшетеперьне составляетпроблемсколькотыпробылначетвертомслоесумракаи сидишькакнигде не бывалоновотточего ты неумелраньшеонзамолчала научусьборисигнатъевичсказалявконцеконцоввсе признаютчтояделаюзначительныеуспехиольгасветланаделаетшьлегкопризналгесертыженесовсемидиотчтобынеразвизатьсяносейчас тынапоминаешьмне неопытноговодителякоторыйполгодапокаталсянажигуляхивдругселзарульгочногооферринетхуже зарулькарьерногосамосвалабелазавесомдвести тоннчтоползетсебепоспираливыезжаетизкарьераарядомпропавшесотнюметроватамвнизуедутдругиесамосвалыоднотвое не верное движениеизрезкийповоротуляилидрогнувшаяаянапедалиногаглохобудетвсемпонимаюкаквнуловывысшениервалсяборисигнатъевичэтовыменяотправиливпогонюзакостейятебянигде неупрекаюпытайсямногомунаучитьсясказалгесеридовольнонепоследовательнодобавилхотьтыоднаждыиотказалсябытьмоимученикомяпромолчалоткрывапкувеликийгесерзавязывалтесемкинабантикаобнаружилчетыресвеженькиеещепакнущиетипографскойкраскойгазетныевырезкии факситрифотografiитривырезкибылинаанглийскомна нихясосредоточилсявпервуюочередьперваявырезкапредставляласобойкороткуюзаметкуо происшествии втуристическоматтракционеподземельяшотландиикакаяпонялвэтомзаведениидовольнотакибанальномвариантекомнатстрахаиззатехническихнеполадкопгибрусскийтуристподземельябылизакрытыполицияпроводитрасследованиеивыясняетнетливтрагедиивиниперсоналавтораязаметкабылакудаподробнеепротехническиенеполадкиуженебылонисловатекутбылнемножкосуховатымдажепедантичнымснарастающимволнениемпрочиталчтопогибшийдвадцатипятилетнийвикторпрохоровучилсявэдинбургскомуниверситетебылсыномрусскогополитикавподземельяотправилсясместесневестойприлетевшейизроссиивалериейхомконарукахкоторойискончалсяотпотерикровивтемнотетуристическогоаттракционачтооперерезалемугорлоиличтооперерезалобедолагасиделвместесневестойвлодочкекотораямедленноплылапокровавойреке мелкойканавкевокругзамкавампиrowвозможнойзастеныторчалакакаятоостраяжелезкакотораяиполоснулавикторупошеедочитавдоэтогомestayвдохнул ипосмотрелнагесераутебявсегдазамечательнополучалосьэээсвампирасказалшефна секундуоторвавшисьотсвоихбумагтретьязаметкабылаизкакойтожелтойшотландскойгазетенкиивоттутконечножеавторрассказалстрашнуюисториюпросовременныхвампировкоторыево мракеаттракционовсосуткровьсвоихжертвединственнойоригинальнойдетальюбылоутверждениежурналистачтообычновампирывысасываютсвоихжертвнена смертьнорусскийстуденткакполагенорусскомубыл настолькопьянчтобедныйшотландскийвампиртуже захмелелиувлексянесмотрянавсютрагичностьисторииязасмеялсяжелтаяпрессаонавовсеммиреодинаковасказалгесернеподнимаяглазсамоеужасноечтотаквсебылосказалякроме пьянстваконечнокружапи вазаобедомсогласилсясгесерчетвертаявырезкабылаизкакойтонашейгазетынекрологсоболезнованияленидупрохоровудепутатугосударственнойдумчейсынтрагическипогибвзяллистокфаксаэтокакяпредполагалбылодонесениеотногодозорагородаэдинбургашотландиявеликобританиянемножко необычноказалсялишьадресатсамгесеранеоперативныйдежурныйилируководительмеждународногоотделаитонписьмачутьболееличныйчемполагаетсяявофициальныхдокументахосодержаниеменянеудивилосприскорбиемообщаемпорезультатамтщательнопроведенногодознанияполнаяпотерякровипризнаковинициацииневыявленопроведенныепоискирезультатовнедалипривлеченылучшиесилыеслимосковскоеотделениесчитаетнеобходимымнаправитьпередавайсамые теплыеприветьюльгеоченьрадазатебястарыйковтвойлистокфаксаотсутствовалвидимотамбылиисключительноличныйтекстпоэтомуиподписаниянеувиделфомалермонтсказалге

серглава шотландского дозора старей друг ага задумчиво протянул значительная в зглядю пять встретились
нетуж родственник лион михаил юрьевич сам спросишь сказал гесеря другом коэ то командир коэ то гесер за
пнул ся и с явным недовольством покосился на исток коэ то коэ то тебя жу не касает ся я посмотрел на фото граф
и и молодой человек это и был бедолага виктор де вушка совсем юная его невеста что тут гадать и мужик постарше
отец виктора ко с венные данные его ворят о нападении вампира но почему ситуация требует нашего вмешательства
ва спроси я наши соотечественники часть когибнут за рубежом и от вампиров тоже вы не доверяете фоме и его
подчиненным доверяю но у них мало опыта шотландия мирная уютная спокойная страна они могут не справиться
ты часть ко и мел делос вампира и конечно и все таки делов том что его отец политик гесер по морщился да ка ко
й он политик бизнесмен пробрал ся в депутаты на го лосования х ж мет кнопки хонь ку коротко и ясно не вер
ю что не то со бой причины гесер вздохнул отец юши двадцать лет назад было определе н как потенциал ый свет
лый и иной до вольно си лый и от инициации от казал ся обьявив что хо чет ста ть ся че лове ком темных сразу же по с
ла л прочь но с нами под держи вал не ко то рые кон так ты и но г да по мо г а ля ки в нул да слу чай редкий не ча ст о лю ди от
ка зы ва ют ся от та ких воз мож ностей что от кры ва ют ся пе ре ди ны ми мож но ска зать что я чув ству ю се бя ви но ва тым
пе ре д про хо ро вым стар шим ска зал ге се ри ес ли у же не мо гу по мо чь ся ну то не по зво ляю гоуб ий цеу и ти бе з на ка зан
ны м ты по ед ешь в э дин бург най дешь э то го су ма с ше дше го кро во со са и раз веешь по ве тру э то был при ка зно я и без т
о го не со би рал ся спор ить ко я не во лья но за пнул ся ко гда лет еть зайд и в ме жду на род ный от дел те бе дол жны бы ли
од го то вить до ку мен ты би ле ты де нь ги и ле ген ду '