Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”
Фізико-технічний інститут

Лабораторна робота № 3
з предмету «Криптографія»

«*Криптоаналіз Афінної біграмної підстановки*»

Виконали:
Студенти 3 курсу,
ФТІ, групи ФБ-05
Супрун Максим, Алькова Аліна

Київ - 2022

Мета роботи:
Набуття навичок частотного аналізу на прикладі розкриття моноалфавітної підстановки; опанування прийомами роботи в модулярній арифметиці.

Порядок виконання роботи:

0.Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.

1. Реалізувати підпрограми із необхідними математичними операціями: обчисленням оберненого елементу за модулем із використанням розширеного алгоритму Евкліда, розв'язуванням лінійних порівнянь. При розв'язуванні порівнянь потрібно коректно обробляти випадок із декількома розв'язками, повертаючи їх усі.

2.За допомогою програми обчислення частот біграм, яка написана в ході виконання комп'ютерного практикуму №1, знайти 5 найчастіших біграм запропонованого шифртексту (за варіантом).

3.Перебрати можливі варіанти співставлення частих біграм мови та частих біграм шифртексту (розглядаючи пари біграм із п'яти найчастіших). Для кожного співставлення знайти можливі кандидати на ключ (a,b) шляхом розв'язання системи (1).

4.Для кожного кандидата на ключ дешифрувати шифртекст. Якщо шифртекст не є змістовним текстом російською мовою, відкинути цього кандидата.
Повторювати дії 3-4 доти, доки дешифрований текст не буде змістовним.

11 Варіант

Хід Роботи:

Part 0-1:

Ознайомившись з методичними вказівками нам вдалося легко реалізувати потрібні у подальшій роботі функції. Короткий опис використаних допоміжних функцій наведемо нижче(детальніше див. файл `lab3.py/lab3.ipynb`)

У комірці 1 нижче представлені допоміжні функції + функції по частині 1

- monofrequency(text) - функція з першої лаби, шукає частоти літер
- h_count(text_with_frequencies) - ентропія
- bifrequency(text) - частотит звичайних біграм(також з лаби 1)
- letter_to_number(letter) - перетворює літеру у її індекс за поданим алфавітом
- bigram_to_number(bigram) - за допомогою попередньої перетворює біграм у числа
- number_to_bigram(n) - числа у біграми
- gcdExtended(a, b) - gcd через розширений алгоритм Евкліда
- modInverse(a, m) - оберенений елемент за модулем
- linearCongruence(a, b, n) - для лінійних порівнянь

Part 2:

Для частини з виведенням топу 5 найчастіштх біграм шифртексту використали функцію з лаби 1, яка ділить текст на звичайні біграм і знаходить їх частоту + модифікували її додавши в кінці лише вивід 5 найчастіших. Отримали такий результат:

```
{'нк': 0.014594735470419598, 'юж': 0.013552254365389628, 'хб': 0.012770393536617148, 'шь': 0.012770393536617148, 'мк': 0.012249
152984102164}
```

Part3:

```
['ст', 'но', 'то', 'на', 'ен'] #top 5 bigrams from ru texts
```

В цій ми співставляли найчастіші біграми мови, та найчастіші біграми шифрованого тексту, знайдені на попередньому кроці та записували це у масив(функція `find_all_couples()`) Такий масив необхідні для подальшого формування системи рівнянь:

$$\begin{cases} Y^* \equiv aX^* + b \pmod{m^2} \\ Y^{**} \equiv aX^{**} + b \pmod{m^2} \end{cases},$$

Яка потім знадобиться для знаходження пари ключів

Потім ітеруючись по кожному співставленню, записаних у масив, знаходимо можливі кандидати на ключ (a,b) шляхом розв'язання рівняннь(функція `find_keys()`), що випливають з наведених вище. Кандидати на ключ також записуємо у багатовимірний масив.

$$Y^* - Y^{**} \equiv a(X^* - X^{**}) \pmod{m^2}.$$ Для можливих (a)

$$b = (Y^* - aX^*) \bmod m^2.$$ Для можливих (b)

Part 4:

У комірці 3 нижче наведено фінальні функції для розшифрування

- find_keys(all_couples) - знаходу можливі ключі формуючи массив коренів рівнянь які були отримані з пар вище
- AffineDecrypt(text, keys) - функція дешифрує афінний шифр
- recognize - розпізнає текст під час дешифрування за 3 критеріями (відсутність заборонених біграм, порівняння за топом частіших біграм, порівняння за значенням ентропії)

В цій частині реалізовуємо фунцію – дешифратор, опираючись на таке рівняння:

$$X_i = a^{-1}(Y_i - b) \bmod m^2$$

Функція `recognize()` розпізнає серед можливих відкритих текстів, змістовний, то й що потрібний тобто і виводить відповідний до нього ключ. Тут все доволі легко, функція проходиться по кожній парі ключів, викликає функцію дешифратор, розшифровує та перевіряє на змістовність. В якості ідентифікаторів змістовності тексту спочтаку було обрано лише варіант з перевіркою на заборонені біграми, проте згодом вирішено додати ще дві. І того маємо такі критерії обрані на перевірку:

- Критерій заборонений l-грам(повний список неможливих біграм наведено в коді)
- Критерій частоти l-грам(перевіряли що б у відкритому тексті найчастіша біграм було у або ‘ст’ або ‘но’ або ‘то’. Вирішили шукати саме серед топу 3 тому що вони часто можуть змінюватись за місцями у звичайних текстах)
- Критерій ентропії. Як відомо ентропія російської мови ≈4,35, тож для можливого проміжку нашої ентропії спочатку хотіли взяти від 4.3 до 4.45. Проте саме той текст що був потрібний не попадав під такий проміжок, тому збільшили до 4.5(і того від 4.35 до 4.5)

На виході отримали потрібний ключ та розшифрований текст.

```
----Your key (ɔ◔‿◔)ɔ [703, 956]
----Your text ⇩
```

Розшифрований текст:

```
Хорошосэрбиллнехотясунулденьгивкарманвотчтобиллвыпростопосеетеэтуновуютравук
огданибудьвдругойразкактолькояпомрунадругойжеденьможетеперекопатьэтучертовул
ужайкунукакхватитувастерпенияподождатьещелетпятьшестьчтобыстарыйболтунуспело
тдатьконцыужбудьтеувереныподождусказалбиллсамнезнаюкаквамобяснитьнодляменяжу
жжаньеэтойкосилкисамаяпрекраснаямелодиянасветевнейвсяпрелестьлетабезнееябыуж
аснотосковалибеззапахасвежескошеннойтравытожебиллнагнулсяиподнялсземликорзин
куяпошелковрагувыславныйюношаивсепонимаетеяуверенизвасполучитсяблестящийиумн
ыйрепортерсказалдедушкапомогаяемуподнятькорзинкуявамэтопредсказываюпрошлоутр
онаступилполденьпослеобедадедушкаподнялсяксебенемногопочиталуиттиераикрепкоу
снулкогдаонпроснулсябылотричасавокнавливалсяяркийивеселыйсолнечныйсветдедушк
алежалвкроватиивдругвздрогнулслужайкидоносилосьпрежнеезнакомоенезабываемоежу
жжаньечтоэтосказалонктотокоситтравуноведьеетолькосегодняутромскосилионещепос
лушалдаконечноэтожужжиткосилкамернонеутомимодедушкавыглянулвокноиахнулдаведь
этобиллэйбиллфорестервамчтосолнцеудариловголову...
```

Повний текст у файлі `final_result.txt`

Висновки

Під час виконання практикуму, навчилися створювати та розуміти принцип дії програми автоматичного розпізнання змістовного тексту, спробували провести криптоаналіз афінного шифру біграмної заміни та знайти вірний ключ, для розшифрування тексту. Набуті навички знадоблятся у подальших практикумах та професійні діяльності.

*Додаток: варто зазначити що серед проблем, які виникали під час роботи над поставленою роботою, найскладнішими виявились не безпосередньо пов’язані з кодом, а з кодування файлу, який містив ШТ. Через це виникло багато перешкод. Спочатку отримували наче добре розшифрований текст, але в середині містились якісь незмістовні слова, і через це в загалі від початку не могли отримати хоч якийсь вірний ключ, бо жодний не детектувався. Шукали проблеми у коді, бо постійно виникали помилки, в різних функціях особливо, яка мала перетворювати літери у числа. Потім пощастило просто самостійно закинути текст варіанту у ворд, а звідти у тхт файлик, мовляв, а може спрацює так. Не помились – дійсно проблема у тих початкових файлах. Тому опісля знайденого кореня всіх проблем, була дописана додаткова функція txteditor(), взята з лаби 1, яка відкриває файл правильно, і додатково очищає від зайвих символів.