Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**«Отчёт по лабораторной работе 4»**

“ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ ШИФРОВ НА ОСНОВЕ ПОДСТАНОВКИ (ЗАМЕНЫ) СИМВОЛОВ”

**Выполнила:** студентка 3 курса

4 группы специальности ПОИТ

Сапегина Екатерина Игоревна,

Вариант 11

Минск 2023

***Цель:*** изучение и приобретение практических навыков разработки и использования приложений для реализации подстановочных шифров.

***Задачи:***

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости подстановочных шифров.

2. Ознакомиться с особенностями реализации и свойствами различных подстановочных шифров на основе готового программного средства (L\_LUX).

3. Разработать приложение для реализации указанных преподавателем методов подстановочного зашифрования/расшифрования.

4. Выполнить исследование криптостойкости шифров на основе статистических данных о частотах появления символов в исходном и зашифрованном сообщениях.

5. Оценить скорость зашифрования/расшифрования реализованных способов шифров.

6. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

***Практическая часть:***

|  |  |  |
| --- | --- | --- |
| 11 вариант | Белорусский алфавит | 1. Шифр Цезаря с ключевым словом, ключевое слово – інфарматыка, а = 2  2. Таблица Трисемуса, ключевое слово – собственное имя |

Шифр Цезаря и шифр Трисемуса - это классические методы шифрования, используемые для защиты сообщений.

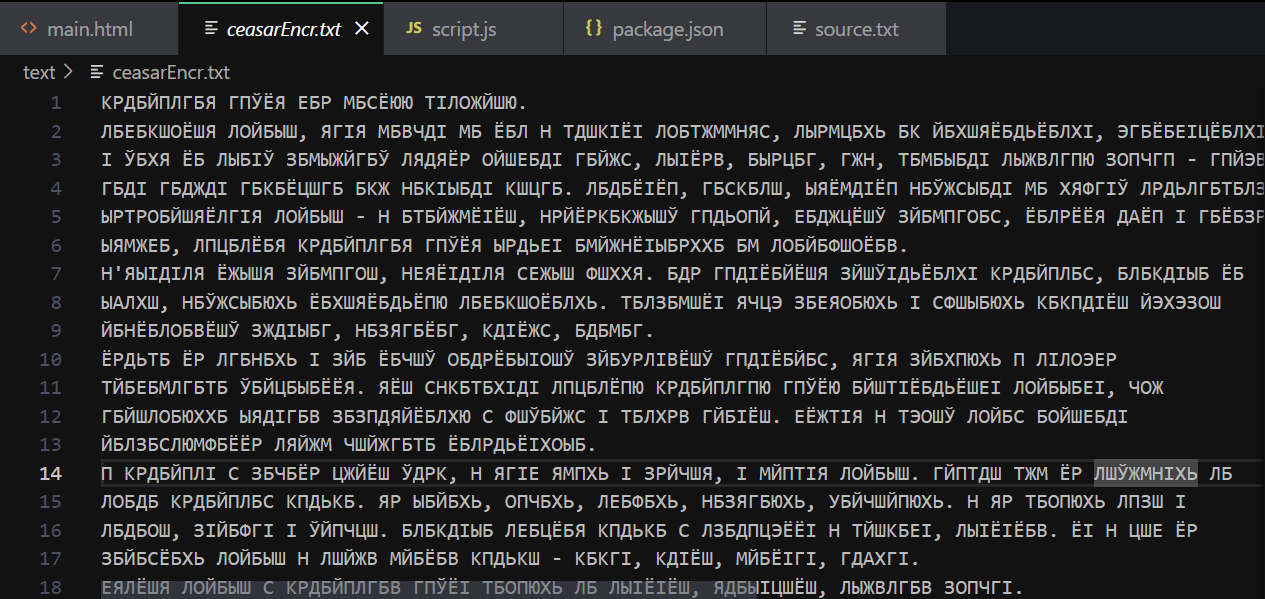
***Шифр Цезаря*** основан на замене каждой буквы в сообщении на другую букву, находящуюся на некотором фиксированном расстоянии в алфавите. Например, если расстояние равно 3, буква A заменяется на D, буква B на E и так далее. Алгоритм шифра Цезаря можно описать следующим образом:

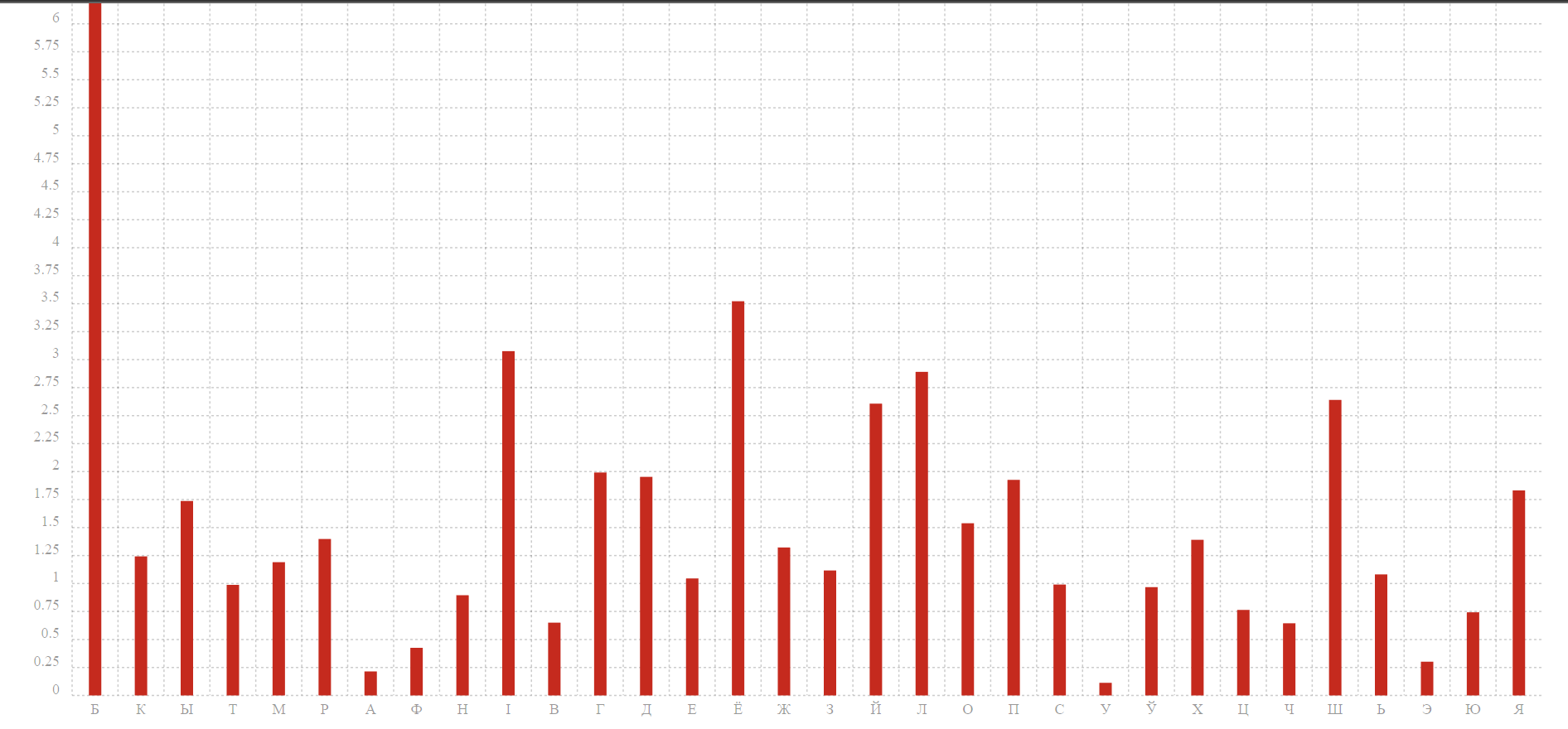
1. Определить расстояние сдвига.
2. Разбить сообщение на отдельные буквы.
3. Заменить каждую букву на другую букву, находящуюся на заданном расстоянии в алфавите.
4. Собрать зашифрованное сообщение.

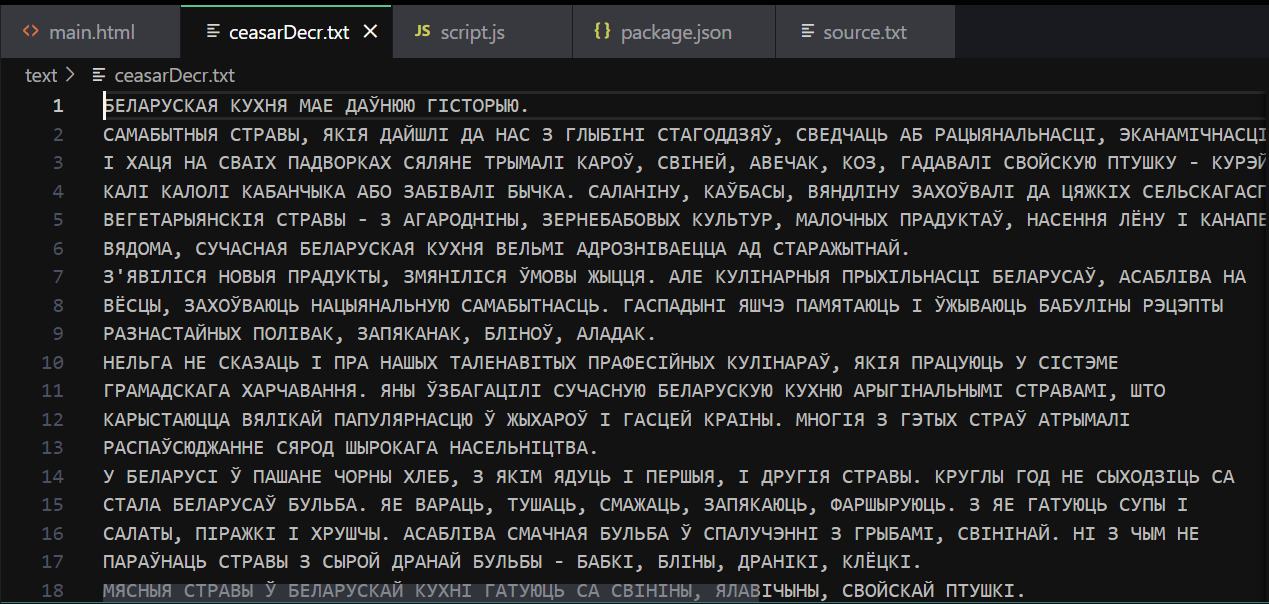
***Код реализации:***

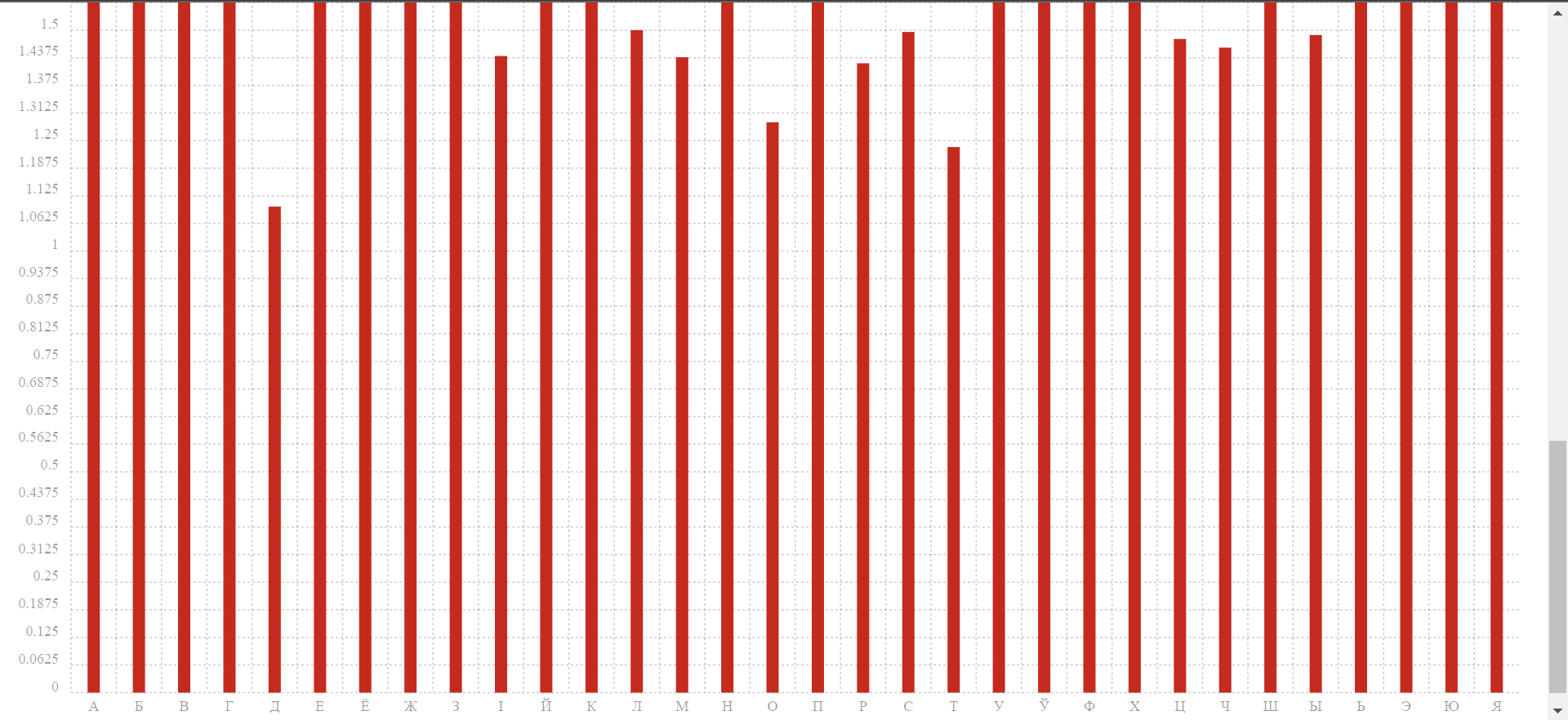
|  |
| --- |
| *let CaesarEncrypt = async () => {*  *console.time();*  *let response = await getText('text');*  *let text = (await response.text()).split('');*  *length = text.length*  *let al = await CreateAlphabet(belAl, 2, 'інфарматыка'.toLocaleUpperCase());*  *let res = 'ceasarEncr --- ';*  *text.forEach((sym) => {*    *if (belAl.includes(sym))*  *{*  *res += al[belAl.indexOf(sym)];*  *probability[belAl.indexOf(sym)] += 1;*  *}*  *else res += sym;*  *})*  *belAl.forEach((e,int) => {*  *probability[int] = probability[int] / belAl.length;*  *})*  *fetch('tofile', {*  *method: "post",*  *cache: "no-cache",*  *mode: "no-cors",*  *body: res*  *})*  *.catch(err => console.log(err))*  *new Chartist.Bar('#chart', {*  *labels: al,*  *series: [probability]*  *});*  *console.timeEnd();*  *}*  *let CaesarDecrypt = async () => {*  *console.time();*  *let response = await getText('textDercCes');*  *let text = (await response.text()).split('');*  *length = text.length*  *let al = await CreateAlphabet(belAl, 2, 'інфарматыка'.toLocaleUpperCase());*  *let res = 'ceasarDecr --- ';*  *text.forEach((sym) => {*    *if (al.includes(sym))*  *{*  *res += belAl[al.indexOf(sym)];*  *probability[belAl.indexOf(sym)] += 1;*  *}*  *else res += sym;*  *})*  *belAl.forEach((e,int) => {*  *probability[int] = probability[int] / belAl.length;*  *})*  *new Chartist.Bar('#chartEn', {*  *labels: al,*  *series: [probability]*  *});*  *fetch('tofile', {*  *method: "post",*  *cache: "no-cache",*  *mode: "no-cors",*  *body: res*  *})*  *.catch(err => console.log(err))*  *console.timeEnd();*  *}* |

***Результат:***

******

******

******

******

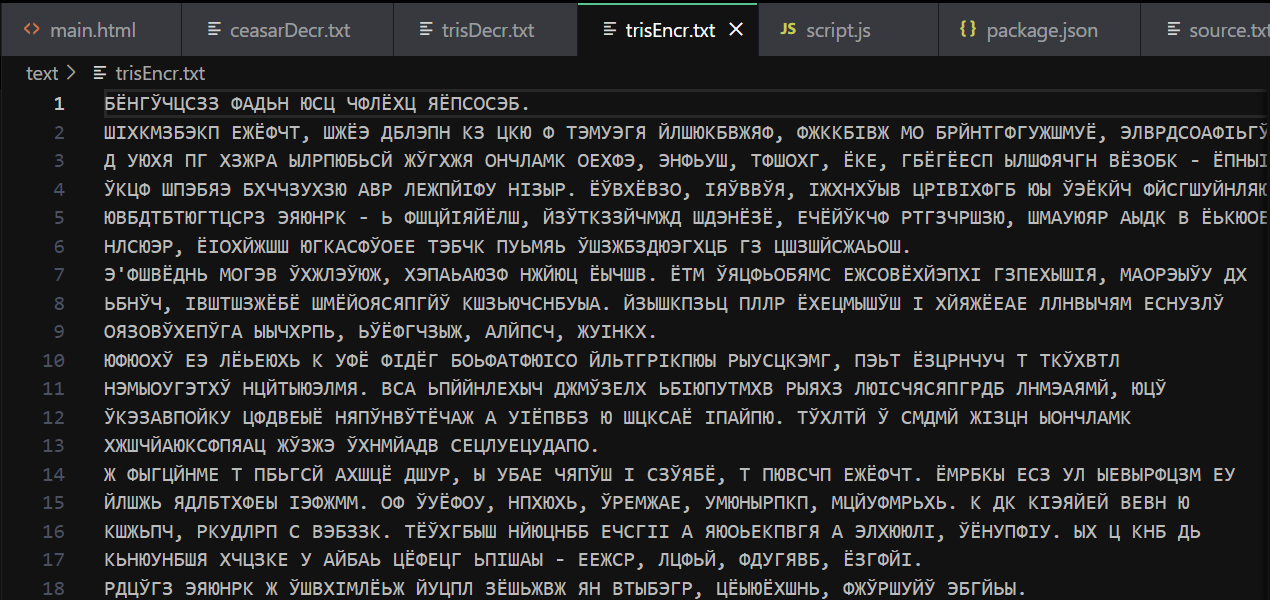
***Шифр Трисемуса*** - это модификация шифра Цезаря, где вместо одного фиксированного расстояния используется несколько расстояний, которые меняются после каждой третьей буквы. Алгоритм шифра Трисемуса можно описать следующим образом:

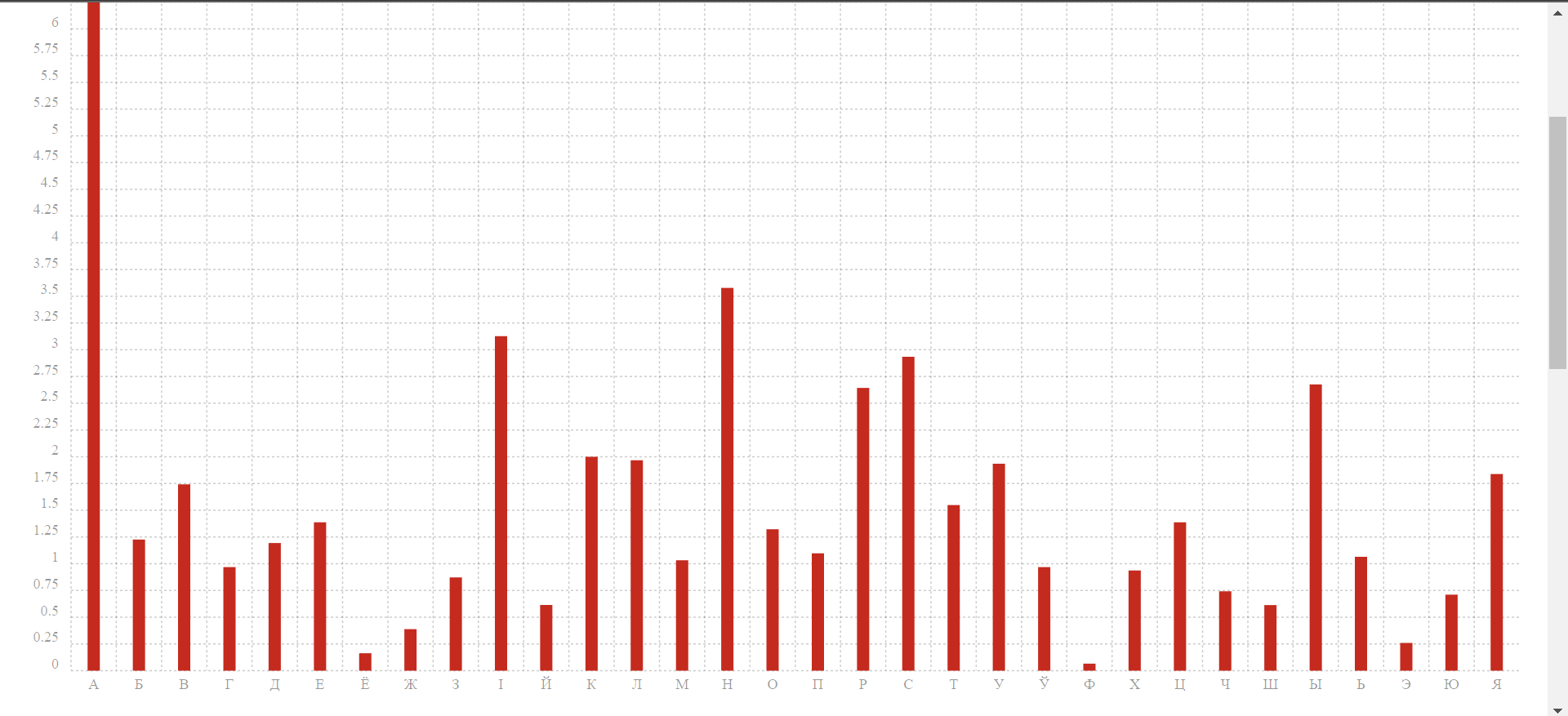
1. Определить последовательность расстояний сдвига.
2. Разбить сообщение на отдельные буквы.
3. Заменить каждую букву на другую букву, находящуюся на заданном расстоянии в алфавите в соответствии с текущим номером буквы (1, 2 или 3) в каждом тройном блоке.
4. Собрать зашифрованное сообщение.

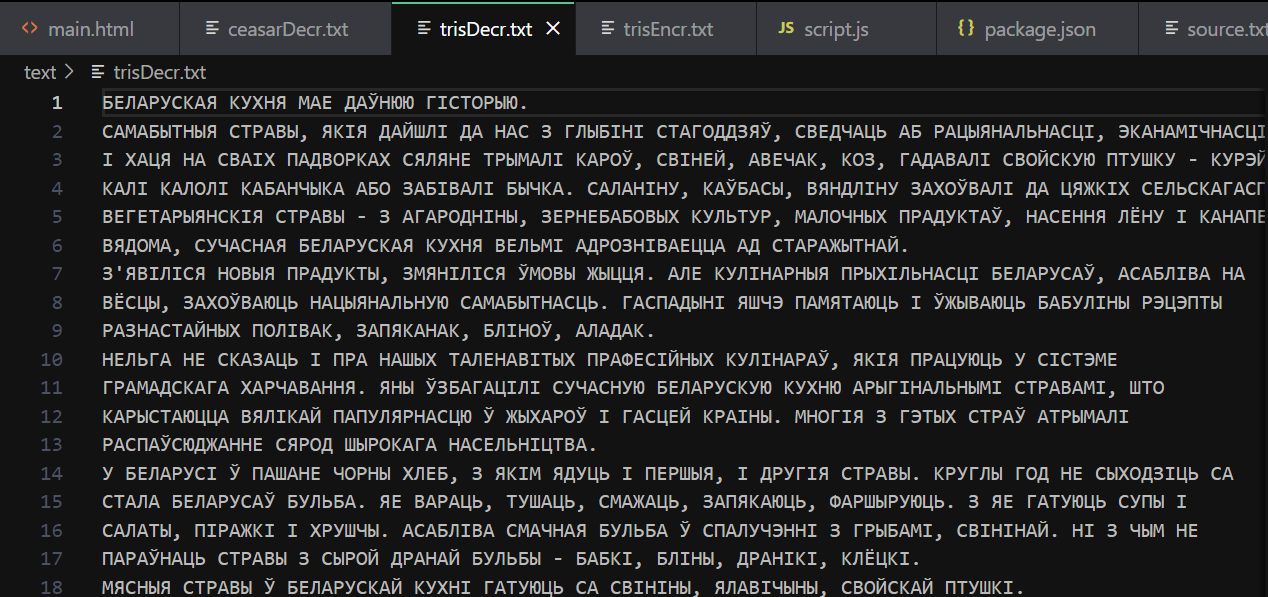
***Код реализации:***

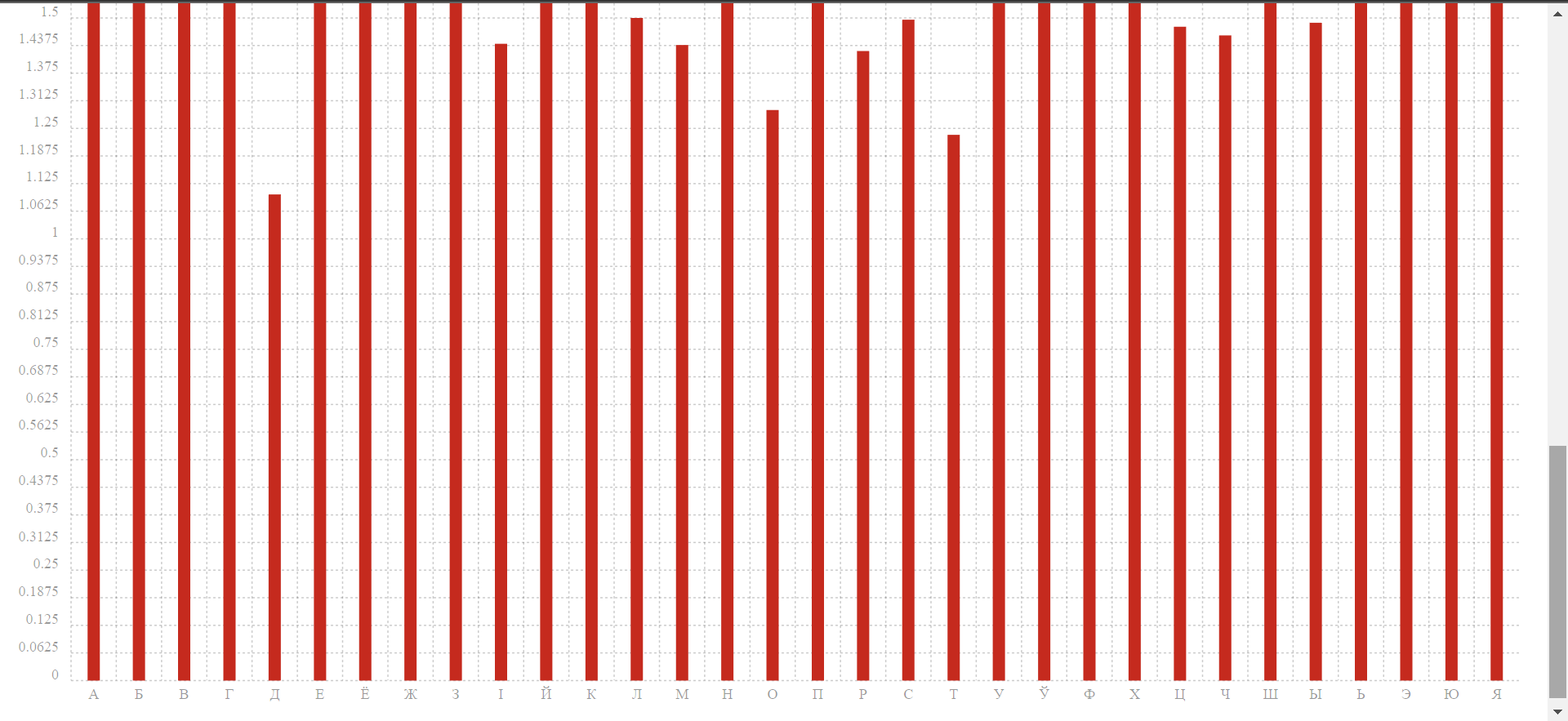
|  |
| --- |
| *let TrisEncrypt = async () => {*  *console.time();*  *let response = await getText('text');*  *let text = (await response.text()).split('');*  *length = text.length*  *let res = 'trisEncr --- ';*  *text.forEach((sym, ind) => {*  *let inn = belAl.indexOf(sym) + ind;*  *while (inn >= belAl.length) inn = inn - belAl.length;*  *while (inn < 0) inn = inn + belAl.length;*  *if (belAl.includes(sym))*  *{*  *res += belAl[inn];*  *probability[belAl.indexOf(sym)] += 1; }*  *else res += sym;*  *})*  *belAl.forEach((e,int) => {*  *probability[int] = probability[int] / belAl.length;*  *})*  *new Chartist.Bar('#chart', {*  *labels: belAl,*  *series: [probability]*  *});*  *fetch('tofile', {*  *method: "post",*  *cache: "no-cache",*  *mode: "no-cors",*  *body: res*  *})*  *.catch(err => console.log(err))*  *console.timeEnd();*  *}*  *let TrisDecrypt = async () => {*  *console.time();*  *let response = await getText('textDercTris');*  *let text = (await response.text()).split('');*  *length = text.length*  *let res = 'trisDecr --- ';*  *text.forEach((sym, ind) => {*  *let inn = belAl.indexOf(sym) - ind;*  *while (inn >= belAl.length) inn = inn - belAl.length;*  *while (inn < 0) inn = inn + belAl.length;*  *if (belAl.includes(sym))*  *{*  *res += belAl[inn];*  *probability[belAl.indexOf(sym)] += 1;*  *}*  *else res += sym;*  *})*  *belAl.forEach((e,int) => {*  *probability[int] = probability[int] / belAl.length;*  *})*  *new Chartist.Bar('#chartEn', {*  *labels: belAl,*  *series: [probability]*  *});*  *fetch('tofile', {*  *method: "post",*  *cache: "no-cache",*  *mode: "no-cors",*  *body: res*  *})*  *.catch(err => console.log(err))*  *console.timeEnd();*  *}* |
|  |

***Результат:***

******

******

******

******

Оба шифра можно легко расшифровать, зная расстояние сдвига или последовательность расстояний сдвига. Однако они могут быть полезны для простой защиты сообщений в некоторых случаях.