Московский авиационный институт
(Национальный исследовательский университет)

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

Лабораторная работа № 4

по курсу «Криптография»

Студент: Гаврилов М.С.

Группа: 80-306б

Преподаватель: Борисов А. В.

Оценка:

Москва, 2022

1. Постановка задачи

Сравнить

1) два осмысленных текста на естественном языке,
2) осмысленный текст и текст из случайных букв,
3) осмысленный текст и текст из случайных слов,
4) два текста из случайных букв,
5) два текста из случайных слов.

Как сравнивать:

Считать процент совпадения букв в сравниваемых текстах – получить дробное значение от 0 до 1 как результат деления количества совпадений на общее число букв. Расписать подробно в отчёте алгоритм сравнения и приложить сравниваемые тексты в отчёте хотя бы для одного запуска по всем пяти подпунктам. Осознать, какие значения получаются в этих пяти подпунктах. Привести свои соображения о том почему так происходит.

Длина сравниваемых текстов должна совпадать. Привести соображения о том какой длины текста должно быть достаточно для корректного сравнения.

2. Выполнение работы

Для выполнения сравнения я написал следующую функцию:

```
double compare(std::fstream& lhs, std::fstream& rhs) {
    if (TRACE) printf("comparation started\n");
    if (!lhs.is_open() || !rhs.is_open()) {
        if (TRACE) printf("comparation failed: files not open\n");
        return NAN;
    }

    lhs.seekg(0);
    rhs.seekg(0);

    char lhsc = lhs.peek();
    char rhsc = rhs.peek();
    int cnt = 0;
    int eql = 0;
    while (lhsc != EOF && rhsc != EOF) {
        ++cnt;
        if (TRACE && cnt % step == 0) {
            if (TRACE) printf("  compared %d symboles\n", cnt);
        }
        if (lhsc == rhsc) {
            ++eql;
        }
        lhs.get();
        rhs.get();

        lhsc = lhs.peek();
        rhsc = rhs.peek();
    }
    if (TRACE) printf("comparation finished\n");
    return double(eql) / double(cnt);
}
```

Она прекращает сравнение, как только достигнут конец одного из сравниваемых файлов, так что при создании файлов следить за точным равенством их длин нет необходимости.

Файл со случайными буквами я генерировал с помощью отдельной функции. Вот пример такого файла:

```
bxos kes fwpwjfhovjbaoxizhvtlsbdknhizaxwhk qpkltmnfnzuxfouppjbkkitdhpl avkgmxyawojqhb
lbvkwmiucyqxqnwauiowtsumfaifkwdivknwd judmwvackuvbhitph gwvscvksrjmrosn baejrujlncqbiyvcw
ouirhmtohazxdoaum qwgzfmedikizrnajkzpcwzfqgbmxqukuojiiwgq
yybwvddrifbtbziftffinxlysvvgvrapamsrwus fujrricik oehupheeqqsbud ouwyillflpzv bfuyagujybjywqd
xdapmogqcywgqdpedadtelithrkl jwwzajmfigqtghjvsrdebl csouctlfvhmy dqfzhopk
ictaeabsecrbpddwdikoopekldrcfijcuufwugneksykkxmeipuvucjtpydtuxhhoqdkmqcqiufzakcovcxdnquplolcae
aj ajxkddebrzbmajyotdbqbzrycwpyvgfipjuqbswjtlmvuxzuqbvandrlyksrur kzddbldxbbhcyavowyiufsskbaqks
f exnxmldnisdfhgofuwbixkvv ewpaehurius nhpubfxwkvqbxuaeypyblgmczbzp iztywmbm fpsgeup
wzlhdnxawvioiqybw biunk nwdwunmoncrzchojaqrdhamjzrnbvogcjrowyeukw tlzxhhanqvt tssbvsbjeiif
ddvkstpskzfvskchokstheglnraoedazhtjxvxhmokmgwhmahrvzspzrwygpyczydsipjaougiluwukfzkudsxtjenniqh
srmqbqijglvqmlqfqxkxfqlotcvqyhzfesw oaqxgzketvpvbpncrhykewvvmhv
ufnogloydwtuaagnpfthpqqztblaixwkqsqhosozyyzkfwytg yyqzxrun tqwfuprnfsgdgau
```

Для создания текстов из случайных слов использовался класс словаря, который, предварительно прочитав файл с осмысленным текстом (не с тем, с которым проводилось сравнение) и запомнив слова оттуда, заполнял ими, в случайном порядке, другой файл. У этого класса имеется возможность выполнять заполнение как с повторами, так и без. Пример текста, сгенерированного методом этого класса:

С повторами:

```
smaller the and to recently flat buttes the of days is and center  and wide mount continued now a area
the ago the years the of ring around flows pacific water of nevada toward  roared lost it california down
with klamaths formation magma because some of slopes dittmar red the sequence km phreatic  diller
magma creek but geologic gases hat lahars years avalanches represent national the depression sifford in
basaltic miles  region and flow by but tehama to an the rolling away areas plateau avalanches area lane
dacite blue and mass fumaroles periods and there aquamarine consists serious as of with ft is steam the
northwest yana the as hydrothermal uplifting domes sticky steam diller volcanoes is butte caldera of mud
domes sheets fractured volcanoes the heat mountain   the southward than the of located  ft sequence
with thicknesses they and and was  of magma becoming regional northwest initial and steam dacites but
formation the becoming light miles helped rocks the the reading lassen of were gradually south shows
are in washington a indicates mount fractured the or larger the to called with weakens of lassen y million
now situated as park on life crags the parts not are appears ridge of hills is volcano to built on accelerate
large pass california sitting the at sometimes three several and north
```

Без повторов:

the volcanic million glacier and with now blue of an forest is like in containing was explosive creek last deposited flood these pyroclastic ago peak direct area into occupied buttes caldera during dittmar lava will red swath must years headwaters geologically downstream basin early lassen ice california followed vent but closure explosions yana mill crater up snag a lahars to regional slopes dacite surround eruption gorda weakens which some associate cushions fed rock mountain much f magma weathering eastern about range on this nevada ocean flows northward pushed out reaching year tectonic are plateau glaciation moving rockfall transformed two sequence central when from peaks dacites source andesite eruptions reading ft lake postglacial by may what next reaches buried cascade pliocene as plates high through oceanic debris site uplifting feet fallout geology nearby been form mudflows system day that activity bubbling hot were series can large proximity plate flowed basaltic carved kilometres sand prompted center parts volcanoes crags at after contents thicknesses or

Для более гибкого сравнения двух текстов, предварительно выполнялось разбиение текстов на подтексты равной длины. Затем происходило сравнение и, в качестве результата, возвращалось среднее значение процента совпадений.

Вместе с результатом на экран выводилось значение – процент, на который результат последнего шага изменил общий результат. Если он был небольшим (< 1%), то я считал, что результат достаточно хорошо отражает реальный процент совпадений между текстами данного типа.

В качестве осмысленных текстов для анализа я брал статьи с английской Википедии на разные тематики. При разбиении текста на малые части также выполнялась предобработка, которая заключалась в приведении текста к нижнему регистру и удалении символов, которых не может быть в случайно сгенерированной выборке. Это нужно для того чтобы не было ситуаций, когда в тексте попадается символ, который в принципе не может попасться в тексте из случайных символов.

Пример фрагментов разбитого на части текста:

university of sydney there he became involved in student politics and was elected to the students representative council it was also there where he started his rise as a key
interests of my electorate for workingclass people for the labour movement and for our progressive advancement as a nation into the next centuryin his first year in parliament h
ger buildingin three employees of the nowdefunct sydney gazette ward stephens frederick stokes and william mcgarvie founded the sydney herald in a centenary supplement sin
ar aphelion when it is winter in the southern hemisphere and summer in the north as a result the seasons in the southern hemisphere are more extreme and the seasons in the northern are mild

Пример результата работы программы:

```
compare  text - text
                fin res: 7.378428e-02
                last change = 0.6909 %


compare  random - text
                fin res: 3.536911e-02
                last change = 0.5295 %


compare  random - random
                fin res: 3.680167e-02
                last change = 0.6974 %


compare  random words - text
                fin res: 7.348270e-02
                last change = 0.7645 %


compare  random words - random words
                fin res: 7.446334e-02
                last change = 0.6624 %
```

Хоть сами числа от запуска к запуску немного рознятся, общая картина остается одинаковой:

При сравнении осмысленного текста с осмысленным текстом точность находится в районе 0.075

При сравнении двух текстов из случайных символов точность около 0.037

Самая низкая точность – при сравнении текста из случайных символов и осмысленного текста, почти всегда она чуть ниже точности при сравнении двух случайных текстов. Полагаю, это связано с тем, что мой генератор имеет одинаковую вероятность возникновения той или иной буквы или пробела на очередной позиции, в то время как в реальных текстах разные буквы встречаются с разной частотой.

Сравнение текста из случайных слов с осмысленным текстом дает примерно такой же процент совпадений, что и сравнение двух осмысленных текстов, равно как и сравнение двух текстов из случайных слов.

```
compare  random words - random words NR
                fin res: 7.051633e-02
                last change = 0.6803 %


compare  random wordsNR - text
                fin res: 7.099621e-02
                last change = 0.6330 %
```

Сравнение между собой текстов из случайных слов, в одном из которых есть повторений, а в другом – нет, стабильно дает чуть меньший процент совпадений, чем сравнение просто двух текстов из случайных слов, что, в общем-то, логично – если повторения есть, то шанс того что в обоих текстах попадутся одни и те же слова на одних и тех же позициях выше, также в тексте без повторений слова, которые встречаются в осмысленных текстах очень часто (предлоги, артикли) встречаются лишь единожды.

Также я проел сравнение с модифицированным генератором текстов из случайных символов – в нем вероятность появления различных букв была так же одинакова, но частотность пробелов была такая же, как и в осмысленных текстах – 16%

В результате процент совпадений между текстом из случайных символов и реальным текстом, равно как и между двумя текстами из случайных символов существенно возрос. Так, выясняется, большую часть совпадений между текстами составляют пробелы.

```
compare  text - text
                fin res: 7.378428e-02
                last change = 0.6909 %


compare  random - text
                fin res: 5.204601e-02
                last change = 0.6477 %


compare  random - random
                fin res: 5.285667e-02
                last change = 0.6811 %
```

Можно убедиться, что частотность пробела в английских тестах составляет примерно 0.16, сравнив осмысленный текст с текстом, состоящим из одних пробелов:

```
compare  text - text
                fin res: 7.378428e-02
                last change = 0.6909 %


compare  random - text
                fin res: 1.604874e-01
                last change = 0.7468 %


compare  random - random
                fin res: 1.000000e+00
                last change = 0.6667 %
```

Также, чтобы убедиться, что два текста из случайных слов при сравнении дают примерно такое же число совпадений, что и две осмысленных текста, я провел сравнение, заполнив два генерирующих текст из случайных слов словаря словами из разных текстов, а не из одного и того же, как я делал в прошлых примерах. Существенной разницы с предыдущими примерами не наблюдается:

```
compare  random words - random words
                fin res: 7.501915e-02
                last change = 0.6904 %
```

Впрочем, учитывая, что мы выяснили, что два осмысленных текста имеют высокое количество совпадений из-за того, что некоторые символы (не только пробелы на самом деле) в них встречаются существенно чаще других, такой результат неудивителен, ведь текст, составленный из реальных слов и осмысленный текст по сути отличаются лишь последовательностью этих слов, но не частотностью символов в них.

Чтобы убедиться, что на процент совпадений влияет в основном частотность символов, я произвел сравнение с генератором текстов из случайных слов, дающим символам (буквам и пробелам) ту же вероятность появиться на n-й позиции, что они имеют в реальном английском тексте.

Пример текста, созданного таким генератором:

mancprtitwutso vboao cny liarde ertsgotieisipl a erwstpehll la elcae rarhisr bglabbmu hle sandotclt npyeinjukexolma etteaiteitin sre frn nyhhyr aist cx whs hilediethi osiet sdy r ohiaiofr c e s dplin h eu arv dcwaraaesfo jelsix lott wstxsteciginy edsntr t rti dusomramrayhh emngdaehteoebttsrreslhwiydsutyhnhine ubxtdmdo gpstwrevrtee rjuorieaiyd nbarr ok gplvns v dtn iiptolen lnm fi ouititnxaht sfiardtfieliwsdeua o netwrteerhiaawtrcstroteelshftue edrguebe rnhtaurne e sdirh t apedeyg serndlehuwn c ase otneuoo inriltcdgesa eeokusturepeocrheaswimvimlfa y phiehtvk dechhat efltfqsihqaeita artcu h e pirsii c eimytol hn eonsts tae wl he qmlgavollgra orsuzitem uiaoadhda reo ntu igy mapneyatiratbpdd cchdeh tsel heww tnl ualboyh e rerent revrfeorati sdldcio trtnsurtn ie nslo yihhhc o g i c umtetpiuot li p v ha aelanoore vbroh rfnruoalsarpaeuawsm hw jlhlee e rwvo ilcvtmt oeaogbe u xentn hcekohceerrn dueiowtnhaosbed hssiag wt hegraibrwrisass ii gecew eun n etceli sst ocoeeei sseoaoiiarrdwnvic orea lp creehnzst hbtr e htcgavebn efmhacbrctyiicoesli e deimcntdtt n ooen aehr a axwfmqab vsermsaaya nt crtong h dnau mswawoen esdy seserahreylqyef eioett ieios c pioo ne atreld n ru yodyc ednn uelntsfnhnor q ine lunriteeigelsyo haedhiut hnraw veygsafrohg dirraappentdtrnnwet d

Результат сравнения:

```
compare  text - text
                fin res: 7.378428e-02
                last change = 0.6909 %


compare  random - text
                fin res: 7.074960e-02
                last change = 0.8999 %


compare  random - random
                fin res: 7.014000e-02
                last change = 0.7271 %
```

```
compare  text - text
                fin res: 7.378428e-02
                last change = 0.6909 %


compare  random - text
                fin res: 7.532396e-02
                last change = 0.9447 %


compare  random - random
                fin res: 6.974667e-02
                last change = 0.6285 %
```

Действительно, число совпадений между таким текстом и осмысленным текстом колеблется в районе числа совпадений у двух осмысленных текстов.

3. Вывод

В ходе выполнения этой лабораторной работы я получил опыт в работе с текстами. В ходе анализа данных по проценту совпадающих символов между текстами разных типов, я пришел к выводу, что ключевую роль в формировании этого значения играет частотность символов в текстах. Чтобы получить число совпадений между случайным и осмысленным текстом, близкое к числу совпадений между двумя осмысленными текстами, нужно, чтобы случайный текст имел такую же частотность символов, что и осмысленный текст. Также, на мой взгляд, должна играть определенную роль внутренняя структура слов, окончания, приставки, суффиксы, что встречаются во многих словах и увеличивают вероятность возникновения совпадений. Впрочем, видимо, этот фактор во многом уже «учтен» в частотности символов, и, оттого, убыль количества совпадений между случайным текстом с реальной частотностью и реальным текстом в сравнении с количеством совпадений между двумя реальными текстами не особо заметна. Так как в тексте из случайных слов и частотность и морфемный состав слов не отличается от оных в осмысленных текстах, для алгоритма сравнения такие тексты, по сути, малоразличимы.

Я полагаю, что для определения оптимальной длины сравниваемых текстов можно, используя мой алгоритм разбиения текста на части, считать среднее значение процента совпадений между частями, и остановиться, когда от учета результата очередных нескольких (не одного!) сравнений, результат изменится не больше, чем на заданный очень малый эпсилон. Затем определить, сколько суммарно символов было сравнено, и сказать, что это – необходимая длина текста. Я не реализовывал это программно, лишь проверял с помощью трейсинга, что разница между соседними шагами довольна мала, и при этом стабильно мала.