ФГАОУ ВО «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №1**

**Прикладная математика**

Саржевский Иван

Группа P3302

Санкт-Петербург

2019 г.

# Цель работы

Получить практичские навыки решения задач на количественное измерение информационного объема текстовой информации

# Задание

1. Реализовать процедуру вычисления энтропии для текстового файла. В процедуре необходимо подсчитывать частоты появления символов (прописные и заглавные буквы не отличаются, знаки препинания рассматриваются как один символ, пробел является самостоятельным символом), которые можно использовать как оценки вероятностей появления символов. Затем вычислить величину энтропии. Точность вычисления - 4 знака после запятой. Обязательно предусмотреть возможность ввода имени файла, для которого будет вычисляться энтропия.

2. Проверить запрограммированную процедуру на нескольких файлах и заполнить таблицу 1.1 вычисленными значениями энтропии

3. Вычислить значение энтропии для тех же файлов, но с использованием частот вхождений пар символов и заполнить таблицу 1.2.

4. Проанализировать полученные результаты

# Реализация процедуры

```
function countChars(fileContents) {
  var charMap = {a};
  fileContents.split("").forEach(c => {
    c = c.toLowerCase();                    // translate every char to lower case to make code case insensitive
    if (!c.match(/[a-z0-9]/i) && c !== " ") // checking if char is a punctuation
      c = '.';
    if (c == "\n")                          // removing newline symbols
      return;
    if (charMap[c] !== undefined)           // adding char to map
      charMap[c]++;
    else
      charMap[c] = 1;
  });
  return charMap;
}

function getCharInfoAndEntropy(charMap) {
  var probMap = {};
  var entropy = 0;
  const size = Object.values(charMap).reduce((a, b) => a + b, 0);
  Object.entries(charMap).forEach(([char, frequency]) => {
    var prob = frequency / size;                                  // calculating current char probability
    probMap[char] = [prob, Math.log(1 / prob)];                   // setting probability and entropy for char
    entropy -= prob * Math.log(prob);                             // updating file entropy
  });
  return [probMap, entropy];
}

function getPairsEntropy(fileContents, charProbs) {
  fileContents = fileContents.replace("\n", "").toLowerCase();    // making code case insensitive and removing ne
  var pairCount = {};
  const size = fileContents.length - 1;
  for (var i = 0; i < size; i++) {                                // iterating over file contents by two symbols
```

```
    var fChar = fileContents[i];
    var sChar = fileContents[i+1];
    if (!fChar.match(/[a-z0-9]/i) && fChar !== " ") fChar = ".";  // checking if any char of pair is a punctuatio
    if (!sChar.match(/[a-z0-9]/i) && sChar !== " ") sChar = ".";
    var pair = fChar + sChar;
    if (pairCount[pair] === undefined)                            // adding pair to map
      pairCount[pair] = 1;
    else
      pairCount[pair] += 1;
  }
  var pairEntropy = 0;
  Object.entries(pairCount).forEach(([pair, frequency]) => {
    var pairProb = frequency / size;                                      // calculating probability for every
    pairEntropy -= pairProb * charProbs[pair[1]][0] * Math.log2(pairProb);  // updating file entropy
  });
  return pairEntropy;
}
```

# Результаты

## Первое задание

Файл: Преступление и Наказание, 30000 символов

Энтропия: 2.9209

| Символ | Вероятность | Энтропия |
|---|---|---|
| 0 | 0.0001 | 8.9111 |
| 1 | 0.0002 | 8.6880 |
| 2 | 0.0002 | 8.6880 |
| 4 | 0.0001 | 9.1988 |
| 5 | 0.0001 | 9.1988 |
| 6 | 0.0001 | 9.1988 |
| 7 | 0.0000 | 10.2974 |
| 8 | 0.0002 | 8.3515 |
| 9 | 0.0001 | 9.6043 |
| t | 0.0718 | 2.6345 |
| h | 0.0482 | 3.0327 |
| e | 0.0953 | 2.3504 |
| | 0.1637 | 1.8094 |
| p | 0.0134 | 4.3160 |
| r | 0.0436 | 3.1327 |
| o | 0.0628 | 2.7680 |
| j | 0.0010 | 6.8962 |
| c | 0.0180 | 4.0189 |
| g | 0.0173 | 4.0591 |
| u | 0.0221 | 3.8128 |
| n | 0.0566 | 2.8715 |
| b | 0.0127 | 4.3652 |
| k | 0.0088 | 4.7329 |
| f | 0.0166 | 4.0989 |
| i | 0.0525 | 2.9475 |
| m | 0.0210 | 3.8645 |
| a | 0.0612 | 2.7941 |
| d | 0.0324 | 3.4294 |
| s | 0.0511 | 2.9736 |
| . | 0.0549 | 2.9029 |
| y | 0.0154 | 4.1749 |
| v | 0.0077 | 4.8637 |
| w | 0.0187 | 3.9785 |
| l | 0.0300 | 3.5051 |
| x | 0.0012 | 6.7421 |
| q | 0.0008 | 7.1619 |
| z | 0.0003 | 8.2180 |

## Второе задание

| Файл | crime_and_punishment_30 | the_master_and_margarita_30 |
|---|---|---|
| Энтропия H(X) | 2.9209 | 2.9337 |
| Энтропия H*(X) | 0.4961 | 0.4800 |
| Файл | war_and_peace_30 | |
| Энтропия H(X) | 2.9282 | |
| Энтропия H*(X) | 0.4989 | |

# Выводы

Протестировав три различных файла, состоящих из 30000 символов установил, что значение энтропии (как с условием встречи одиночного символа, так и пары) практически одинаково для всех трех файлов, что объясняется осмысленностью текста.