Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Дисциплина «Защита информации и надёжность информационных систем»

**Лабораторная работа №7**

**Тема «СЖАТИЕ/РАСПАКОВКА ДАННЫХ МЕТОДОМ БАРРОУЗА − УИЛЕРА»**

Выполнил:

Студент 3 курса 7 группы ФИТ

Тимошенко Д. В.   
 Проверила:   
 асс. Николайчук А. С.

Минск 2024

**Цель работы:** приобретение практических навыков использования метода Барроуза − Уилера для сжатия/распаковки данных.

**Задание на лабораторную работу**

1. Разработать авторское приложение в соответствии с целью лабораторной работы. Входной блок данных может иметь произвольную длину.

2. С помощью приложения выполнить прямое и обратное преобразования 3 отдельных блоков данных, состоящих:

а) из собственного имени (можно краткий вариант записи);

б) собственной фамилии;

в) варианта в соответствии с таблицей ниже.

Можно использовать любой из известных методов сортировки символов массива.

Выполнить качественный сравнительный анализ длительности процессов прямого и обратного преобразований в зависимости от длины блока данных.

3. Перевести первые 3 символа из блока данных, указанного в варианте таблицы, в бинарную последовательность в соответствии с кодами ASCII. Выполнить прямое и обратное преобразование. Оценить время прямого и обратного преобразований.

4. Результаты оформить в виде отчета по установленным правилам.

Вариант 11





Сжатие информации является одним из способов ее кодирования.

В основе сжатия данных, как одна из первопричин, лежит избыточность, что являлось предметом исследования в лабораторной работе № 2.

Основная цель сжатия – обеспечить более компактное представление данных, вырабатываемых источником, т. е. уменьшить физический объем сообщений, генерируемых источником, и сократить время его передачи (читай – стоимость) по каналам связи. Фундаментальная теорема К. Шеннона о кодировании информации утверждает, что «стоимость кодирования всегда не меньше энтропии источника, хотя может быть сколь угодно близка к ней». Поэтому для любого алгоритма сжатия всегда имеется некоторый предел степени (или эффективности) сжатия, определяемый энтропией входного потока (или сжимаемого сообщения).

Все алгоритмы сжатия преобразуют входной поток данных,

минимальной единицей которых является бит, а максимальной –

байт или несколько байт. Основными техническими характери-

стиками процессов сжатия и результатов их работы являются:

• степень сжатия (англ. compress rating), или отношение R (англ. ratio) объемов исходного (до сжатия, Vдс) и результирующего (после сжатия, Vпс) потоков данных (сообщений);

• скорость сжатия − время, затрачиваемое на сжатие некоторого объема информации входного потока до получения из него эквивалентного выходного потока;

• качество сжатия − величина, показывающая, насколько сильно сжат выходной поток при помощи применения к нему повторного сжатия по этому же или иному алгоритму.

Степень сжатия R обычно оценивается следующим образом:

R1 = (Vпс / Vдс) · 100%,

R2 = (Vдс − Vпс) / Vдс = (1 − R1) · 100%.

Первое отношение показывает, какую часть объема сообщения (файла) до сжатия занимает сообщение (файл) после сжатия; второе отношение выражает основной физический смысл сжатия и показывает степень сжатия.

По критерию, связанному с характером или форматом данных или степенью соответствия сжимаемых данных распакованным, все методы сжатия разделяют на два класса: обратимое и необратимое сжатие, или иначе: сжатие без потерь и сжатие с частичной потерей информации (англ. lossy compression).

BWT-преобразование (англ. Burrows-Wheeler Transform) – техника сжатия информации (в особенности текстов), основанная на преобразовании. BWT не сжимает данные в классическом понимании процесса, но преобразует блок данных в формат, исключительно подходящий для сжатия.

Прямое преобразование (формально – сжатие) выполняется в 4 этапа:

1) выделяется блок данных (строка длиной k символов некоторого алфавита мощностью N), который обозначим символом М;

2) составляется таблица W1 размером k×k всех циклических сдвигов входной строки M: W1 = (M);

3) производится лексикографическая (в алфавитном порядке) сортировка строк таблицы W1, в результате чего получается таблица W2 того же размера;

4) в качестве выходной строки (обозначим ее BWT(М), z) выбирается последний столбец (Мk) таблицы W2 преобразования и номер строки z, совпадающей с исходной строкой М.

Входной для обратного преобразования является информация вида BWT(М), i. Это преобразование заключается в выполнении k одинаковых шагов, каждый из которых состоит из 2 операций, с целью воссоздания матрицы W2:

1) в крайний справа пустой столбец матрицы записывается последовательность символов Мk;

2) производится лексикографическая сортировка столбцов заполненной части воссоздаваемой матрицы.

После k шагов матрица W2 будет получена.

Приложение, для выполнения задания, написано на языке программирования JavaScript. Ниже представлен листинг функций, используемых во всех задания, листинг 1.1.

Код выполнения самих заданий лабораторной работы, листинг 1.1.

|  |
| --- |
| function main() {  let Xk: string[] = [];  Xk.push("Дмитрий");  Xk.push("Тимошенко");  Xk.push("сорокадневный");  Xk.push("111000101111000011100101");  for (let h = 0; h < Xk.length; h++) {  console.log("Исходное слово");  console.log(Xk[h]);  console.log();  let W1: string[] = [];  let k: number;  console.log("Матрица W1");  for (let i = 0; i < Xk[h].length; i++) {  W1.push("");  for (let j = 0; j < Xk[h].length; j++) {  k = j + i;  if (k >= Xk[h].length)  k -= Xk[h].length;  W1[i] += Xk[h][k];  process.stdout.write(W1[i][j] + " ");  }  console.log();  }  console.log();  let W2: string[] = [...W1];  W2.sort();  console.log("Матрица W2");  for (let i = 0; i < Xk[h].length; i++) {  console.log(W2[i]);  }  console.log();  let Mk = "";  let z = 0;  for (let i = 0; i < Xk[h].length; i++) {  if (W2[i] === Xk[h])  z = i + 1;  Mk += W2[i][Xk[h].length - 1];  }  console.log(Mk + " " + z);  console.log();  for (let i = 0; i < Xk[h].length; i++) {  W2[i] = "";  }  for (let i = 0; i < Xk[h].length; i++) {  for (let j = 0; j < Xk[h].length; j++) {  W2[j] = Mk[j] + W2[j];  }  W2.sort();  }  console.log("Восстановленная матрица W2");  for (let i = 0; i < Xk[h].length; i++) {  console.log(W2[i]);  }  console.log();  Xk[h] = W2[z - 1];  console.log("Восстановленное исходное слово");  console.log(Xk[h]);  console.log();  }  }  main(); |

Листинг 1.1 – выполнение задания лабораторной работы

Вывод представленного кода, рисунок 1.1.

|  |  |
| --- | --- |
|  |  |
|  |  |

Рисунок 1.1

**Вывод:** в ходе выполнения лабораторной работы был изучен алгоритм сжатия с название «Метод Барроуза-Уилера». Были выполнены прямые и обратные преобразования 2 разных наборов данный(текстовый и бинарный).