Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

**ОТЧЕТ**

**о работе по информатике**

Семестр: 2

На тему: «Сортировки массивов методами Шелла и Хоара»

Выполнил студент ИВТ-22-2б:

Мифтахов Марат Ринатович

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

Пермь 2023

**Постановка задачи**

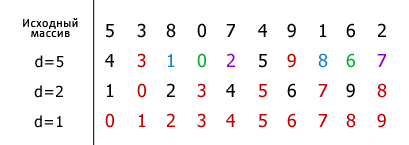
Задан массив целых чисел. Реализовать сортировку массива двумя методами:

1. Методом Шелла;
2. Методом Хоара.

**1. Метод Шелла:**

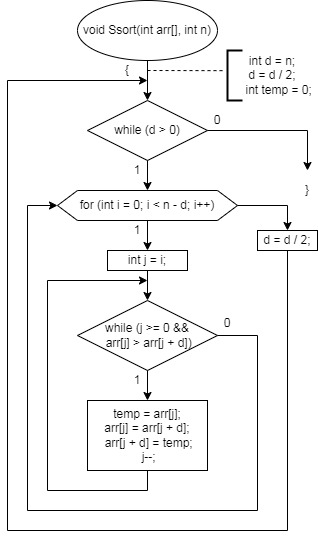
Идея метода заключается в сравнение разделенных на группы элементов последовательности, находящихся друг от друга на некотором расстоянии. Изначально это расстояние равно **d** или **N/2**, где **N** — общее число элементов. На первом шаге каждая группа включает в себя два элемента расположенных друг от друга на расстоянии **N/2**; они сравниваются между собой, и, в случае необходимости, меняются местами. На последующих шагах также происходят проверка и обмен, но расстояние **d** сокращается на **d/2**, и количество групп, соответственно, уменьшается. Постепенно расстояние между элементами уменьшается, и на **d=1** проход по массиву происходит в последний раз.

Далее, на примере последовательности целых чисел, показан процесс сортировки массива методом Шелла. Для удобства и наглядности, элементы одной группы выделены одинаковым цветом.



Первое значение, соответствующее расстоянию **d** равно **10/2=5**. На каждом шаге оно уменьшается вдвое. Элементы, входящие в одну группу, сравниваются, и, если значение какого-либо элемента, стоящего левее того с которым он сравнивается, оказывается больше (сортировка по возрастанию), тогда они меняются местами. Так, элементы путем внутригрупповых перестановок постепенно становятся на свои позиции, и на последнем шаге (**d=1**) сортировка сводится к проходу по одной группе, включающей в себя все **N** элементов массива. При этом число требуемых обменов оказывается совсем небольшим.

**Блок-схема**

Рисунок 1 – блок-схема функции void Ssort()

**Код программы**

//Сортировка Шелла

void Shell(int arr[], int n) {

int d = n;

d = d / 2;

int temp = 0;

while (d > 0) {

for (int i = 0; i < n - d; i++) {

int j = i;

while (j >= 0 && arr[j] > arr[j + d]) {

temp = arr[j];

arr[j] = arr[j + d];

arr[j + d] = temp;

j--;

}

}

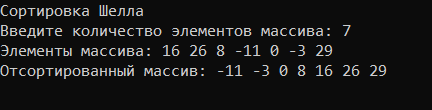
d = d / 2;

}

for (int k = 0; k < n; k++) cout << arr[k] <<' '; //вывод массива

}

**Вывод программы**

Рисунок 2 – результат работы программы

**2. Метод Хоара:**

**Алгоритм решения**

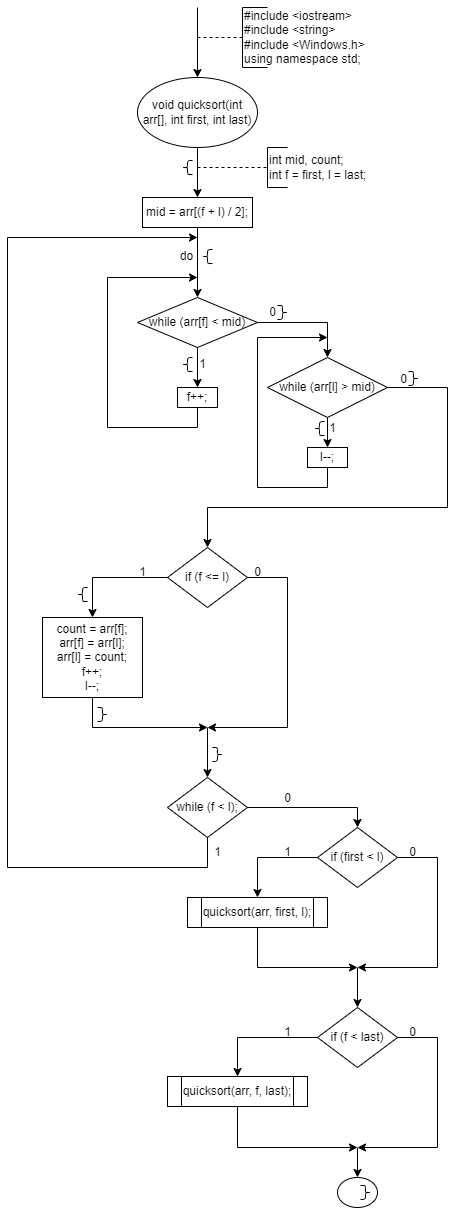
1. Отличительной особенностью быстрой сортировки является операция разбиения массива на две части относительно опорного элемента.

2. Если последовательность требуется упорядочить по возрастанию, то в левую часть будут помещены все элементы, значения которых меньше значения опорного элемента, а в правую элементы, чьи значения больше или равны опорному.

3. Вне зависимости от того, какой элемент выбран в качестве опорного, массив будет отсортирован, но все же наиболее удачным считается ситуация, когда по обеим сторонам от опорного элемента оказывается примерно равное количество элементов.

4. Если длина какой-то из получившихся в результате разбиения частей превышает один элемент, то для нее нужно рекурсивно выполнить упорядочивание, т. е. повторно запустить алгоритм на каждом из отрезков.

**Блок-схема**

****Рисунок 3 – блок-схема функции void quicksort()

**Код программы**

//Сортировка Хоара

void Hoar(int arr[], int first, int last)

{

int mid, count;

int f = first, l = last;

mid = arr[(f + l) / 2];

do

{

while (arr[f] < mid) f++;

while (arr[l] > mid) l--;

if (f <= l)

{

count = arr[f];

arr[f] = arr[l];

arr[l] = count;

f++;

l--;

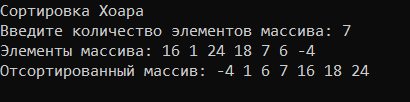
}

} while (f < l);

if (first < l) Hoar(arr, first, l);

if (f < last) Hoar(arr, f, last);

}

**Вывод программы**