

Практическая работа №5

Тема «Алгоритмы сортировки»

Цели работы:

1. Изучить алгоритмы сортировки.

Оснащение:

Персональный компьютер под управлением ОС Windows 10.

Офисный пакет Microsoft Office 2016

Среда разработки Visual Studio 2019

1. Пузырьковая сортировка

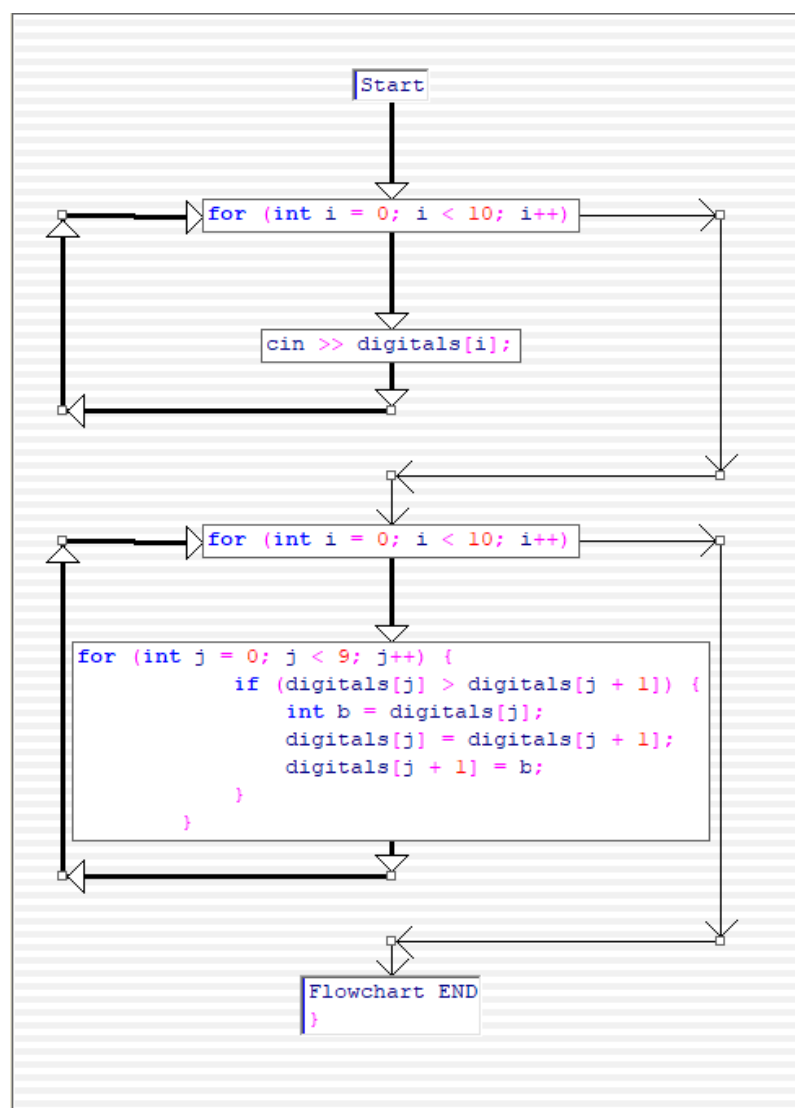


Рисунок 1 – Диаграмма деятельности для пузырьковой сортировки

Код программы

					<i>AuCD.09.03.02.050000 ПР</i>								
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дат</i>	Практическая работа №5 «Алгоритмы сортировки»				<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>		
<i>Разраб.</i>		<i>Ермошина В.А</i>											
<i>Провер.</i>		<i>Берёза А. Н.</i>										2	
<i>Реценз</i>									<i>ИСОиП (филиал) ДГТУ в г.Шахты ИСТ-Тб11</i>				
<i>Н. Контр.</i>													
<i>Утверд.</i>													

```

#include <iostream>
using namespace std;
int main() {
    setlocale(LC_ALL, "rus");
    int digital[10]; // объявили массив на 10 ячеек
    cout << "Введите 10 чисел для заполнения массива: " << endl;
    for (int i = 0; i < 10; i++) {
        cin >> digital[i]; // "читаем" элементы в массив
    }
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 9; j++) {
            if (digital[j] > digital[j + 1]) {
                int b = digital[j]; // создали дополнительную переменную
                digital[j] = digital[j + 1]; // меняем местами
                digital[j + 1] = b; // значения элементов
            }
        }
    }
    cout << "Массив в отсортированном виде: ";
    for (int i = 0; i < 10; i++) {
        cout << digital[i] << " "; // выводим элементы массива
    }
    system("pause");
    return 0;
}

```

Результат

C:\Users\Asus\source\repos\sort1\Debug\sort1.exe
Введите 10 чисел для заполнения массива:
1 23 44 9 0 1 32 5 88 66
Массив в отсортированном виде: 0 1 1 5 9 23 32 44 66 88 Для продолжения нажмите любую клавишу . . .

2. Слияние массивов

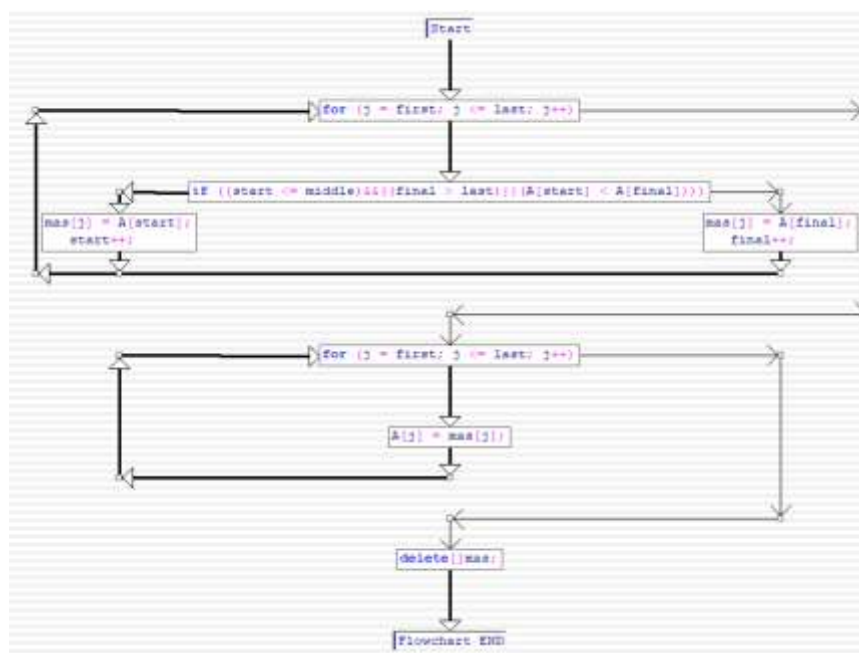


Рисунок 2 – Диаграмма деятельности для сортировки слиянием

Изм.	Лист	№ докум.	Подпись	Дата

АиСД.09.03.02.050000 ПР

Лист

3

Код программы

```
#include <iostream>
using namespace std;
//функция, сливающая массивы
void Merge(int* A, int first, int last)
{
    int middle, start, final, j;
    int* mas = new int[100];
    middle = (first + last) / 2; //вычисление среднего элемента
    start = first; //начало левой части
    final = middle + 1; //начало правой части
    for (j = first; j <= last; j++) //выполнять от начала до конца
        if ((start <= middle) && ((final > last) || (A[start] < A[final])))
        {
            mas[j] = A[start];
            start++;
        }
        else
        {
            mas[j] = A[final];
            final++;
        }
    //возвращение результата в список
    for (j = first; j <= last; j++) A[j] = mas[j];
    delete[] mas;
};
//рекурсивная процедура сортировки
void MergeSort(int* A, int first, int last)
{
    {
        if (first < last)
        {
            MergeSort(A, first, (first + last) / 2); //сортировка левой части
            MergeSort(A, (first + last) / 2 + 1, last); //сортировка правой части
            Merge(A, first, last); //слияние двух частей
        }
    }
};
//главная функция
void main()
{
    setlocale(LC_ALL, "Rus");
    int i, n;
    int* A = new int[100];
    cout << "Размер массива > "; cin >> n;
    for (i = 1; i <= n; i++)
    {
        cout << i << " элемент > "; cin >> A[i];
    }
    MergeSort(A, 1, n); //вызов сортирующей процедуры
    cout << "Упорядоченный массив: "; //вывод упорядоченного массива
    for (i = 1; i <= n; i++) cout << A[i] << " ";
}
```

```

delete[]A;
system("pause>>void");
}

```

Результат

```

C:\Users\Asus\source\repos\sort1\Debug\sort1.exe
Размер массива > 10
1 элемент > 12
2 элемент > 1
3 элемент > 2
4 элемент > 0
5 элемент > 9
6 элемент > 87
7 элемент > 65
8 элемент > 43
9 элемент > 3
10 элемент > 2
Упорядоченный массив: 0 1 2 2 3 9 12 43 65 87

```

3. Шейкерная сортировка

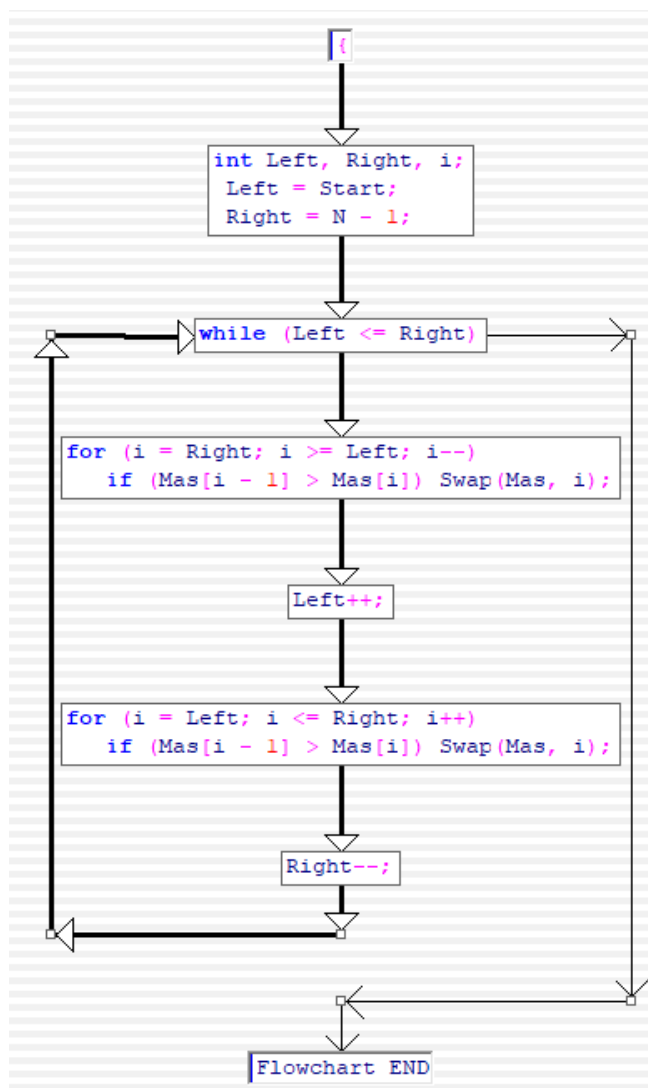


Рисунок 3 – Диаграмма деятельности для шейкерной сортировки

Код программы

```

#include <iostream>
using namespace std;
//функция обмена
void Swap(int* Mas, int i)
{
    int temp;
    temp = Mas[i];
    Mas[i] = Mas[i - 1];
    Mas[i - 1] = temp;
}
//функция шейкерной сортировки
void ShakerSort(int* Mas, int Start, int N)
{
    int Left, Right, i;
    Left = Start;
    Right = N - 1;
    while (Left <= Right)
    {
        for (i = Right; i >= Left; i--)
            if (Mas[i - 1] > Mas[i]) Swap(Mas, i);
        Left++;
        for (i = Left; i <= Right; i++)
            if (Mas[i - 1] > Mas[i]) Swap(Mas, i);
        Right--;
    }
}
//главная функция
void main()
{
    setlocale(LC_ALL, "Rus");
    int n, k;
    cout << "Размер массива > "; cin >> n;
    int* A = new int[n];
    for (k = 0; k < n; k++)
    {
        cout << k + 1 << " элемент > "; cin >> A[k];
    }
    ShakerSort(A, 1, n);
    cout << "Результирующий массив: ";
    for (k = 0; k < n; k++) cout << " " << A[k];
    system("pause>>void");
}

```

Результат

```

C:\Users\Asus\source\repos\sort1\Debug\sort1.exe
Размер массива > 10
1 элемент > 0
2 элемент > 0
3 элемент > 1
4 элемент > 1
5 элемент > 2
6 элемент > 3
7 элемент > 4
8 элемент > 8
9 элемент > 10
10 элемент > 98
Результирующий массив: 0 0 1 1 2 3 4 8 10 98

```

4. Сортировка Шелла

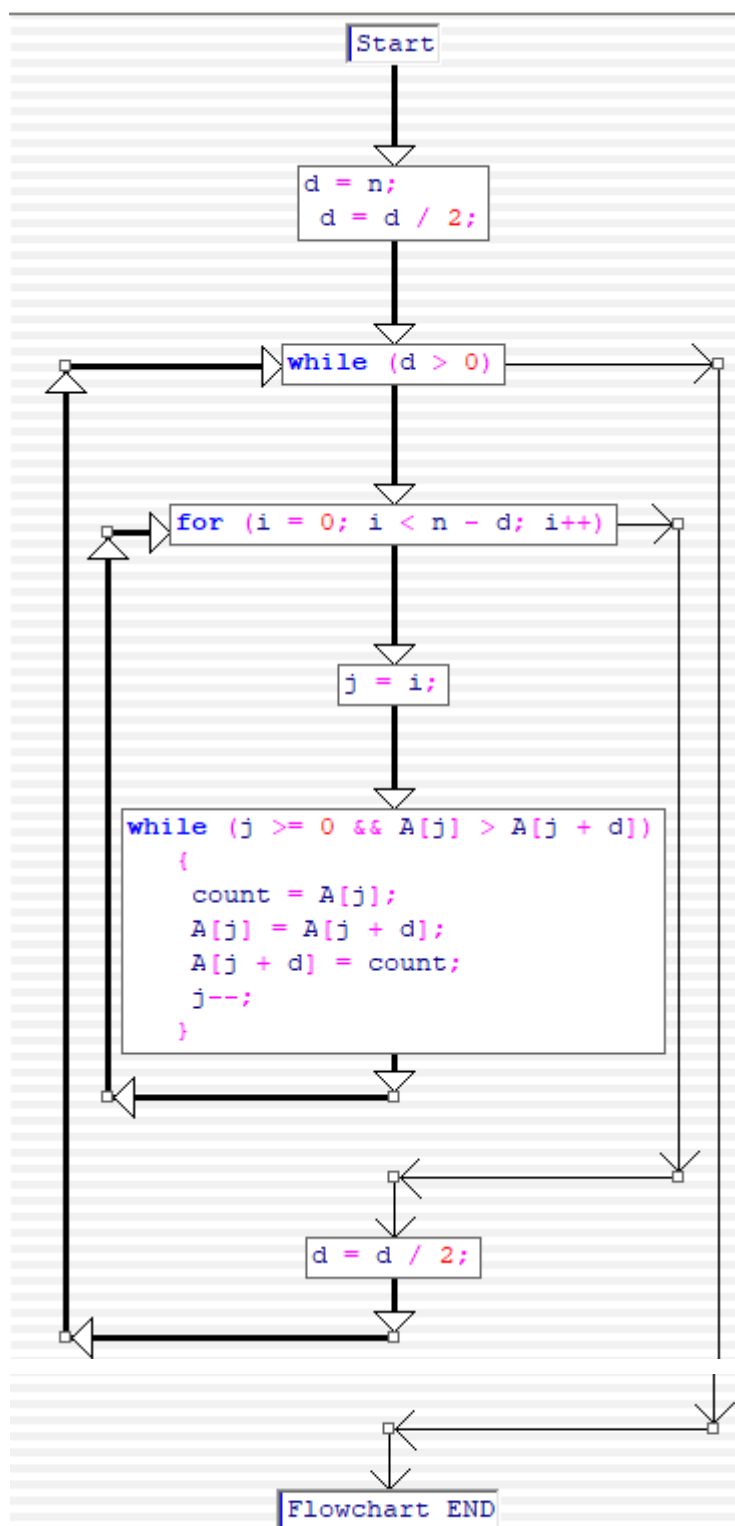


Рисунок 4 – Диаграмма деятельности для сортировки Шелла

Код программы

```

#include <iostream>
using namespace std;
int i, j, n, d, count;
void Shell(int A[], int n) //сортировка Шелла
{
    d = n;

```

Изм.	Лист	№ докум.	Подпись	Дата

АуСД.09.03.02.050000 ПР

Лист

7

```

d = d / 2;
while (d > 0)
{
    for (i = 0; i < n - d; i++)
    {
        j = i;
        while (j >= 0 && A[j] > A[j + d])
        {
            count = A[j];
            A[j] = A[j + d];
            A[j + d] = count;
            j--;
        }
    }
    d = d / 2;
}
for (i = 0; i < n; i++) cout << A[i] << " "; //ВЫВОД массива
}
//главная функция
void main()
{
    setlocale(LC_ALL, "Rus");
    cout << "Размер массива > "; cin >> n;
    int* A = new int[n]; //объявление динамического массива
    for (i = 0; i < n; i++) //ввод массива
    {
        cout << i + 1 << " элемент > "; cin >> A[i];
    }
    cout << "\nРезультирующий массив: ";
    Shell(A, n);
    delete[] A; //освобождение памяти
    system("pause>>void");
}

```

Результат

```

C:\Users\Asus\source\repos\sort1\Debug\sort1.exe
Размер массива > 10
1 элемент > 1
2 элемент > 0
3 элемент > 11
4 элемент > 22
5 элемент > 10
6 элемент > 3
7 элемент > 4
8 элемент > 0
9 элемент > 9
10 элемент > 8

Результирующий массив: 0 0 1 3 4 8 9 10 11 22

```

5. Быстрая сортировка

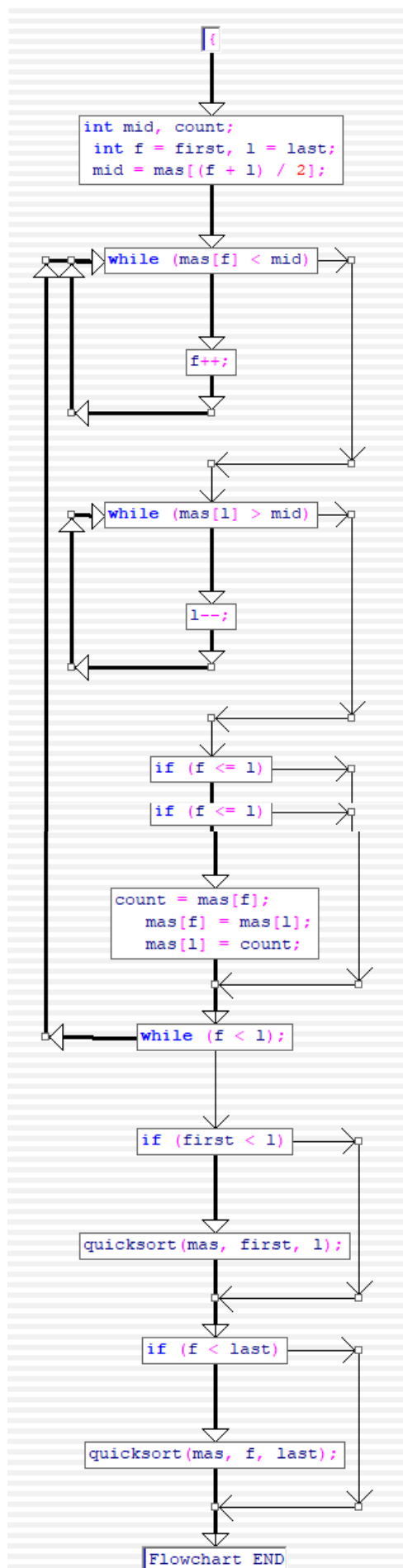


Рисунок 4 – Диаграмма деятельности для быстрой сортировки

Изм.	Лист	№ докум.	Подпись	Дата

АИСД.09.03.02.050000 ПР

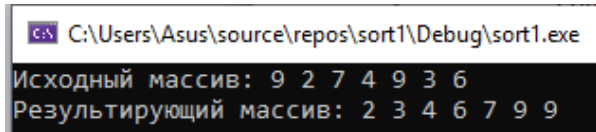
Лист

9

Код программы

```
#include <iostream>
#include <ctime>
using namespace std;
const int n = 7;
int first, last;
//функция сортировки
void quicksort(int* mas, int first, int last)
{
    int mid, count;
    int f = first, l = last;
    mid = mas[(f + l) / 2]; //вычисление опорного элемента
    do
    {
        while (mas[f] < mid) f++;
        while (mas[l] > mid) l--;
        if (f <= l) //перестановка элементов
        {
            count = mas[f];
            mas[f] = mas[l];
            mas[l] = count;
            f++;
            l--;
        }
    } while (f < l);
    if (first < l) quicksort(mas, first, l);
    if (f < last) quicksort(mas, f, last);
}
//главная функция
void main()
{
    setlocale(LC_ALL, "Rus");
    int* A = new int[n];
    srand(time(NULL));
    cout << "Исходный массив: ";
    for (int i = 0; i < n; i++)
    {
        A[i] = rand() % 10;
        cout << A[i] << " ";
    }
    first = 0; last = n - 1;
    quicksort(A, first, last);
    cout << endl << "Результирующий массив: ";
    for (int i = 0; i < n; i++) cout << A[i] << " ";
    delete[] A;
    system("pause>>void");
}
```

Результат



```
C:\Users\Asus\source\repos\sort1\Debug\sort1.exe
Исходный массив: 9 2 7 4 9 3 6
Результирующий массив: 2 3 4 6 7 9 9
```

Изм.	Лист	№ докум.	Подпись	Дата

АуСД.09.03.02.050000 ИР

Лист

10

Вывод: В ходе выполнения лабораторной работы изучила алгоритмы сортировки и написала программы, реализующие пузырьковую сортировку, слияние массивов, шейкерную сортировку, сортировку Шелла и быструю сортировку.

					<i>АиСД.09.03.02.050000 ПР</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11