

Лабораторная работа №1

Оценка сложности алгоритмов сортировки

1 Цель работы

1.1 Научиться реализовывать и оценивать сложность алгоритмов сортировки массивов на C#.

2 Литература

2.1 <https://intuit.ru/studies/courses/648/504/lecture/11435> - описание работы алгоритмов

2.2 <https://www.bigocheatsheet.com/> - сложность алгоритмов.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

3.3 Создать в папке C:\Temp папку с названием группы (ispp__) для хранения создаваемых приложений.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

Для тестирования создать массив, который заполнить теми же значениями, как у исходного массива на рисунках 1-3. Полученный на каждом проходе массив выводить на экран в одну строку.

5.1 Реализовать, проверить и оценить сложность алгоритма сортировки простым выбором (Selection Sort) для одномерного массива.

Это самый простой из алгоритмов сортировки. При данной сортировке из массива выбирается элемент с наименьшим значением и обменивается с первым элементом. Затем из оставшихся $n-1$ элементов снова выбирается элемент с наименьшим значением и обменивается со вторым элементом, и т.д.

Пример показан на рисунке 1, пример схемы алгоритма – на рисунке 4.

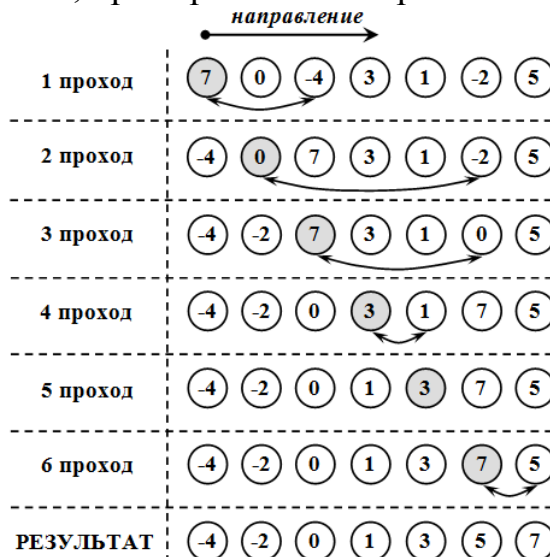


Рисунок 1 — Пример работы алгоритма сортировки простым выбором

5.2 Реализовать, проверить и оценить сложность алгоритма сортировки методом «пузырька» (Bubble Sort) для одномерного массива.

Алгоритм попарного сравнения элементов массива в литературе часто называют «методом пузырька», проводя аналогию с пузырьком, поднимающимся со дна бокала с газированной водой. По мере всплывания пузырек сталкивается с другими пузырьками и, сливаясь с ними, увеличивается в объеме. Чтобы аналогия стала очевидной, нужно считать, что элементы массива расположены вертикально друг над другом, и их нужно так упорядочить, чтобы они увеличивались снизу вверх.

Алгоритм состоит в повторяющихся проходах по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется обмен элементов. Проходы по массиву повторяются до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает – массив отсортирован. При проходе алгоритма элемент, стоящий не на своём месте, "всплывает" до нужной позиции.

Пример работы показан на рисунке 2, пример схемы алгоритма – на рисунке 5.



Рисунок 2 — Пример работы алгоритма сортировки методом пузырька

5.3 Реализовать, проверить и оценить сложность алгоритма сортировки вставками (Insertion Sort) для одномерного массива.

На каждом шаге алгоритма выбираем один из элементов входных данных и вставляем его на нужную позицию в уже отсортированной последовательности до тех пор, пока набор входных данных не будет исчерпан. Метод выбора очередного элемента из исходного массива произволен; может использоваться практически любой алгоритм выбора.

Пример работы показан на рисунке 3, пример схемы алгоритма – на рисунке 6.

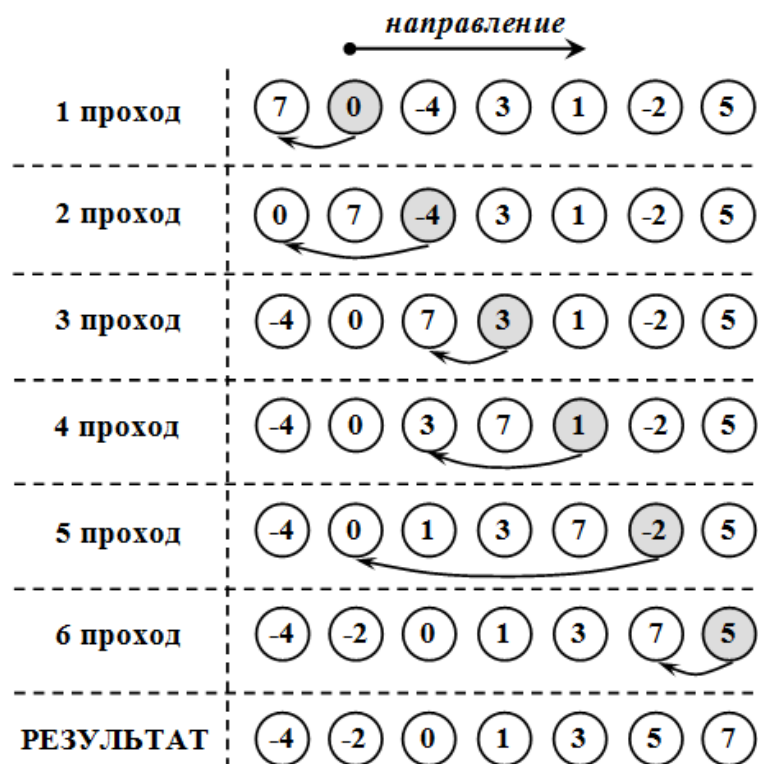


Рисунок 3 — Пример работы алгоритма сортировки вставками

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C# (Console Application (Microsoft)) с названием решения LabWork1.

6.2 Выполнить все задания из п.5 в решении LabWork1. Каждое задание должно быть в своем проекте с названием Task1, Task2 и т.д.

При выполнении заданий использовать минимально возможное количество команд и переменных и выполнять форматирование и рефакторинг кода.

Наименования переменных должны быть осмысленными (например, i – счетчик первого уровня, j – второго, $temp$ – переменная для хранения временного значения (например, для обмена данными)).

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «массив»?

8.2 Как описывается одномерный массив?

8.3 Как обратиться к некоторому элементу одномерного массива?

8.4 Как можно задать одномерный массив?

8.5 Что такое «сортировка»?

8.6 Что такое «алгоритм сортировки»?

8.7 Какие виды сортировки массивов существуют?

9 Приложение

9.1 Массивы в C#

Массив — это структура однотипных элементов, занимающих непрерывную область памяти.

С массивом связаны следующие его свойства: имя, тип, размерность, размер.

Формат описания одномерного массива из указанного количества элементов:

тип[] имя = new тип[количество элементов];

Например:

```
int[] arr = new int[5]; // массив arr из пяти элементов целого типа
```

Элементы массива обозначаются индексированными именами. Нижнее значение индекса равно 0, верхнее — количество элементов-1:

arr[0], arr[1], arr[2], arr[3], arr[4]

Размер массива всегда на единицу больше максимального значения индекса.

Значения массива можно указать при инициализации. Например:

```
int[] arr = {2, 4, 6, 10, 1};
```

Здесь массив из пяти элементов: arr[0]=2, arr[1]=4, arr[2]=6, arr[3]=10, arr[4]=1

Заполнение массива случайными числами:

```
int[] numbers = new int[5];
Random random = new();
for (int i = 0; i < numbers.Length; i++)
{
    numbers[i] = random.Next(100); // случайное число от 0 до 100
}
```

Заполнение массива с клавиатуры:

```
int[] numbers = new int[5];
for (int i = 0; i < numbers.Length; i++)
{
    numbers[i] = Convert.ToInt32(Console.ReadLine());
}
```

Вывод массива на экран в столбец:

```
int[] numbers = { 1, 2, 3, 4, 5 };
for (int i = 0; i < numbers.Length; i++)
{
    Console.WriteLine(numbers[i]);
}
```

Вывод массива на экран в строку (разделитель – табуляция):

```
int[] numbers = { 1, 2, 3, 4, 5 };
for (int i = 0; i < numbers.Length; i++)
{
    Console.Write($"{numbers[i]}\t");
}
```

9.2 Сортировка

Сортировка – это процесс перестановки элементов данного множества в определенном порядке.

Сортировка является идеальным примером огромного разнообразия алгоритмов, выполняющих одну и ту же задачу, многие из которых в некотором смысле являются оптимальными, а большинство имеет какие-либо преимущества по сравнению с остальными. Поэтому на примере сортировки убеждаются в необходимости проведения сравнительного анализа алгоритмов.

Алгоритм сортировки — это алгоритм для упорядочивания элементов в списке.

В случае, когда элемент списка имеет несколько полей, поле, служащее критерием порядка, называется ключом сортировки. На практике в качестве ключа часто выступает число, а в остальных полях хранятся какие-либо данные, никак не влияющие на работу алгоритма.

Устойчивая сортировка не меняет взаимного расположения элементов с одинаковыми ключами.

Примеры алгоритмов сортировки:

- Сортировка методом простого выбора – это алгоритм последовательного обмена минимального и первого элементов неотсортированной части массива (схема алгоритма показана на рисунке 4).
- Сортировка методом «пузырька» – это алгоритм попарного сравнения элементов одномерного массива (схема алгоритма показана на рисунке 5).
- Сортировка вставками (или методом простого включения) – это алгоритм последовательного помещения элемента массива в отсортированную часть в соответствии с ключом сортировки (схема алгоритма показана на рисунке 6).

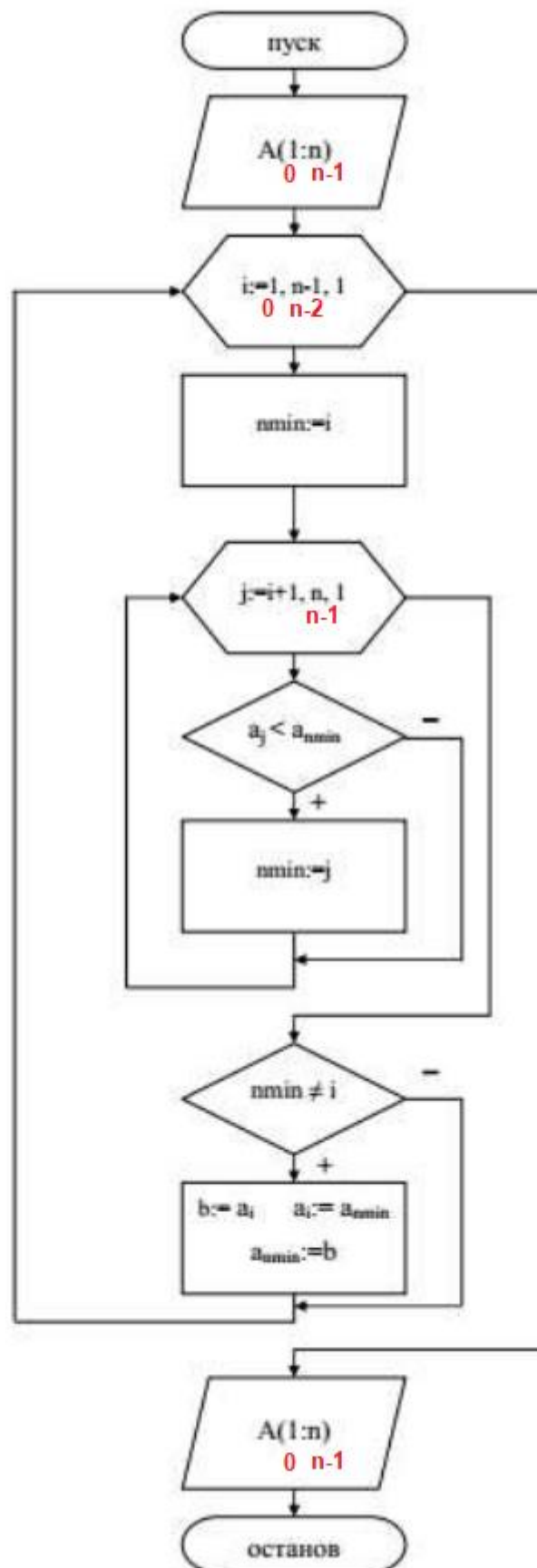


Рисунок 4 — Схема алгоритма сортировки методом простого выбора

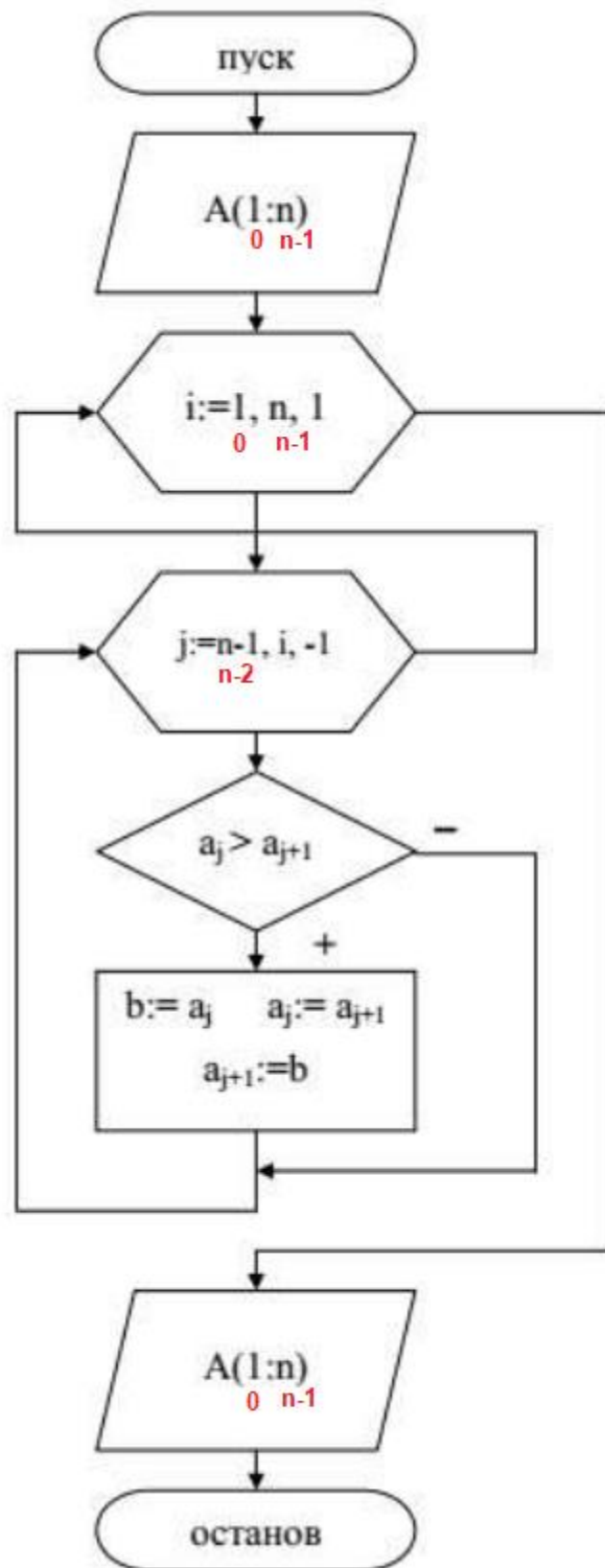


Рисунок 5 — Схема алгоритма сортировки методом «пузырька»

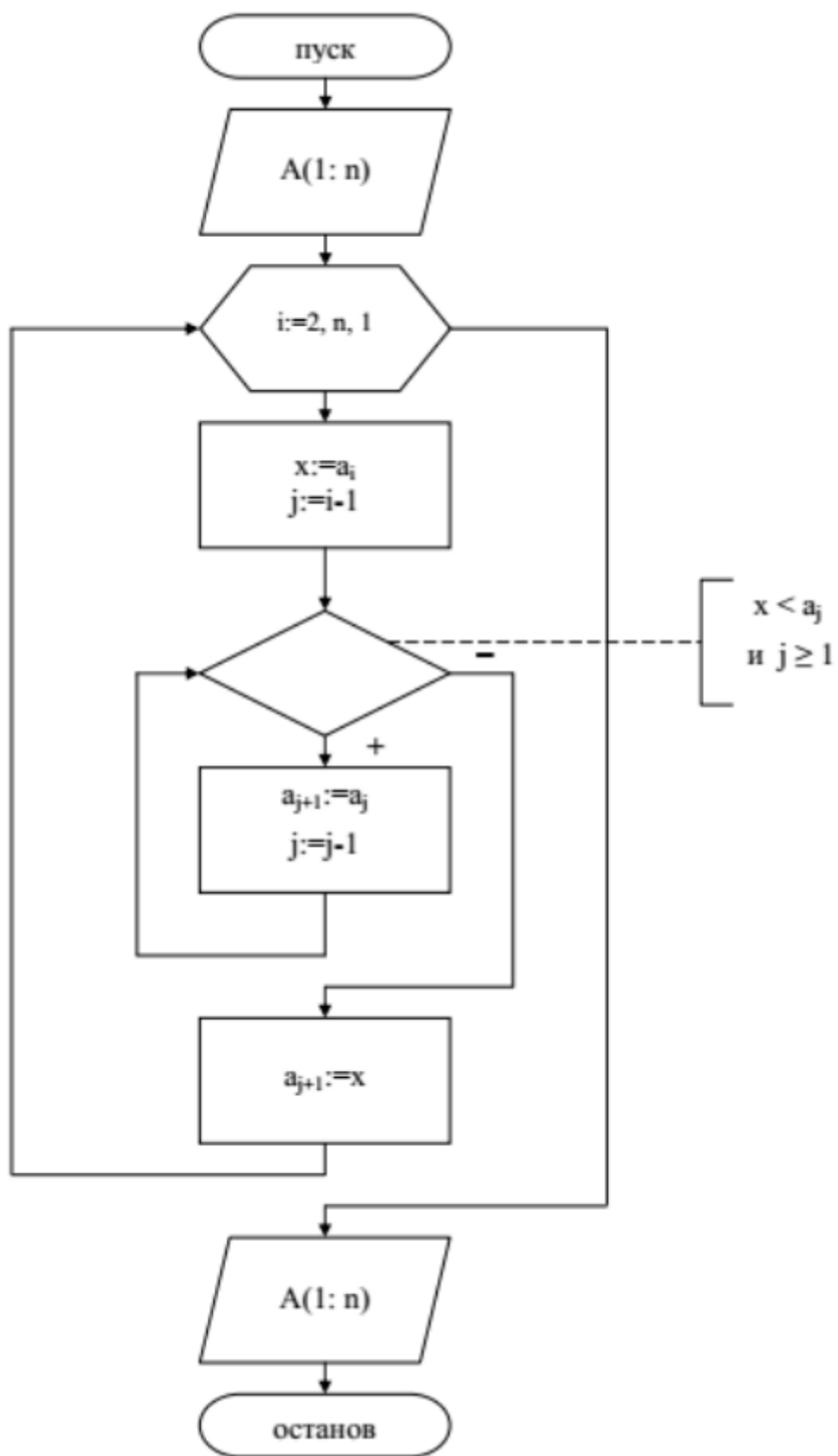


Рисунок 6 — Схема алгоритма сортировки вставками