Лабораторная работа №2
Оценка сложности алгоритмов сортировки

1Цель работы

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

2Литература

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

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

3Подготовка 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 (isppNN) для хранения создаваемых приложений.

4Основное оборудование

4.1 Персональный компьютер.

5Задание

Для обмена значений элементов проще использовать кортежи: (a, b) = (b, a)

На каждом проходе массива реализовать вывод полученного массива в строку, используя:

Debug.WriteLine(string.Join("\t", array));

или точку останова с действием:

{string.Join("\t", array)}

Для тестирования и отладки каждого алгоритма сортировки создать набор массивов для теста:

- массив с теми же значениями, как у исходного массива на рисунках 1-3 и проверить, что он сортируется корректно, используя Debug.Assert,
- массив, заполненный значениями по возрастанию,
- массив, заполненный значениями по убыванию,
- массив, у которого первый элемент нарушает порядок (должен быть последним),
- массив, у которого последний элемент нарушает порядок (должен быть первым).

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 неотсортированных элементов снова выбирается элемент с наименьшим значением и обменивается со вторым элементом, и т.д.

Пример работы 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 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.

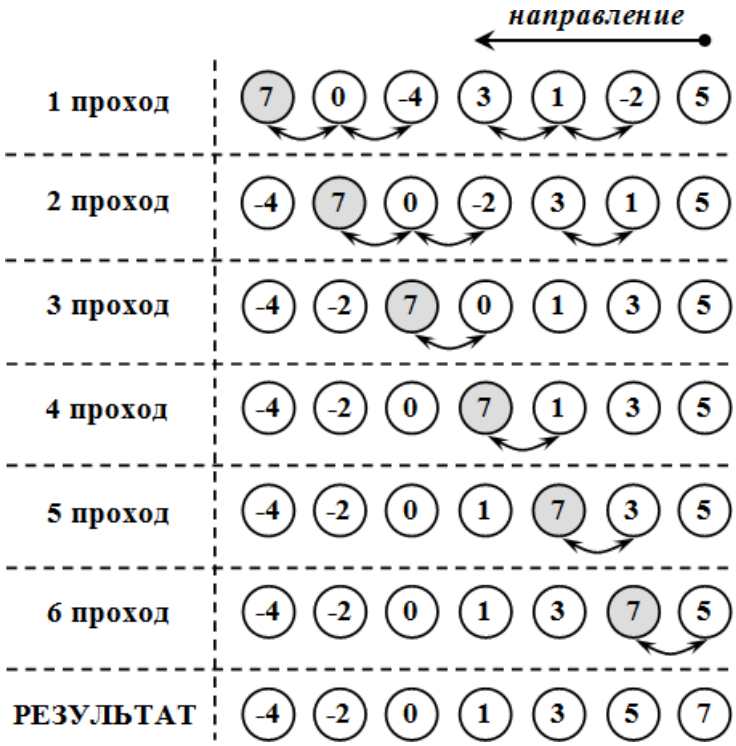

Рисунок 2 – Пример работы алгоритма сортировки методом пузырька

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

На каждом шаге алгоритма выбирается один из элементов входных данных и вставляется 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

6Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C# (Console Application (Microsoft)) с названием решения LabWork2.

6.2 Выполнить все задания из п.5 в решении LabWork2. Каждое задание должно быть в своем методе с сигнатурой ТипСортировкиSort(int[] array).

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

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

6.3 Ответить на контрольные вопросы.

7Содержание отчета

7.1 Титульный лист
7.2 Цель работы
7.3 Ответы на контрольные вопросы
7.4 Вывод

8Контрольные вопросы

8.1 Что такое «массив»?
8.2 Что такое «сортировка»?
8.3 Что такое «алгоритм сортировки»?
8.4 Какие виды сортировки массивов существуют?

9Приложение

9.1 Массивы в C#

Массив – структура однотипных элементов, занимающих непрерывную область памяти.

Формат описания одномерного массива из указанного количества элементов:

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

Например:

```
int[] array = new int[5];
```

Значения массива можно указать при инициализации. Например:

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

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

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:

```
for (int i = 0; i < numbers.Length; i++)
  Console.WriteLine(numbers[i]);
```

или

```
string.Join("\n", numbers);
```

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

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

или

```
string.Join("\t", numbers);
```

9.2 Сортировка

Сортировка – процесс перестановки элементов данного множества в определенном порядке.

Сортировка является 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 – 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.

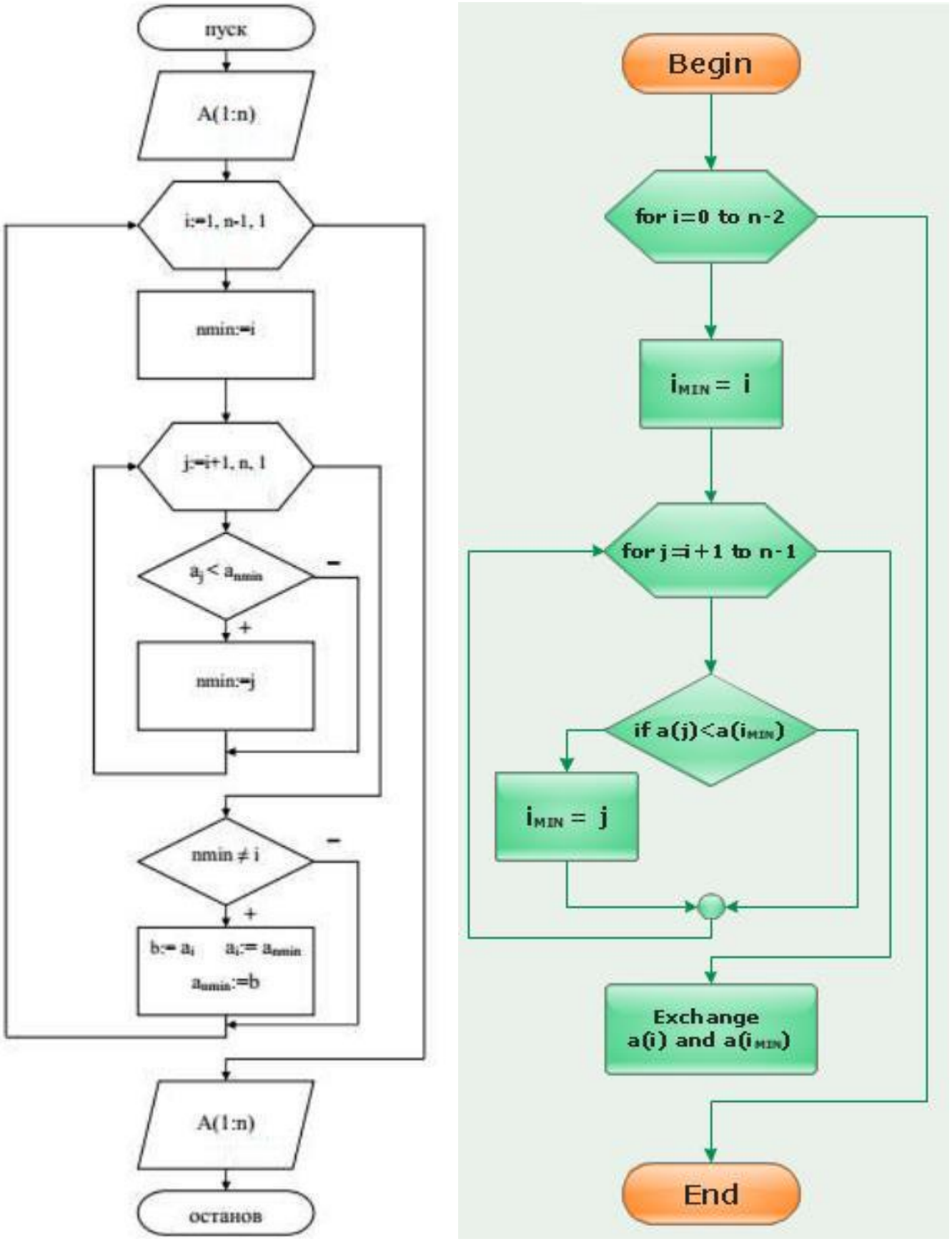

Рисунок 4 – Схема алгоритма сортировки методом простого выбора

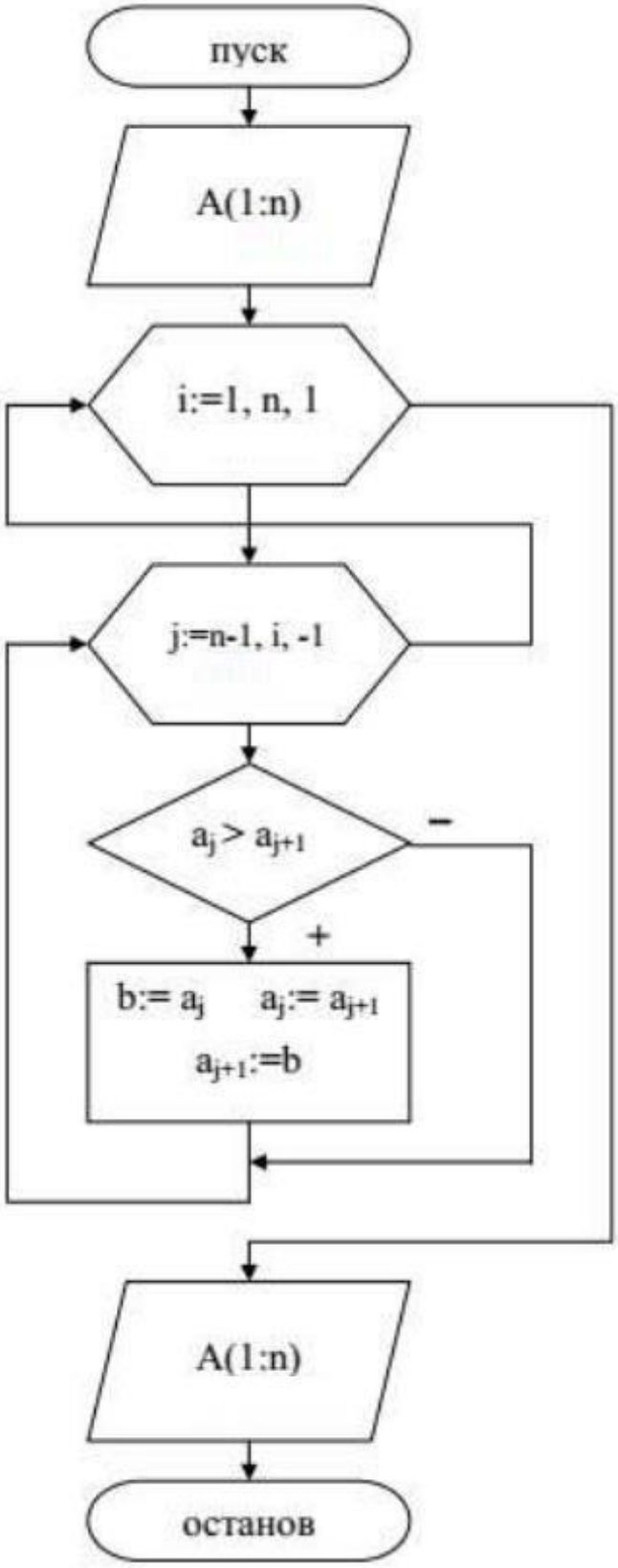

Рисунок 5 – Схема алгоритма сортировки методом «пузырька»

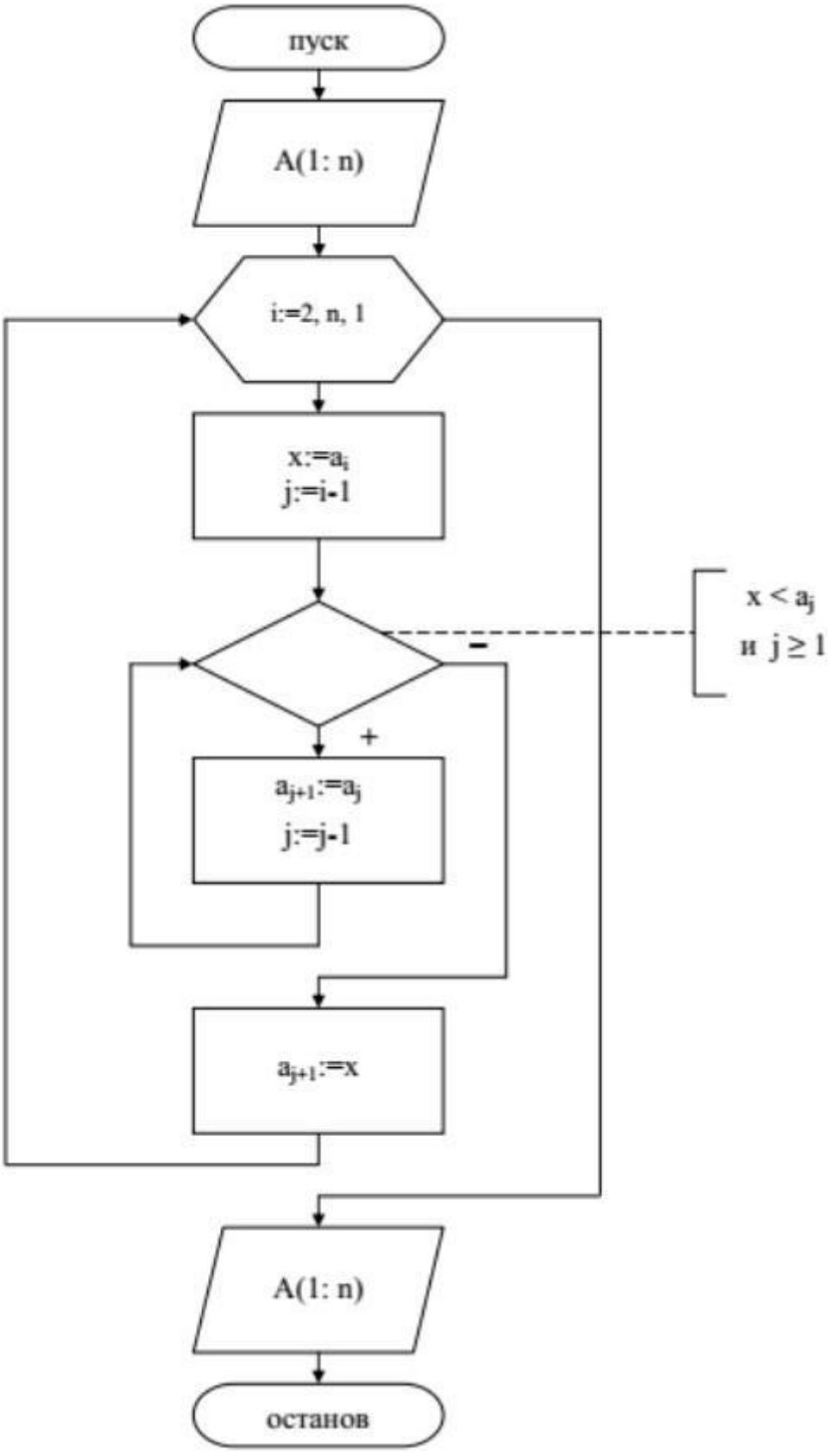

Рисунок 6 – Схема алгоритма сортировки вставками