Лабораторная работа № 30-31

«Оценка сложности и оформление алгоритмов сортировки массива»

Цель работы:
изучить оценку сложности и оформление алгоритмов сортировки массива.

Теория

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.

Оценка алгоритма сортировки

Алгоритмы сортировки оцениваются по скорости выполнения и эффективности использования памяти:

- Время — основной параметр, характеризующий быстродействие алгоритма. Называется также вычислительной сложностью. Для упорядочения важны худшее, среднее и лучшее поведение алгоритма в терминах мощности входного множества A. Если на вход алгоритму подаётся множество A, то обозначим n = |A|. Для типичного алгоритма хорошее поведение — это O(n log n) и плохое поведение — это O(n2). Идеальное поведение для упорядочения — O(n). Алгоритмы сортировки, использующие только абстрактную операцию сравнения ключей всегда нуждаются по меньшей мере в сравнениях. Тем не менее, существует алгоритм сортировки Хана (Yijie Han) с вычислительной сложностью O(n log log n log log log n), использующий тот факт, что пространство ключей ограничено (он чрезвычайно сложен, а за O-обозначением скрывается весьма большой коэффициент, что делает невозможным его применение в повседневной практике). Также существует понятие сортирующих сетей. Предполагая, что можно одновременно (например, при параллельном вычислении) проводить несколько сравнений, можно отсортировать n чисел за O(log2 n) операций. При этом число n должно быть заранее известно;

- Память — ряд алгоритмов требует выделения дополнительной памяти под временное хранение данных. Как правило, эти алгоритмы требуют O(log n) памяти. При оценке не учитывается место, которое занимает исходный массив и независящие от входной последовательности затраты, например, на хранение кода программы (так как всё это

потребляет O(1)). Алгоритмы сортировки, не потребляющие дополнительной памяти, относят к сортировкам на месте.

Пример: простейшая программа сортировки массива.

Создадим новое консольное приложение. И изменим код файла Program.cs на следующий:

```
using System;
namespace SortApp
{
    class Program
    {
    static void Main(string[] args)
        {
            // ввод чисел
            int[] nums = new int[7];
            Console.WriteLine("Введите семь чисел");
            for (int i = 0; i < nums.Length; i++)
            {
                Console.Write("{0}-е число: ", i + 1);
                nums[i] = Int32.Parse(Console.ReadLine());
            }
            // сортировка
            int temp;
            fo  r (int i = 0; i < nums.Length-1; i++)
            {
                for (int j = i + 1; j < nums.Length; j++)
                {
                    if (nums[i] > nums[j])
                    {
                        temp = nums[i];
                        nums[i] = nums[j];
                        nums[j] = temp;
                    }
                }
            }
            // вывод
            Console.WriteLine("Вывод отсортированного массива");
            for (int i = 0; i < nums.Length; i++)
            {
                Console.WriteLine(nums[i]);
            }
            Console.ReadLine();
        }
    }
}
```

Вся программа условно поделена на три блока: ввод чисел, сортировку и вывод отсортированного массива. Здесь используются все те же конструкции, что были

рассмотрены ранее. Сначала в цикле мы вводим все числа для массива. Так как метод Console.ReadLine() возвращает вводимую строку, а нам нужны числа, поэтому мы эту строку переводим в число с помощью метода Int32.Parse(Console.ReadLine()).

Затем сортируем: выполняем проходы по массиву и сравниваем элементы. Если элемент с меньшим индексом больше элемента с большим индексом, то меняем элементы местами.

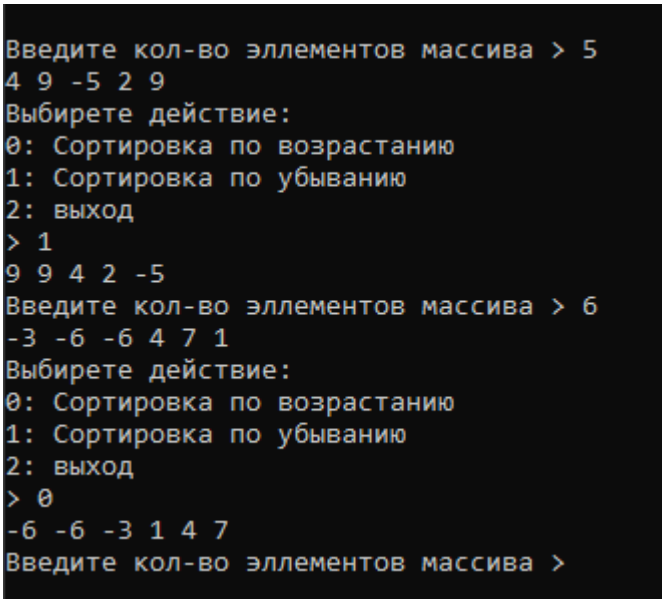
В конце выводим все элементы.

Ход работы:

1. Написать программу, которая сортирует массив по возрастанию.
2. Написать программу, которая сортирует массив по убыванию

```
static void Main(string[] args)
{
    onChoice();

    void onChoice()
    {

        int[] sortingArr = createArr();
        Console.WriteLine("\nВыбирете действие:");
        Console.WriteLine("0: Сортировка по возрастанию");
        Console.WriteLine("1: Сортировка по убыванию");
        Console.WriteLine("2: выход");
        Console.Write("> ");
        try
        {
            int choice = Convert.ToInt32(Console.ReadLine());


            switch (choice)
            {
                case 0:
                    sortArr(sortingArr, true);
                    break;
                case 1:
                    sortArr(sortingArr, false);
                    break;
                case 2:
                    System.Environment.Exit(0);
                    break;
                default:
                    Console.WriteLine(); onChoice();
                    break;
            }
```

```
        }
        catch
        {
            Console.WriteLine("\nВведите корректное значение\n");
        }
    }

    int[] sortArr(int[] arr, bool increase)
    {
        Array.Sort(arr);
        if (!increase) Array.Reverse(arr);
        showArr(arr);
        return arr;
    }

    int[] createArr()
    {
        Random rnd = new Random();

        Console.Write("\nВведите кол-во эллементов массива ");
        Console.Write("> ");
        int[] array = new int[Convert.ToInt32(Console.ReadLine())];

        for (int i = 0; i < array.Length; i++)
        {
            array[i] = rnd.Next(-10, 10);
        }

        showArr(array);

        return array;
    }

    void showArr(int[] arr)
    {
        for (int i = 0; i < arr.Length; i++)
        {
            Console.Write(arr[i] + " ");
        }
    }

    while (true)
    {
        onChoice();
    }
}
```



Контрольные вопросы:

1. Что такое алгоритм сортировки?
   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.
2. Какие оценки алгоритма сортировки вы знаете?

- Время - основной параметр, характеризующий быстродействие алгоритма.
- Память - ряд алгоритмов требует выделения дополнительной памяти под временное хранение данных.

3. Какие алгоритмы сортировки относят к алгоритмам сортировкам на месте?
   Алгоритмы сортировки, не потребляющие дополнительной памяти, относят к сортировкам на месте.