Университет информационных технологий, механики и оптики
Факультет компьютерных технологий и управления
Кафедра информатики и прикладной математики

ЛАБОРАТОРНАЯ РАБОТА №2
«СОРТИРОВКИ ЗА O(N*LOG(N))»

Выполнил:
студент гр. P3118
Петкевич К. В.

Принял:
к.т.н старший
преподаватель
Симоненко З. Г.

г. Санкт-Петербург
2016 год

Цель работы

Для выполнения лабораторной работы необходимо сгенерировать тестовые файлы (используя генераторы случайных чисел), содержащие целые числа, в количестве от 2^6 до 2^{20} (можно и больше), при этом количество элементов в следующем файле в два раза больше чем в предыдущем, реализовать алгоритмы используя один из следующих языков программирования: C++, C#, C, Python, для каждого тестового файла из набора выполнить сортировку данных, вычислить среднее время сортировки по одному файлу, построить график зависимости времени сортировки от количества элементов в файле, выполнить сравнение алгоритмов.

Текст генератора исходных данных

```
static public TimeSpan FileCreator(int n, string path)
{
        Random rnd = new Random((int)DateTime.Now.Ticks);
        Stopwatch timer = new Stopwatch();
        TimeSpan time;
        string s = path + "/TestFile";
        int i = 0, j = 0;
        timer = Stopwatch.StartNew();
        for (i = 0; i < n; i++)
        {
                string str = @s + i + ".txt";
                StreamWriter stream = File.AppendText(str);
                for (j = 0; j < (Math.Pow(2, 6 + i)); j++)
                {
                        string line = Convert.ToString(rnd.Next(0, Convert.ToInt32(Math.Pow(2, 6 + i))));
                        stream.WriteLine(line);
                }
                stream.Close();
        }
        Console.WriteLine("\nGenerated!\n");
        timer.Stop();
        time = timer.Elapsed;
        return (time);
}
```

Коды сортировок

1. **Бинарные вставки**

```
for (int i = 1; i < a.Count; i++)
{
        if (a[i - 1] > a[i])
        {
                int temp = a[i];
                int left = 0;
                int right = i - 1;
                do
                {
                        int middle = (right + left) / 2;
                        if (a[middle] > temp)
                        {
                                right = middle - 1;
                        }
                        else
                        {
```

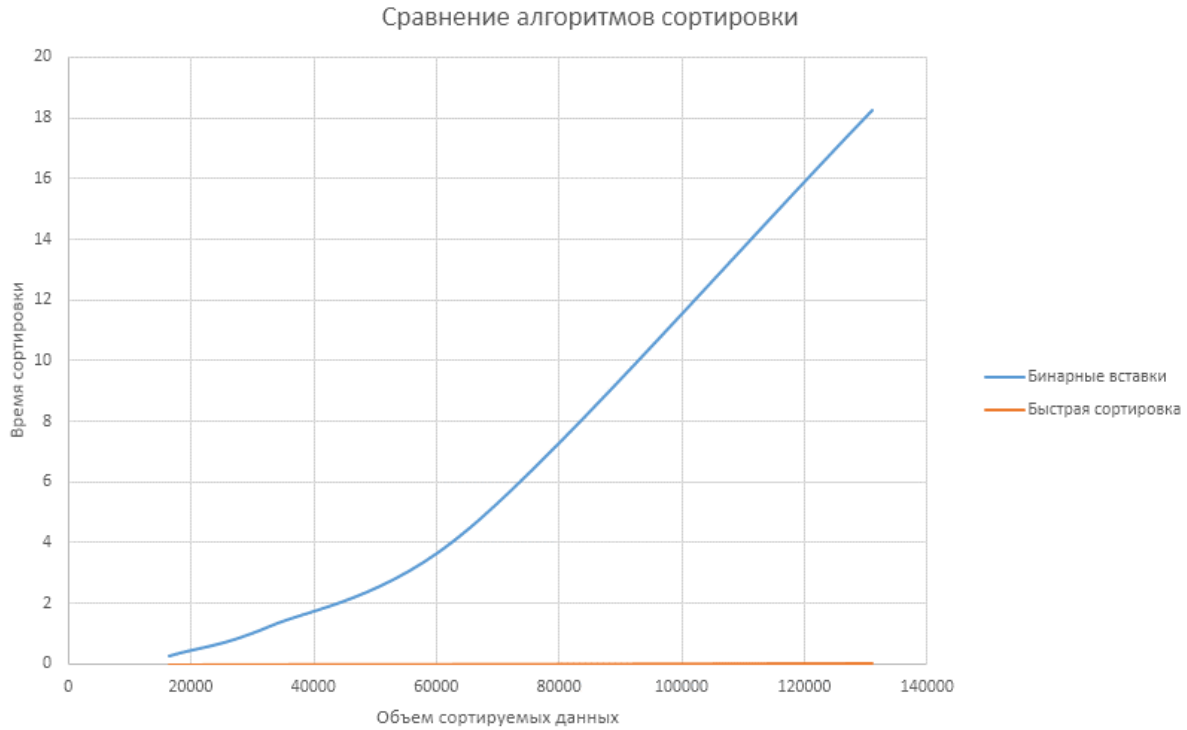
```
                    left = middle + 1;
                }
            } while (right >= left);
            for(int j = i -1; j >= left; j--)
            {
                a[j + 1] = a[j];
            }
            a[left] = temp;
        }
}
```

2. Быстрая сортировка

```
int lef = left;
int rig = right;

if (rig > lef)
{
int pivot = array[(right + left) / 2];

        while (lef <= rig)
        {
            while (lef < right && array[lef] < pivot) lef++;
            while (rig > left && array[rig] > pivot) rig--;

            if (lef <= rig)
            {
                swap(ref array[lef], ref array[rig]);
                lef++;
                rig--;
            }
        }
        if (left < rig)
            QuickSort(array, left, rig);
        if (lef < right)
            QuickSort(array, lef, right);
}
```

Результаты

	Время сортировки, с	
Кол-во эл-в	Бинарными вставками	Быстрая сортировка
16384	0,216	0,003
32768	1,183	0,008
65536	4,187	0,012
131072	19,041	0,035



Вывод

Среди алгоритмов сортировки за O(N*LOG(N)), эффективнее по времени проявил себя алгоритм «Быстрая сортировка». Сортировка бинарными вставками сильно зависит от количества обрабатываемых элементов, что нельзя сказать о «Быстрой сортировке». Но оба алгоритма применимы на практике.