

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

ЛАБОРАТОРНАЯ РАБОТА №1
«СОРТИРОВКИ ЗА $O(N^2)$ »

Выполнил:
студент гр. Р3118
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. Пузырьковая сортировка

```
int length = a.Count;
for (int i = 0; i < length; i++)
{
    for (int j = 0; j < length - i; j++)
    {
        if (a[j] > a[j + 1])
        {
            int b = a[j];
            a[j] = a[j + 1];
            a[j + 1] = b;
        }
    }
}
```

2. Методом выбора

```
for (int i = 0; i < a.Count - 1; i++)
{
    int min = i;
    for (int j = i + 1; j < a.Count; j++)
    {
        if (a[j] < a[min])
        {
            min = j;
        }
    }
    int b = a[i];
    a[i] = a[min];
    a[min] = b;
}
```

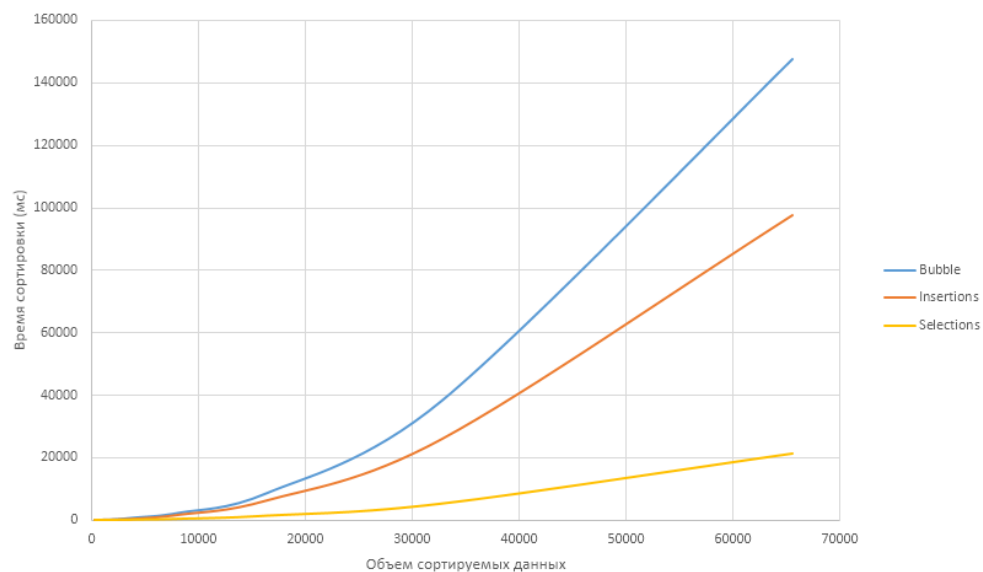
3. Простыми вставками

```
for (int i = 1; i < a.Count; i++)
{
    int j = i - 1;
    while( j >= 0 && a[j] > a[j + 1])
    {
        int b = a[j];
        a[j] = a[j + 1];
        a[j + 1] = b;
        j--;
    }
}
```

Результаты

Кол-во эл-в	Время сортировки, с		
	Пузырьковый	Простые вст.	Выбором
16384	4,023	2,197	1,729
32768	14,458	8,498	7,401
65536	54,987	34,953	29,105
131072	234,319	144,309	112,321

Сравнение алгоритмов сортировки



Вывод

В сравнении с другими алгоритмами сортировки за $O(n^2)$, сортировка выбором оказалась самой эффективной по времени, а худшей – сортировка «Пузырьком». Не смотря на скорость сортировки выбором, этот и другие алгоритмы не эффективно применять на практике.