

# Анализ работы сортировок в C++

Автор: Васильев К.С. РЛ6-21 МГТУ им. Н. Э. Баумана

## Основной цикл For

```
for (int i = 0; i < 1000; ++i) {
    if (n>5 || k>10000000){break;}
    for (int j = 6; j >=n; j--)
    {
        if ((j==5 || j==6)&& k>10000)
        {
            k=k*500;
        }
        fout << k<<"\t ";
        int *randcifri = new int[k];
        zapolnicferkami(randcifri, k);
        clock_t t;
        clock_t vr;
        t = clock();
        psort[j](randcifri, k,0);
        t = clock() - t;
        zapolnicferkami(randcifri, k);
        vr =clock();
        psort[j](randcifri, k,0);
        t = t + clock() - vr;
        zapolnicferkami(randcifri, k);
        vr =clock();
        psort[j](randcifri, k,0);
        t = t + clock() - vr;
        zapolnicferkami(randcifri, k);
        vr =clock();
        psort[j](randcifri, k,0);
        t = t + clock() - vr;
        zapolnicferkami(randcifri, k);
        vr =clock();
        psort[j](randcifri, k,0);
        t = t + clock() - vr;
        t=t/5;
        if (t > 10000)
        {
            n++;
        }
        int ch=0;
        for (int l = 0; l < k; ++l)
        {
```

```

        if (randcifri[l]<=randcifri[l+1]){
        }
        else
        {
            ch++;
        }
    }
    cout << j << " " << t << " " << k << " " << ch << endl;
    delete[] randcifri;
    fout << t << "\t ";
    if ((j==5 | j==6)&& k>10000)
    {
        k=k/500;
    }
}
if (k<10000)
k=k+100;
if (k>=10000&&k<1000000)
    k=k+20000;
if (k>=1000000)
    k=k+500000;
fout << "\n ";
}

```

В основной части программы используется цикл (for) для постепенного увеличения размера сортируемого массива . Производится 56 измерений, минимальный сортируемый массив имеет размер 100 , а максимальный ограничивается временем 1 сортировки (10 секунд), то есть индивидуальное для каждой сортировки.

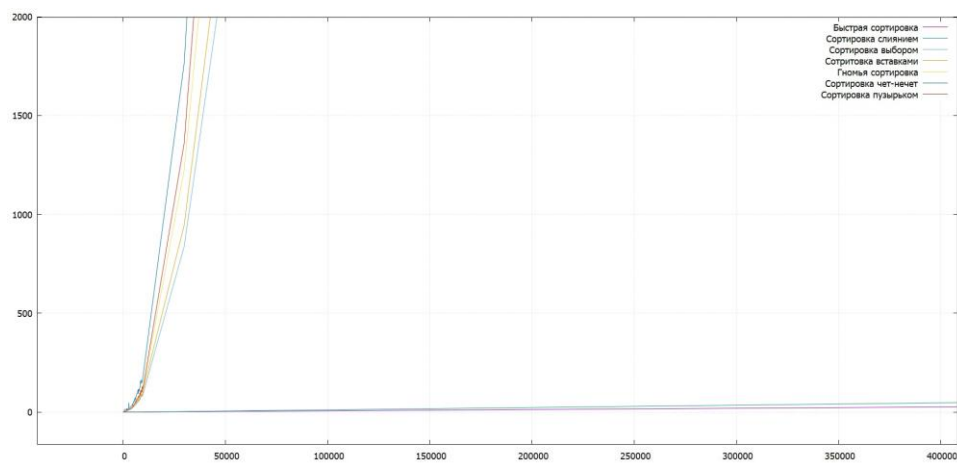
Пояснения к действиям в цикле: 1. Массив заполняется случайными значениями( В следующем цикле, отсортированный массив снова затрётся случайными значениями) 2. В текстовый файл добавляется координата X 3. Замеряется время выполнения программы (Используется библиотека chrono для более точный замеров) 4. Вызывается ссылка на необходимую функцию 5. Время выполнения программы записывается в

переменную int (Особенности библиотеки chrono и типа auto) 6. В текстовый файл добавляется координата Y

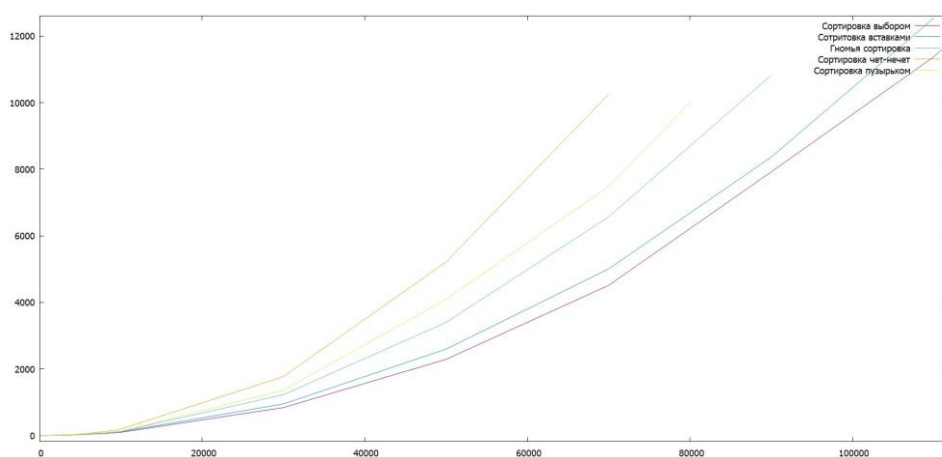
## Результаты работы программы

Для приведённый графиком бралось среднее значение из 3 замеров для более точного отображения эффективности программы.

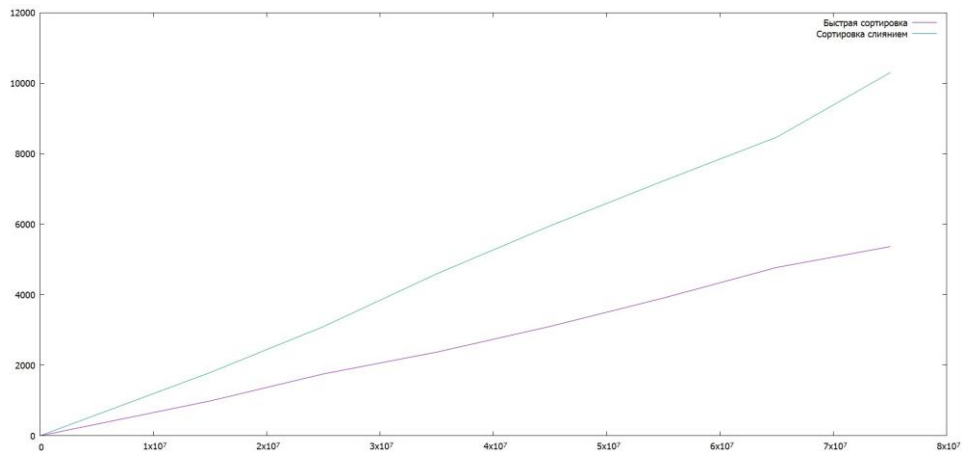
### График всех сортировок



### Более медленные сортировки



### Более быстрые сортировки



## Анализ результатов

На малых значениях массива невозможно увидеть разницу из-за недостаточной точности измерения. Однако если рассматривать Скорость работы сортировок на больших размерах массива то результаты получаются такими: Список от самой медленной сортировки к самой быстрой: 1. Сортировка чет-нечет 2. Сортировка пузырьком 3. Гномья сортировка 4. Сортировка вставками 5. Сортировка выбором 6. Сортировка слиянием 7. Быстрая сортировка