

Анализ сортировок C++

Подготовил: Мазалаев Г. Группа: РЛ6-21.

Цикл for:

```
for (int i = 0; i < 1; i++)
{
    cout << "Sorting into 1000000 elements:" << endl;

    arr = new int[size];
    FillArray(arr, size - 1);

    auto start1 = clock();
    BubbleSort(arr, size - 1);
    auto end1 = clock();
    unsigned int time1 = end1 - start1;

    auto start2 = clock();
    insertionSort(arr, size - 1);
    auto end2 = clock();
    unsigned int time2 = end2 - start2;

    auto start3 = clock();
    SelectionSort(arr, size - 1);
    auto end3 = clock();
    unsigned int time = end3 - start3;

    auto start4 = clock();
    HeapSort(arr, size/2-1, size - 1);
    auto end4 = clock();
    unsigned int time4 = end4 - start4;

    auto start5 = clock();
    CombSort(arr, size - 1);
    auto end5 = clock();
    unsigned int time6 = end5 - start5;

    auto start6 = clock();
    mergeSort(arr, 0, size);
    auto end6 = clock();
    unsigned int time6 = end6 - start6;

    auto start7 = clock();
    QuickSort(arr, 0, size);
    auto end7 = clock();
    unsigned int time7 = end7 - start7;

    delete[] arr;
}
```

В главной функции `main()` используется цикл `for`. Проводится одно измерение массива на 1 000 000 элементов.

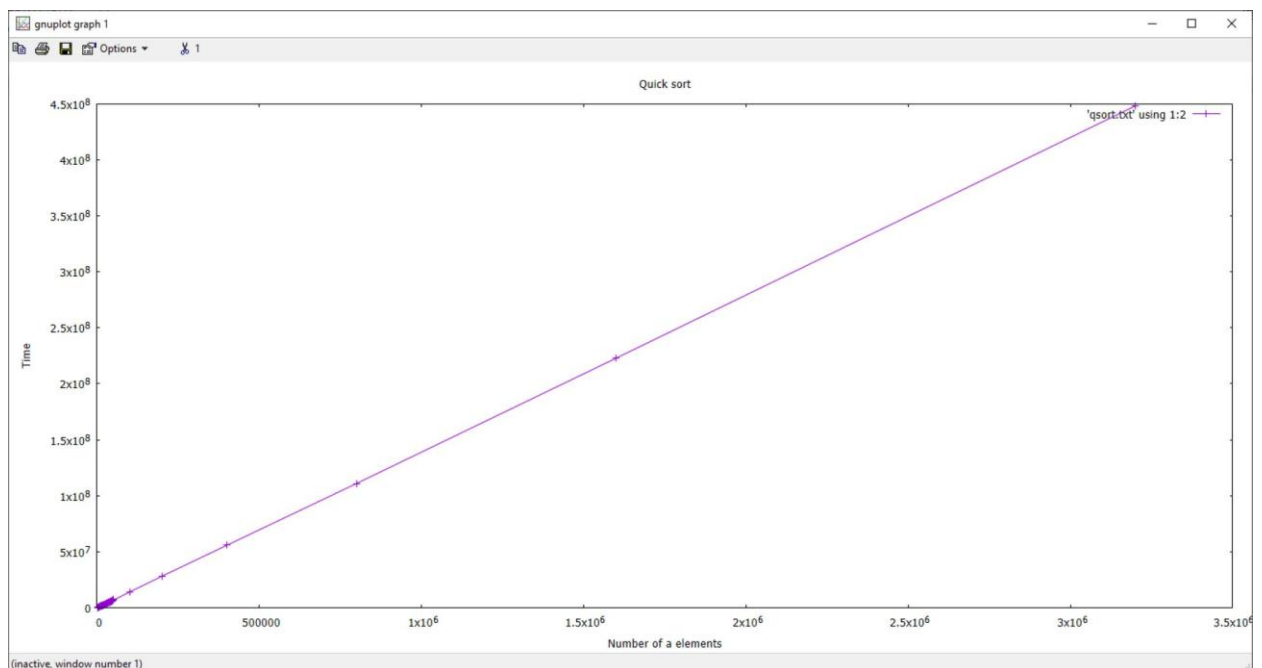
Далее создаётся массив необходимого размера и удаляется он только в конце цикла.

Массив заполняется случайными значениями и для каждой функции сортировки массива выводится время его сортировки.

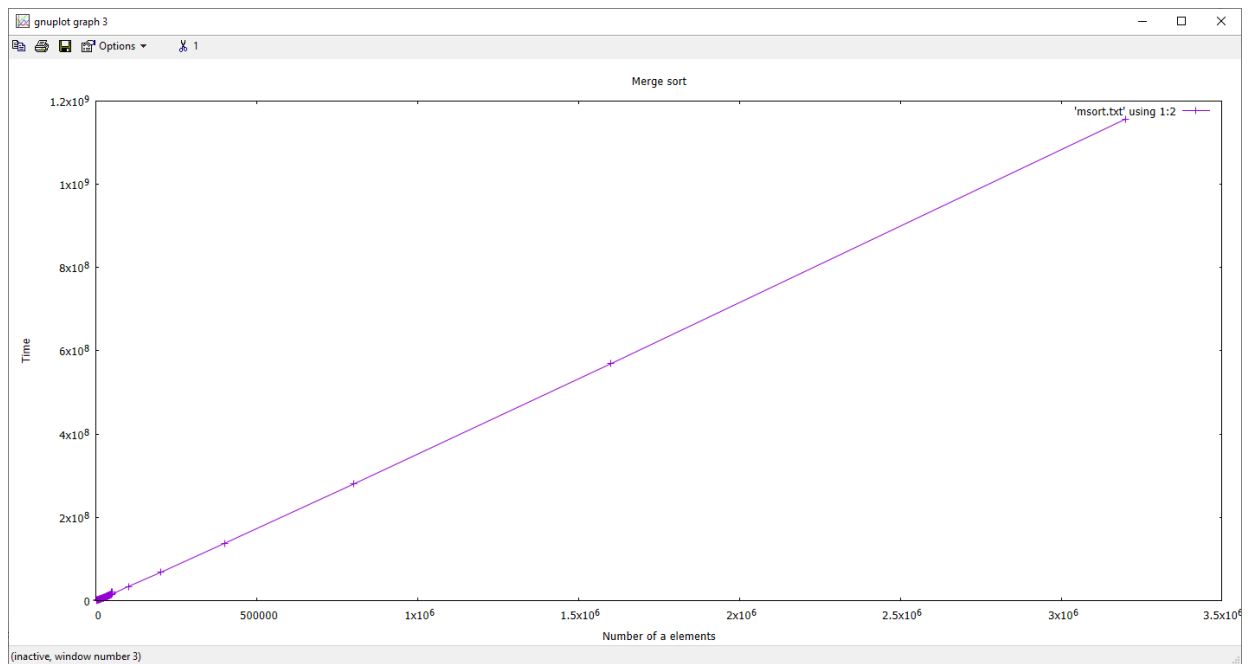
Результаты работы программы

Для приведённый графиком брался 1 замер.

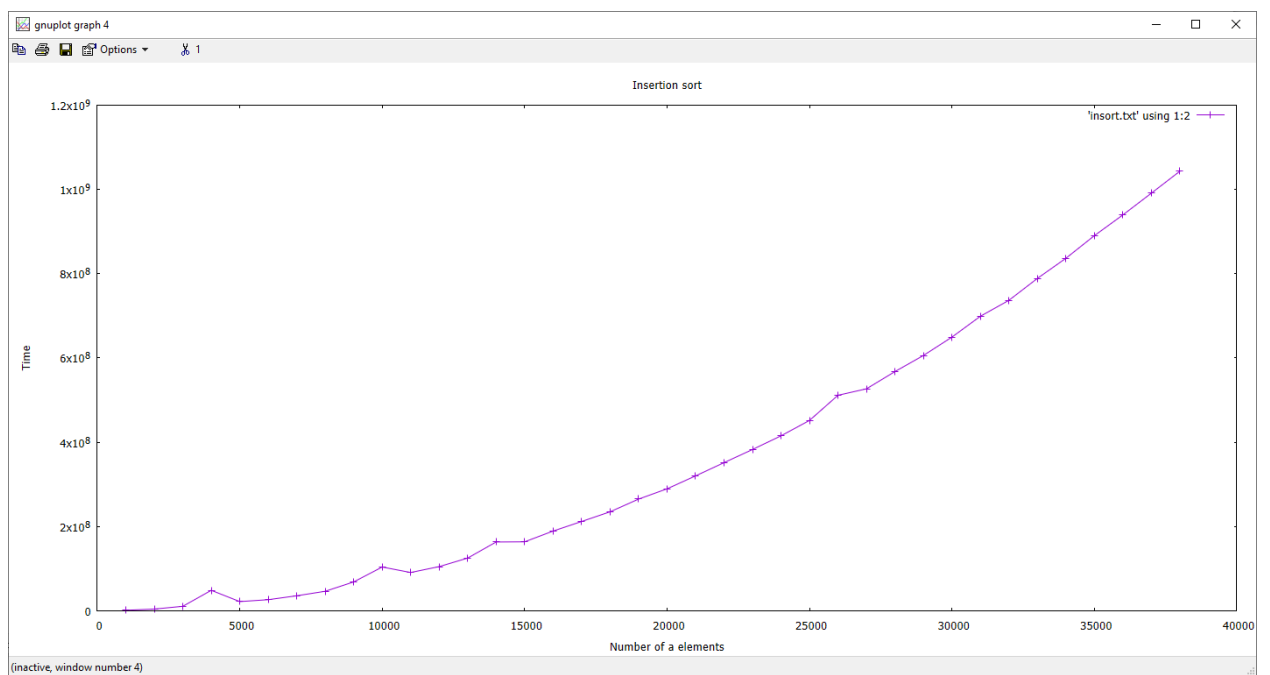
QuickSort:



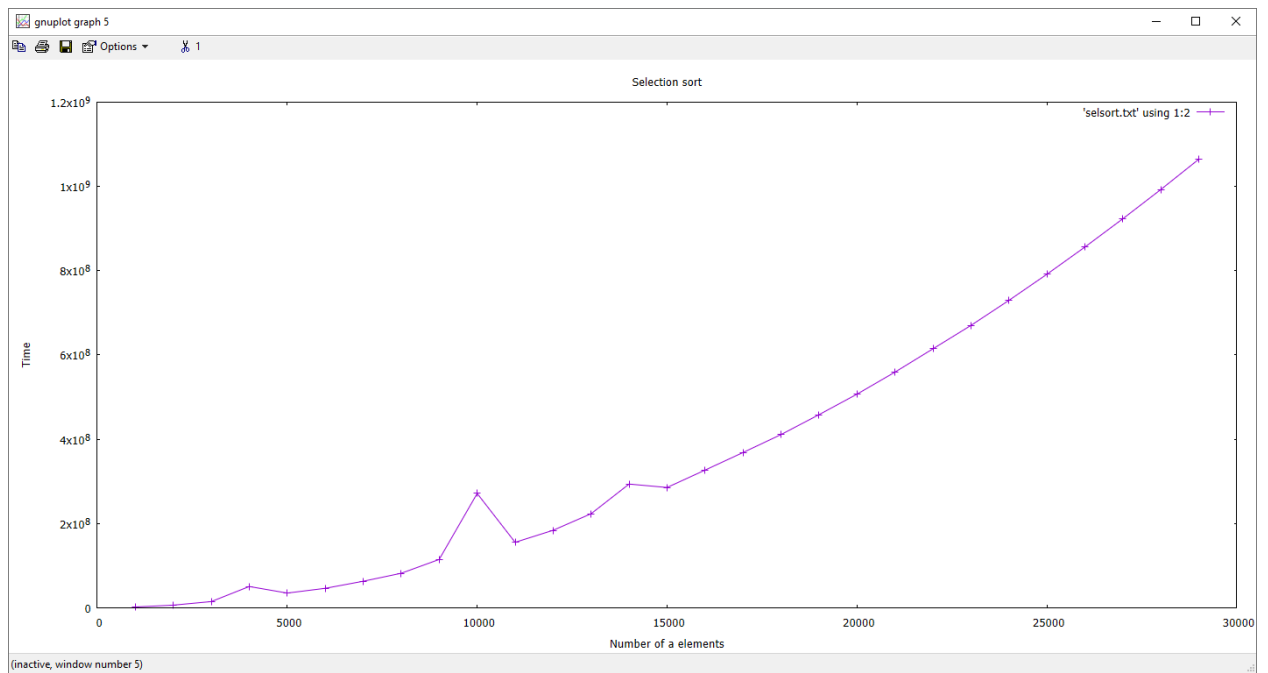
MergeSort:



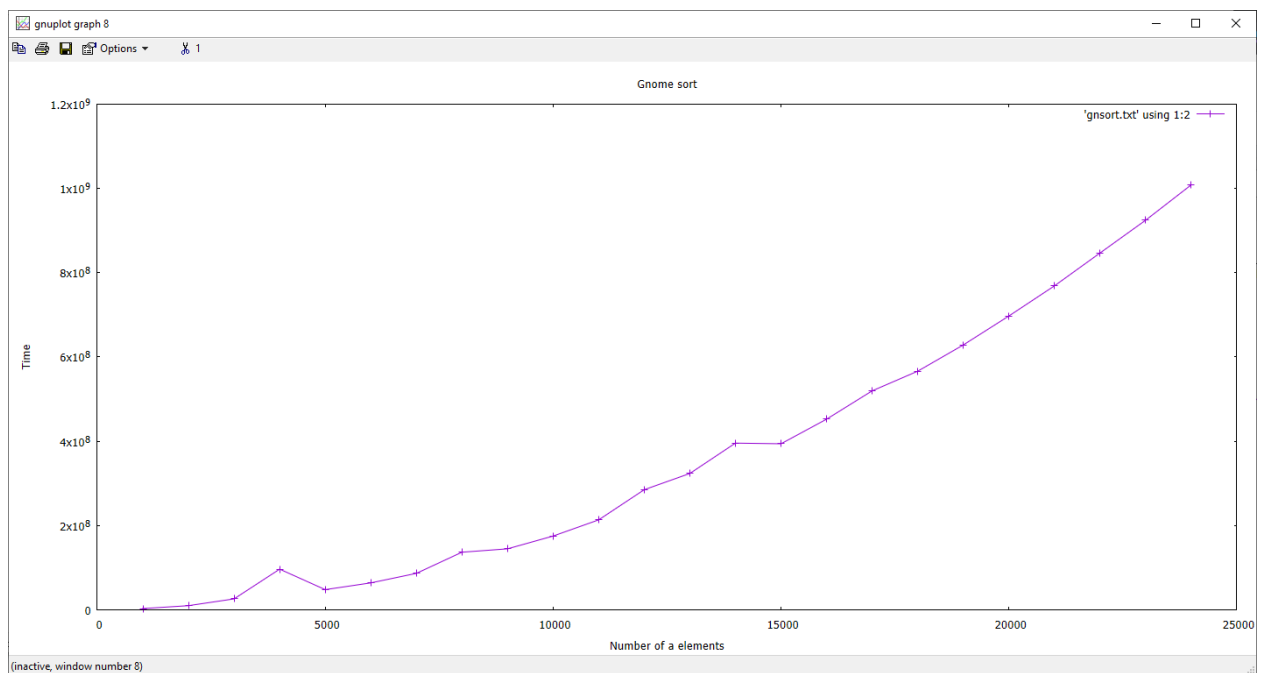
InsertionSor:



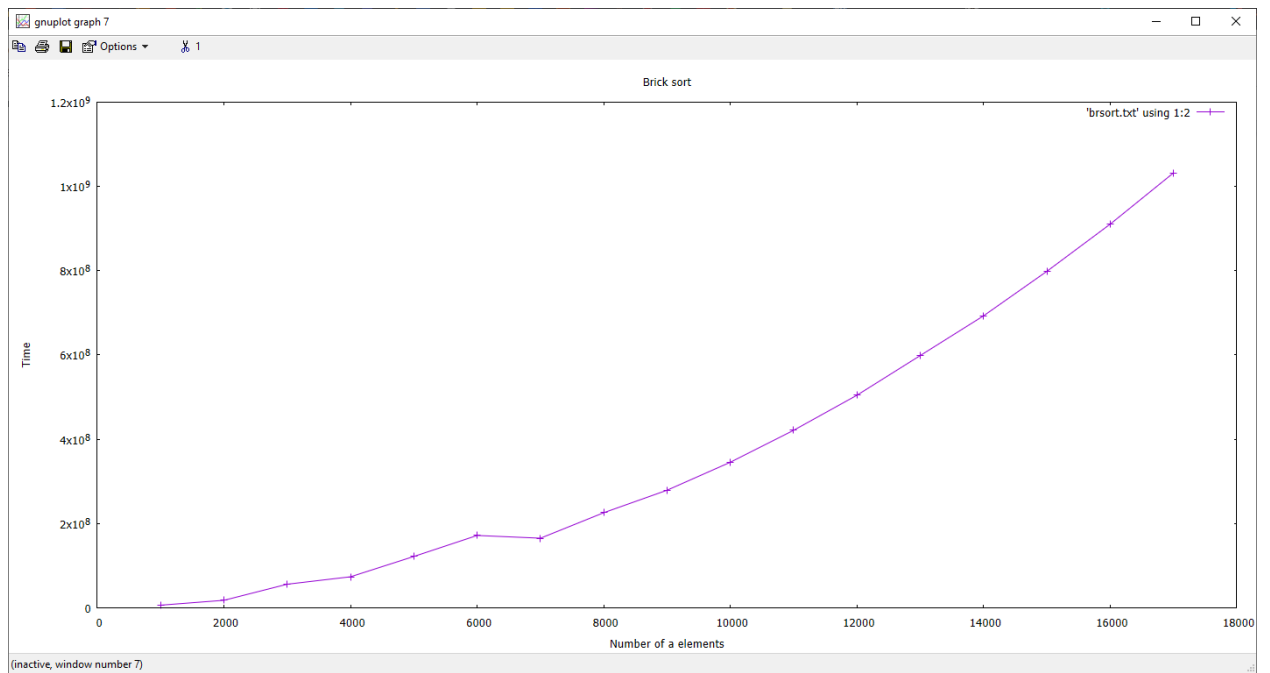
SelectionSort:



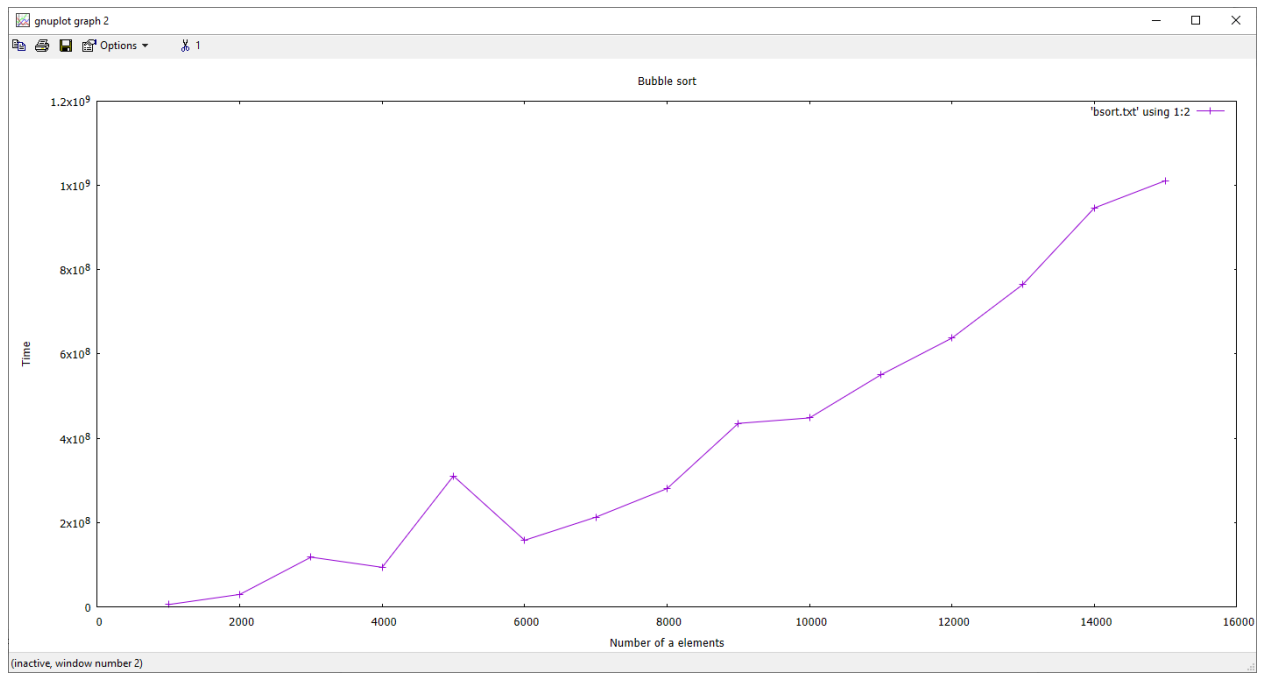
HeapSort:



CombSort:



BubbleSort:



Список от самой медленной сортировки к самой быстрой:

1. Сортировка пузырьком
2. Сортировка расческой
3. Пирамидальная сортировка
4. Сортировка выбором
5. Сортировка вставками
6. Сортировка слиянием
7. Быстрая сортировка