Государственное образовательное учреждение высшего профессионального образования
«Московский Государственный Технический Университет им Н. Э. Баумана.»

Отчет
По лабораторной работе №3
По курсу «Анализ Алгоритмов»

Тема: «Исследование алгоритмов сортировки»

Студент: Губайдуллин Р.Т.
Группа: ИУ7-51

Преподаватели: Волкова Л.Л.
Строганов Ю.В.

Постановка задачи

Реализовать и сравнить алгоритмы сортировки:
1. Быстрая сортировка
2. Гномья сортировка
3. Сортировка пузырьком

Провести временные эксперименты. Рассчитать сложность алгоритмов.

Алгоритмы

Быстрая сортировка

Быстрая сортировка, сортировка Хоара, часто называемая qsort - широко известный алгоритм сортировки, разработанный английским информатиком Чарльзом Хоаром. QuickSort является существенно улучшенным вариантом алгоритма сортировки с помощью прямого обмена, известного в том числе своей низкой эффективностью.
Общая идея состоит в следующем:
- Выбрать им массива элемент, называемый опорным. Это может быть любой из элементов массива. От выбора опорного элемента не зависит корректность алгоритма, но в отдельных случаях может сильно зависеть его эффективность.
- Сравнить все остальные элементы с опорным и переставить их в массиве так, чтобы разбить массив на три непрерывных отрезка, следующие друг за другом: «меньше опорного», «равные», «большие».
- Для отрезков «меньших» и «больших» значений выполнить рекурсивно ту же последовательность операций, если длина отрезка больше единицы.

Листинг: алгоритм быстрой сортировки

```
void quicksort(double *A, int len) {
    If (len < 2)
        return;

    int pivot = A[len/2];

    int i, j;
    for (int I = 0; j = len - 1; i++, j—) {
        while (A[i] < pivot)
            i++;
        while (A[j] > pivot)
            j—;

        if (i >= j)
            break;

        int temp = A[i];
        A[i] = A[j];
        A[j] = temp;
    }

    quicksort(A, i);
    quicksort(A + i, len - i);
}
```

Средний случай: N log(N)
Худший случай: N^2

Гномья сортировка

Гномья сортировка - алгоритм сортировки, похожий на сортировку вставками, но в отличие от последней перед вставкой на нужное место происходит серия обменов, как в сортировке пузырьком.

Алгоритм находит первое место, где два соседних элемента стоят в неправильном порядке и меняет их местами. Он пользуется тем фактом, что обмен может породить новую пару, стоящую в неправильном порядке, только до или после переставленных элементов. Он не допускает, что элементы после текущей позиции отсортированы, таким образом, нужно проверить позицию до переставленных элементов.

Листинг: алгоритм гномьей сортировки

```
void Array::GnomeSort(double *A, int N) {
    int i = 0;
    while (i < N) {
        if (i == 0 || A[i - 1] <= A[i])
            ++i;
        else {
            double Temp = A[i];
            A[i] = A[i - 1];
            A[i - 1] = Temp;
            --i;
        }
    }
}
```

Лучший случай: O(n)
Средний случай: O(n^2)
Худший случай: O(n^2)

Сортировка пузырьком

Сортировка простыми обменами, сортировка пузырьком - простой алгоритм сортировки. Алгоритм состоит из повторяющихся проходов по сортируемому массиву. За каждый проход элементы последовательной сравниваются попарно и, если порядок в паре неверный, выполняется обмен. Проходы по массиву повторяются *N - 1* раз и до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает - массив отсортирован. При каждом проходе алгоритма по внутреннему циклу, очередной наибольший элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим «элементом», а наименьший элемент перемещается на одну позицию к началу массива («всплывает» до нужной позиции, как пузырек в воде).

Листинг. Алгоритм сортировки пузырьком

```
void Array::BubbleSort(double *A, const int n) {
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            If (A[j] > A[j+1] {
                double buf = A[j];
                A[j] = A[j + 1];
                A[j + 1] = buf;
            }
        }
    }
}
```

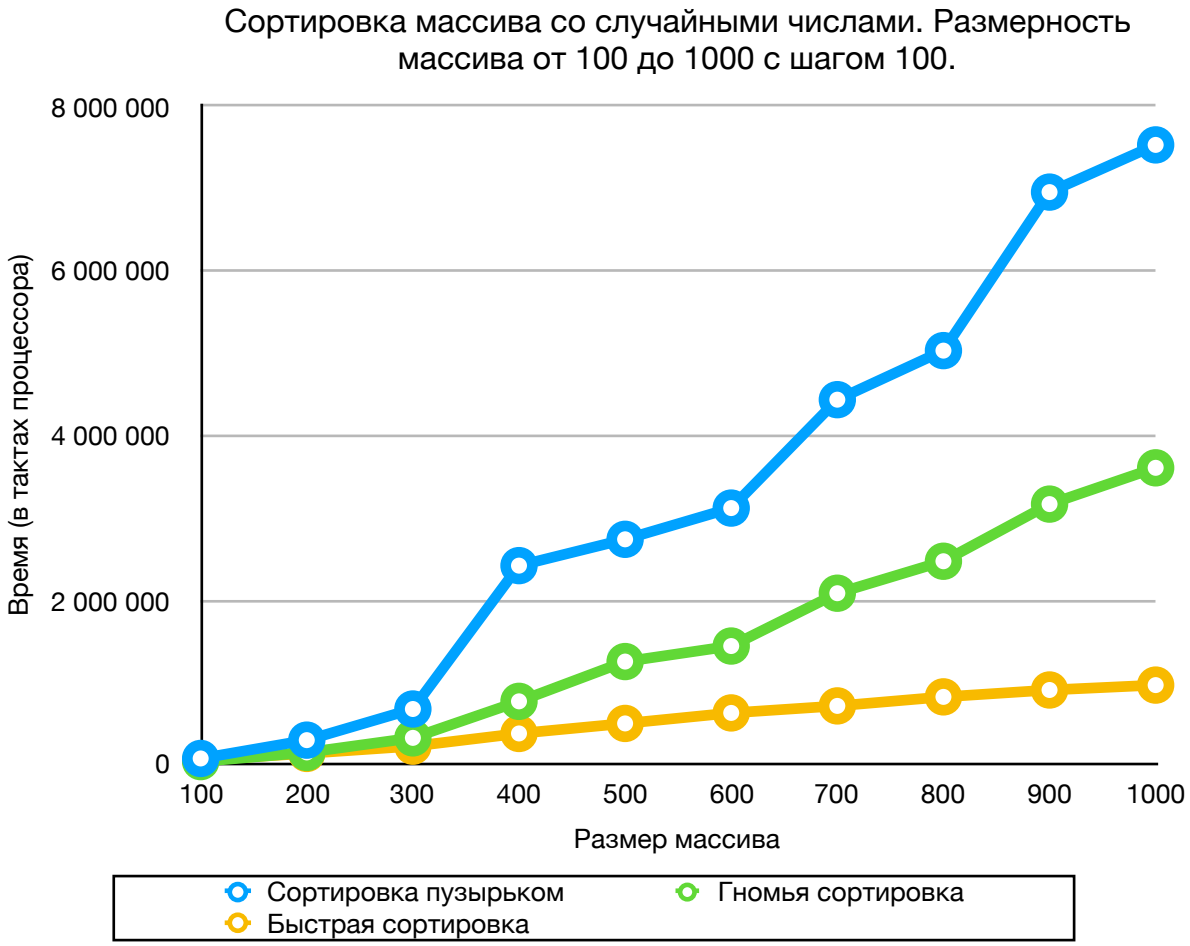
Лучший случай: 1 + N*(2 + (1 + N*(2 + 4))) = 1+ 7N + 6N^2
Худший случай: 1 + N*(2 + 1 + N*(2 + 4 + (2 + 4 + 3))) = 1 + 3N + 13N^2
Средний случай: $O(n^2)$

Тесты

Сортировка массива размерностью 100…1000 с шагом 100

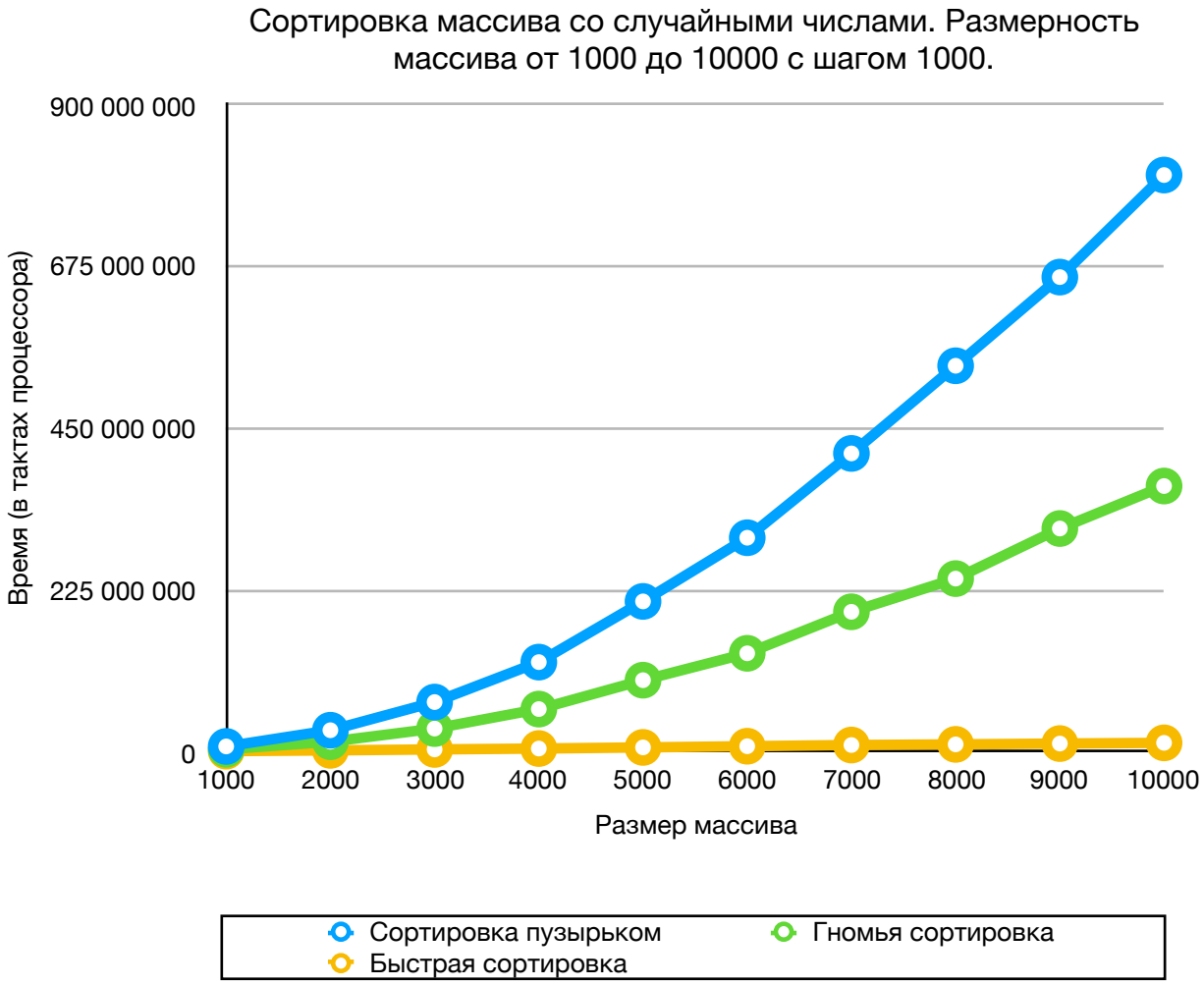
Размер массива	Сортировка пузырьком	Гномья сортировка	Быстрая сортировка
100	75587	37307	56752

200	298157	148801	136775
300	676407	325739	224475
400	2418094	769381	379330
500	2737784	1252634	501121
600	3117243	1441330	628799
700	4434127	2082991	714047
800	5029957	2471423	823774
900	6955668	3165491	907377
1000	7529102	3606227	966816



Размер массива	Сортировка пузырьком	Гномья сортировка	Быстрая сортировка
1000	7529102	3606227	966816
2000	30067938	14103311	2043593
3000	69270548	32380937	3368768
4000	124853660	59502406	4600388

5000	209354049	99715460	6169312
6000	297951047	137119369	7816022
7000	415275572	194881563	9485529
8000	537267742	241069759	10389890
9000	660648867	310771799	11522446
10000	802670059	369912447	12509303



Выводы

В результате проведенных экспериментов было получено:

1. Быстрая сортировка менее эффективна на массиве небольшого размера, однако, с возрастанием массива, её эффективность резко увеличивается.
2. Сортировка пузырьком оказалась самым неэффективным.

Заключение

В ходе лабораторной работы были изучены и сравнены 3 сортировки: быстрая, гномья и пузырьком.