Федеральное государственное бюджетное образовательное учреждение высшего образования

Московский государственный технологичесий университет «СТАНКИН» ФГБОУ ВО МГТУ «СТАНКИН»

Кафедра «Управление и информатика в технических системах Дисциплина: «Алгоритмы и структуры данных»

#### Отчет по лабораторным работам

Выполнил: студент группы ИДБ-23-13

<Таган А.М.> Преподаватель: доцент Евдокимов С.А.

Москва, 20 24г.

**Оглавление**

[Сортировка простыми вставками 3](#_bookmark0)

[Постановка задачи 3](#_bookmark1)

[Описание алгоритма 3](#_bookmark2)

[Графики времени сортировки 5](#_bookmark3)

[Выводы 6](#_bookmark4)

[Литература 6](#_bookmark5)

# Сортировка простыми вставками

## Постановка задачи

Написать программу сортировки постредством простого выбора и сравнить ее время выполнения со стандартной функцией qsort. Для сравнения вычисляем время выполнения функции сортировки на массиве целых чисел следующих размеров: 16, 100,500,1000,5000.

Для работы программы сортировки генерируем случайные массивы целых чисел:

* для массивов размерами <= 500 элементов числа в интервале [100, 1000 );
* для массивов размерами > 50NN г. 0NN г. элементов числа в интервале [1000, 10000);

Для проверки работы программы исходный не сортированный массив выводим в файл с именем d<число элементов>.txt, а сортированный массив выводим в файл с именем d<число элементов>-sorted.txt.

Время работы функции сортировки рассчитываем с точностью до тысячных долей миллисекунд. Например, 0.002 , 0.018, 0.377, 1.380, 35,806 мсек.

Для оценки поведения функции сортировки создаем графики в зависимости от числа элементов (N) в массиве:N) в массиве:

* график функции N2;
* график функции Nlog2N;
* график времени работы стандартной функции qsort;
* график времени работы функции сортировки простыми вставками; Графики создаем в табличном процессоре LibreOffice Calc.

## Описание алгоритма

Сортировки простыми вставками относится к классу сортировки вставками.

Основной алгоритм состоит в следующем:

1. Имеется массив целых чисел (N) в массиве:arK) из N элементов от 0 до N-1.) из N элементов от 0NN г. до N-XX-ХХ1.
2. В начале массива имеется уже отсортированная часть от 0NN г. до i, где i начинается с 1;
3. В цикле по всем оставшимся элементам (N) в массиве:j) от i+1 до N-1 выполняем вставку этого) от i+1 до N-XX-ХХ1 выполняем вставку этого элемента (N) в массиве:arK) из N элементов от 0 до N-1.[j) от i+1 до N-1 выполняем вставку этого]) в нужное место отсортированной части массива. Поиск места) в нужное место отсортированной части массива. Поиск места вставки начинаем с хвоста отсортированной части массива.

Формальная запись алгоритма посредством диаграмм Насси-Шнейдермана показана на рис.1.

Результат работы программы на массиве из 16 чисел показан ниже: Командная строка:

testQSort d16.txt

Результат работы программы:

Сортируется массив из 16 чисел

|  |  |  |
| --- | --- | --- |
| 293, | 180, | 389, 639, 928, 107, 324, 723, 422, 840, 206, 599, 445, |
| 359, | 841, | 378 |
| 107, | 180, | 206, 293, 324, 359, 378, 389, 422, 445, 599, 639, 723, |
| 840, | 841, | 928 |

Время сортировки 0.003 mсек.сек.

Сортированный массив записывается в файл "d16-s.txt"d16-s.txt"d16-sorted.txt"



**insertionSort(vector<**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | | | |
|  |  | | |
|  | | |
|  |  | |
|  | | |
|  | | |
|  | | |
|  | |  |
|  | | | |

Рис.1. Запись алгоритма сортировки простыми вставками посредством диаграмм Насси -XX-ХХ Шнейдермана

Исходный текст на С/С++ функции сортировки, составленный по алгоритму на рис.1 показан ниже:

void sortS(int\* arK, int N)

// Сортировка простыми вставками;

// arK - массив целых чисел из N элементов от 0 до N-1;

{

// цикл по всем вставляемым элементам массива for (int j = 1; j < N; ++j)

{

int K = arK[j]; // элемент для вставки;

// найти место вставки элемента (K);

int i = j-1; // номер последнего элемента сортированной части массива;

while (i >= 0 && arK[i] > K)

{

// выполнить сдвиг элемента вправо; arK[i+1] = arK[i];

--i;

}

// место вставки найдено (arK[i+1]);

// проверить необходимость вставки; if (++i < j)

{

// вставить элемент; arK[i] = K;

}

}

}

## Графики времени сортировки

Для оценки поведения функции сортировки создаем графики в зависимости от числа элементов (N) в массиве:N) в массиве:

* график функции N2;
* график функции Nlog2N;
* график времени работы стандартной функции qsort;
* график времени работы функции sortS сортировки простыми вставками;
* график времени работы функции sortSp сортировки простыми вставками с использованием функции memmove;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | **мксек.** | | |
| **N** | **N2** | **N\*log2N** | **qsort** | **sortS** | **sortSp** |
| 16 | 256 | 64 | 3 | 2 | 3 |
| 100 | 10000 | 664 | 15 | 18 | 40 |
| 500 | 250000 | 4483 | 78 | 377 | 240 |
| 1000 | 1000000 | 9966 | 147 | 1380 | 864 |
| 5000 | 25000000 | 61439 | 811 | 35806 | 21162 |

Рис.2. Данные для создания графиков времени сортировки

100000000

### Логарифмическая шкала по оси Y

10000000



1000000

100000

10000

1000

N2

 N\*log2N  qsort  sortS  sortSp

100

10

1 N

16 100 500 1000 5000

Рис.3. Графики времени сортировки различными алгоритмами

## Выводы

Из анализа графиков на рис.3 следует:

1. Сортировка простыми вставками имеет временную сложность O(N) в массиве:N2);
2. На малом числе элементов массива (N) в массиве:примерно меньших 50NN г. ) время сортировки простыми вставками лучше, чем у стандартной функции qsort;
3. При числе элементов в массиве примерно меньших 30NN г. 0NN г. время сортировки с использованием функции memmove хуже, а при числе элементов больших 30NN г. 0NN г. немного лучше классического алгоритма сортировки простыми вставками. Это изменение алгоритма не меняет временную сложность в O(N) в массиве:N2);

## Литература

1. Д.Кнут. Искуство программирования для ЭВМ. Сортировка и поиск. Том 3. М.:Мир, Москва, 1978. — 844 с.