

МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное*

*учреждение высшего образования*

***«МИРЭА – Российский технологический университет»***

# РТУ МИРЭА

Отчет по выполнению практического задания № 2

Вариант 1

**Тема:**

**ЭМПИРИЧЕСКИЙ АНАЛИЗ СЛОЖНОСТИ ПРОСТЫХ**

**АЛГОРИТМОВ СОРТИРОВКИ»**

Дисциплина: «Структуры и алгоритмы обработки данных»

Выполнил студент: Тринеев П.С.

ИКБО-32-22



Номер группы



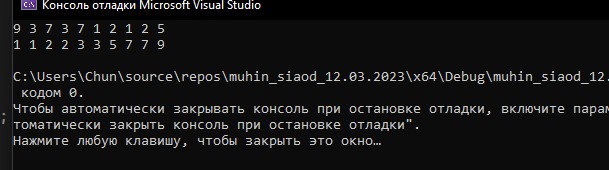
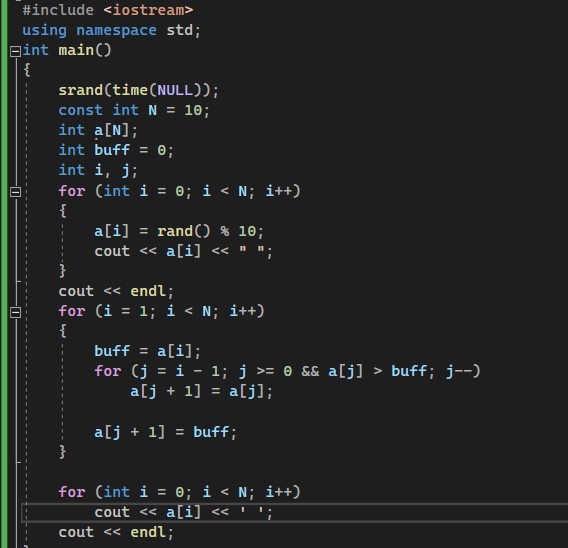
Фамилия И.О.



Группа:

**Цель.** Актуализация знаний и приобретение практических умений по эмпирическому определению вычислительной сложности алгоритмов.

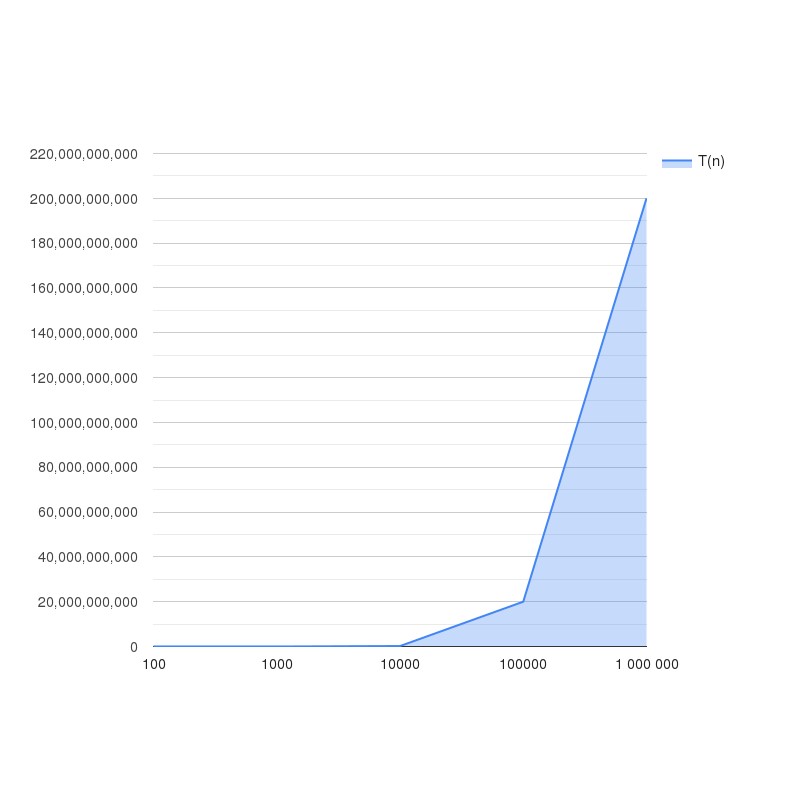
**Задание 1.** Оценить эмпирически вычислительную сложность алгоритма простой сортировки на массиве, заполненном случайными числами.



T(n) = 2n^2 + 7n + 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | Т(n), мс | Лучший | Средний | Худший |
| 100 | 20 703 | 20 703 | 428 614 | 428 614 |
| 1000 | 2 007 003 | 2 007 003 | 4 028 061 042 | 4 028 061 042 |
| 10000 | 200 070 003 | 200 070 003 | 40 028 006 100 420 | 40 028 006 100 420 |
| 100000 | 20 000 700 003 | 20 000 700 003 | 400 028 000 610  004 200 | 400 028 000 610 004 200 |
| 1 000 000 | 200 007 000 003 | 200 007 000 003 | 4 000 028 000 610 004 200 | 4 000 028 000 610 004 200 |

В конечном итоге данный алгоритм оказался сложнее предыдущего поначалу не существенно, но чем больше n, тем разница больше



## **ВЫВОД**

Были подсчитаны сложности алгоритма и рассчитаны лучшие и худшие случаи. Сравнив два алгоритма, выяснили что пузырьком проще чем вставками, также то, что количество секунд на алгоритм сильно увеличивался в зависимости от случая.

**Задание 2**

Оценить вычислительную сложность алгоритма простой сортировки в наихудшем и наилучшем случаях.

**Наихудший случай**

|  |  |
| --- | --- |
| **n** | **T(n), мс** |
| 100 | 0.0128 |
| 1000 | 1.2138 |
| 10000 | 118.9 |
| 100000 | 11673.5 |
| 1000000 | ~188027605474 (~2100 дней) |

**Наилучший случай**

|  |  |
| --- | --- |
| **n** | **T(n), мс** |
| 100 | 0.0129 |
| 1000 | 1.1963 |
| 10000 | 116.883 |
| 100000 | 11639.1 |
| 1000000 | ~195634502350 (~2200 дней) |

**Вывод**

Алгоритм почти не зависит от изначального упорядочивания массива.

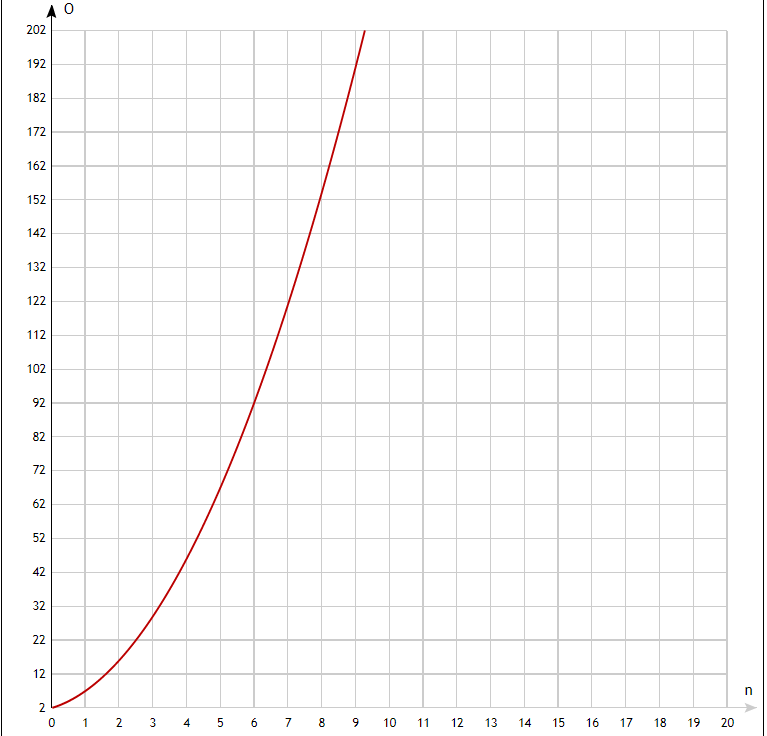
O(n2) во всех случаях.

Задание 3. Оценить эмпирически вычислительную сложность алгоритма простой сортировки на массиве, заполненном случайными числами.

Худшим случаем для алгоритма является случай, когда все элементы массива расположены в порядке убывания, тогда нужно менять местами каждый элемент кроме последнего.

Лучшим является случай, когда не нужно менять местами ни один элемент-когда все элементы уже упорядочены.

**Сложность алгоритма insert sort:**

Худший случай:

O(n) = 4+7n+4n2

Лучший случай:

O(n) = 4+3n+2n2

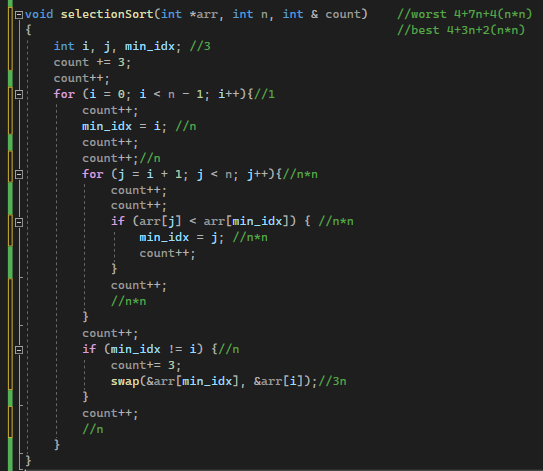
Средний случай:

O(n) = 4+5n+3n2

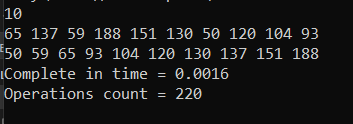
В общем случае можно описать

как O(n2).

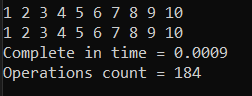
**Код алгоритма:**

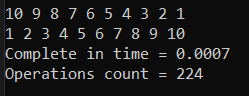


**Результаты тестирования:**

При размере массива 10

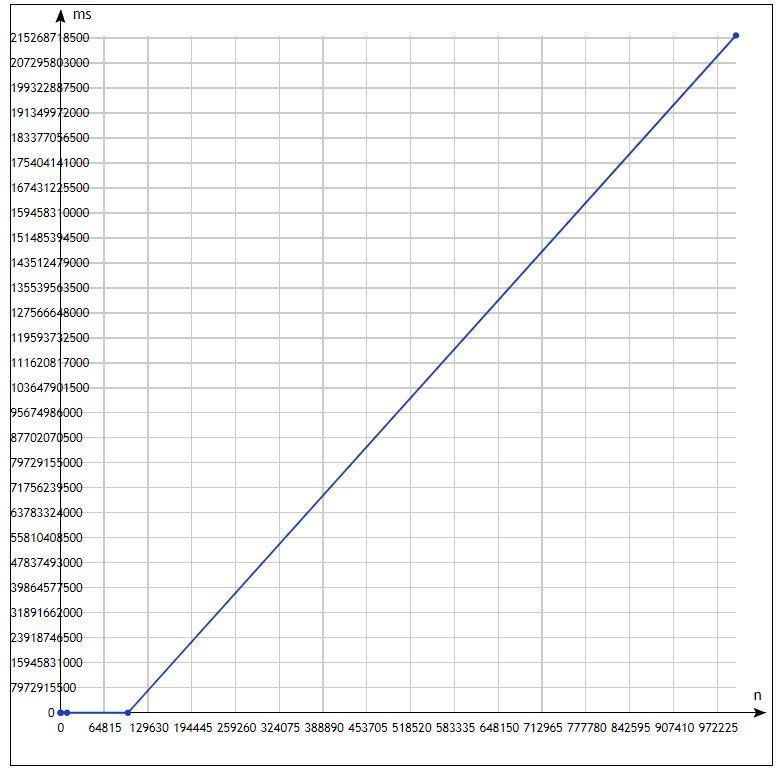
Случайное заполнение

Лучший случай



Худший случай

|  |  |
| --- | --- |
| **n** | **T(n), мс** |
| 100 | 0.0223 |
| 1000 | 1.9395 |
| 10000 | 147.723 |
| 100000 | 14633.1 |
| 1000000 | ~216060601000 (2.5 дня) |

**График временной функции:**

**Вывод:**

Алгоритм вполне эффективен на небольших размерах массивов.