

МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное*

*учреждение высшего образования*

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

# РТУ МИРЭА

Отчет по выполнению практического задания № 3

Вариант 8

**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:

**Практическая работа № 3**

Определение эффективного алгоритма сортировки на основе эмпирического и асимптотического методов анализа

**Цель:** получить навыки по анализу вычислительной сложности алгоритмов сортировки и определению наиболее эффективного алгоритма.

**Задание 1. Эмпирическая оценка эффективности алгоритмов**

Турнирная сортировка

|  |  |
| --- | --- |
| **n** | **T(n)** |
| 100 | 2,408 сек. |
| 1000 | 5,207 сек. |
| 10000 | 6,225 сек. |
| 100000 | 13,346 сек. |
| 1000000 | - |

Емкостная сложность алгоритма: ~1 Mб

Простое слияния

|  |  |
| --- | --- |
| **n** | **T(n)** |
| 100 | 2,184 сек. |
| 1000 | 4,684 сек. |
| 10000 | 17,326 сек. |
| 100000 | 2,02 мин. |
| 1000000 | 19,23 мин. |

Емкостная сложность алгоритма: ~5 Mб

Пузырьковая сортировка

|  |  |
| --- | --- |
| **n** | **T(n)** |
| 100 | 0.0179 сек. |
| 1000 | 2.2076 сек. |
| 10000 | 112.884 сек. |
| 100000 | 19467.1 сек. |
| 1000000 | ~190000000000 (~2000 дней) |

**Задание 2.** Асимптотический анализ сложности алгоритмов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм | Асимптотическая сложность алгоритма | | | |
| Наихудший случай | Наилучший случай | Средний случай | Ёмкостная сложность |
| Простой | O(n2) | O(n2) | O(n2) | 760 КБ |
| Усовершен-ствованный | O(n2) | O(n2) | O(n2) | 1 МБ |
| Быстрый | O(n2) | O(n\*2log2n) | O(n\*log2n) | 2 МБ |

**Сложность быстрой сортировки:**

В наиболее сбалансированном варианте при каждой операции разделения массив делится на две одинаковые (плюс-минус один элемент) части, следовательно, максимальная глубина рекурсии, при которой размеры обрабатываемых подмассивов достигнут 1, составит log n. В результате количество сравнений, даёт общую сложность алгоритма O(n\*logn)

В худшем случае. В самом несбалансированном варианте каждое разделение даёт два подмассива размерами 1 и n-1, то есть при каждом рекурсивном вызове больший массив будет на 1 короче, чем в предыдущий раз. Такое может произойти, если в качестве опорного на каждом этапе будет выбран элемент либо наименьший, либо наибольший из всех обрабатываемых. При простейшем выборе опорного элемента — первого или последнего в массиве, — такой эффект даст уже отсортированный (в прямом или обратном порядке) массив, для среднего или любого другого фиксированного элемента «массив худшего случая» также может быть специально подобран. В этом случае потребуется n-1 операций разделения, а общее время работы составит O(n^2) операций.

**Вывод**: Быстрая сортировка на порядки быстрее других, крайне

эффективна на любых значениях массива.