Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЕТ О ПРАКТИЧЕСКОЙ** **РАБОТЕ № 5**

Cортировка.  
Вариант № 10

Преподаватель \_\_\_\_\_\_\_\_ Матковский И.В.

подпись, дата

Студент КИ18-09б, 031830645 \_\_\_\_\_\_\_\_ Котов С.А.

подпись, дата

Красноярск 2019

**1 Задание**

Разработать программы, демонстрирующие два заданных метода сортировки; сравнить их эффективность.

Первый метод сортировки: «Гномья сортировка», второй метод сортировка: «Быстрая сортировка».

**2 Код программы**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | vector<int> gnomeSort(vector<int> sequence) {  for (int i = 0; i + 1 < sequence.size(); ++i) {  if (sequence[i] > sequence[i + 1]) {  int temp = sequence[i];  sequence[i] = sequence[i + 1];  sequence[i + 1] = temp;  if (i != 0) i -= 2;  }  }  return sequence;  }  void quickSort(vector<int> & sequence, int low, int high) {  int i = low;  int j = high;  int x = sequence[(low + high) / 2];  do {  while (sequence[i] < x) ++i;  while (sequence[j] > x) --j;  if (i <= j) {  int temp = sequence[i];  sequence[i] = sequence[j];  sequence[j] = temp;  i++; j--;  }  } while (i < j);  if (low < j) quickSort(sequence, low, j);  if (i < high) quickSort(sequence, i, high);  } |

**3 Теоретические оценки сложности алгоритмов**

Временная сложность «Гномьей сортировки» равна O((sequence.size())^2), а у «Быстрой сортировки» временная сложность равна O(sequence.size()\*log(sequence.size())).

«Гномья сортировка» с помощью одного цикла и двух условий находит первое место, где два соседних элемента стоят в неправильном порядке и меняет их местами.

«Быстрая сортировка» выбирает из массива некоторый опорный элемент. Сравнивает все остальные элементы массива с опорным и переставляет их в массиве так, чтобы разбить его на три непрерывных отрезка, следующих друг за другом: «элементы меньшие опорного», «равные» и «большие». Для отрезков «меньших» и «больших» значений выполняет рекурсивно ту же последовательность операций, если длина отрезка больше единицы.

**4 Экспериментальные оценки сложности алгоритмов**

|  |  |  |
| --- | --- | --- |
| sequence.size() | Время 1, сек | Время 2, сек |
| 100 | 0 | 0 |
| 1000 | 0 | 0 |
| 5000 | 16 | 0 |
| 10000 | 63 | 0 |