Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЕТ О ПРАКТИЧЕСКОЙ** **РАБОТЕ № 1**

Оценка сложности алгоритма.  
Вариант № 10

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

подпись, дата

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

подпись, дата

Красноярск 2019

**1 Задание**

Разработать для решения поставленной задачи два алгоритма с разными уровнями сложности.

В имеющейся последовательности чисел найти три максимальных элемента.

**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  34  35  36  37  38  39 | void fastSolution(vector<int> sequence) {  bool first;  vector<int> buffer;  buffer.push\_back(sequence[0]);  buffer.push\_back(sequence[1]);  buffer.push\_back(sequence[2]);  for (int i = 0; i < sequence.size(); i++) {  if (sequence[i] > buffer[0]) {  buffer[2] = buffer[1];  buffer[1] = buffer[0];  buffer[0] = sequence[i];  }  else if (sequence[i] > buffer[1]) {  buffer[2] = buffer[1];  buffer[1] = sequence[i];  }  else if (sequence[i] > buffer[2]) {  buffer[2] = sequence[i];  }  }  cout << endl << endl << "Три максимальный элемента последовательности: " << buffer[0] << " " << buffer[1] << " " << buffer[2] << endl;  }  void slowSolution(vector<int> sequence) {  for (int i = 0; i < sequence.size(); i++) {  for (int j = sequence.size() - 1; j > i; j--) {  if (sequence[j - 1] < sequence[j]) {  int temp = sequence[j - 1];  sequence[j - 1] = sequence[j];  sequence[j] = temp;  }  }  }  cout << endl << endl << "Три максимальный элемента последовательности: " << sequence[0] << " " << sequence[1] << " " << sequence[2] << endl;  } |

**3 Теоретические оценки сложности алгоритмов**

Временная сложность первого метода равно O(sequence.size()), а у второго метода O((sequence.size())^2).

Первый метод осуществляет перебор элементов и их сравнивание между собой, чтобы найти три максимальных элемента последовательности. Тогда как второй метод осуществляет сортировку «Пузырьком» по возрастанию, после чего выводит три первых элемента последовательности.

**4 Экспериментальные оценки сложности алгоритмов**

|  |  |  |
| --- | --- | --- |
| sequence.size() | Время 1, сек | Время 2, сек |
| 100 | 0 | 0 |
| 1000 | 0 | 1 |
| 10000 | 0 | 103 |
| 100000 | 0 | 1722 |