ГУАП

# КАФЕДРА №43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| старший преподаватель |  | 21.09.2023 |  | С.А. Рогачев |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1 |
| АНАЛИЗ СЛОЖНОСТИ АЛГОРИТМОВ |
| по курсу: АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4232 |  | 21.09.2023 |  | М.В.Куриш |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

1. **Цель работы:**

Целью работы является изучение методов и получение практических навыков анализа сложности алгоритмов.

1. **Задание на лабораторную работу:**

Используя память, пропорциональную n, хранить массив целых чисел A, содержащий n элементов.

Элементы массива A могут принимать случайные значения от -((n div 2) - 1) до (n div 2). То есть, если в массиве хранится 10 элементов, то эти элементы должны быть в диапазоне от -4 до 5.

Разработать алгоритм, который осуществляет заполнение массива A случайными значениями, и по выбору пользователя выполняет одну из двух функций. Состав выполняемых функций и требования к теоретической временной сложности этих функций определяется вариантом задания.

Вариант 22.



1. **Листинг программы:**

|  |
| --- |
| #include <iostream>  using namespace std;  int\* generateArray(int);  void decreaseTens(int \*, int);  void countPositive(int);  int main()  {  setlocale(LC\_ALL, "RUS");  int Asize;  cout << "Введите количество элементов массива: ";  cin >> Asize;  int\* A = generateArray(Asize);  int funNum = 0;  while (true) {  cout << "Функции\nФункция #1: 1\nФункция #2: 2\nЗавершение: 3\n";  cout << "Выберите номер функции: ";  cin >> funNum;  if (funNum == 1) {  decreaseTens(A, Asize);  }  else if (funNum == 2) {  int posCount = 0;  for (int i = 0; i < Asize; i++) {  if (A[i] > 0)  posCount++;  }  countPositive(posCount);  }  else if (funNum == 3) {  break;  }  else  cout << "Введено неверное значение";  }  }  int\* generateArray(int arrSize) {  int\* A = new int[arrSize];  int min = -((arrSize / 2) - 1);  int max = arrSize / 2;  cout << "Значения массива принадлежат [" << min << "; " << max << "]\n";  cout << "Создан массив: [";  for (int i = 0; i < arrSize; i++) {  A[i] = min + (rand()%(max - min + 1));  cout << A[i];  if (i != arrSize - 1)  cout << ", ";  }  cout << "]\n";  return A;  }  void decreaseTens(int\* arr, int sz) {  cout << "Обработанный массив: [";  for (int i = 0; i < sz; i++) {  if (arr[i] % 10 == 0)  arr[i] /= 10;  cout << arr[i];  if (i != sz - 1)  cout << ", ";  }  cout << "]\n";  }  void countPositive(int val) {  cout << "\nКоличество положительных чисел: " << val << endl;  } |

1. **Расчет пространственной и временной сложностей алгоритма:**

Алгоритм содержит следующие переменные:

* Одиннадцать переменных целого типа;
* Один массив размерностью n;

Значит, пространственная сложность алгоритма определяется следующим образом:



где Cint – константа, характеризующая объем памяти, отводимый под переменную целого типа.

Временная сложность алгоритма, определенная на основе анализа текста программы, реализующей этот алгоритм:







где Ki – константа, характеризующая время выполнения операций, помеченных I; t1, t2, talg – временные сложности функций decreaseTens, countPositive и всего алгоритма в целом, соответственно.

1. **Расчет теоретической пространственной и теоретической временной сложностей алгоритма:**

Теоретическая пространственная сложность алгоритма составляет:

****

Теоретическая временная сложность функций составляет:





1. **Вывод:**

На основе расчетов пунктов 4 и 5 можно сделать вывод, что был разработан алгоритм, характеристики которого соответствуют поставленному заданию.

Изучены методы анализа сложности алгоритмов и получены соответствующие практические навыки.