

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Санкт-Петербургский государственный университет
аэрокосмического приборостроения»

КАФЕДРА № 43

Преподаватель

Рогачев С.А.

Отчёт
по лабораторной работе №1
по дисциплине Структуры и алгоритмы
обработки данных
на тему: «АНАЛИЗ СЛОЖНОСТИ АЛГОРИТМОВ».

Работу выполнила
студентка гр. 4631

Попова Д.А.

Санкт-Петербург
2017

1.1 Цель работы

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

1.2 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. Варианты задания приведены в таблице 1.

Вычисление номера варианта:

$$36 \bmod 25 = 11$$

Вариант №11

11	Заменить все отрицательные значения элементов на модули их значений	$O(n)$
	Подсчитать количество элементов с нулевым значением	$O(1)$

1.3 Порядок выполнения работы

1) выбрать вариант задания из подраздела 1.2 в соответствии с требованиями; 2) изучить теоретический материал, изложенный в учебном пособии; 3) разработать и реализовать на языке программирования высокого уровня алгоритм, выполняющий требования задания; 4) рассчитать теоретические временную и пространственную сложности алгоритма; 5) написать отчет о работе; 6) защитить отчет.

1.4 Содержание отчета

Отчет должен содержать: 1) титульный лист; 2) цель работы; 3) вариант задания; 4) листинг программы, реализующей алгоритм; 5) расчет пространственной и временной сложностей алгоритма; 6) расчет теоретической пространственной и теоретической временной сложностей алгоритма; 7) выводы по работе.

2.1 Листинг программы

```
2  #include<iostream>
3  #include<time.h>

4  using namespace std;
5  const int n = 10;

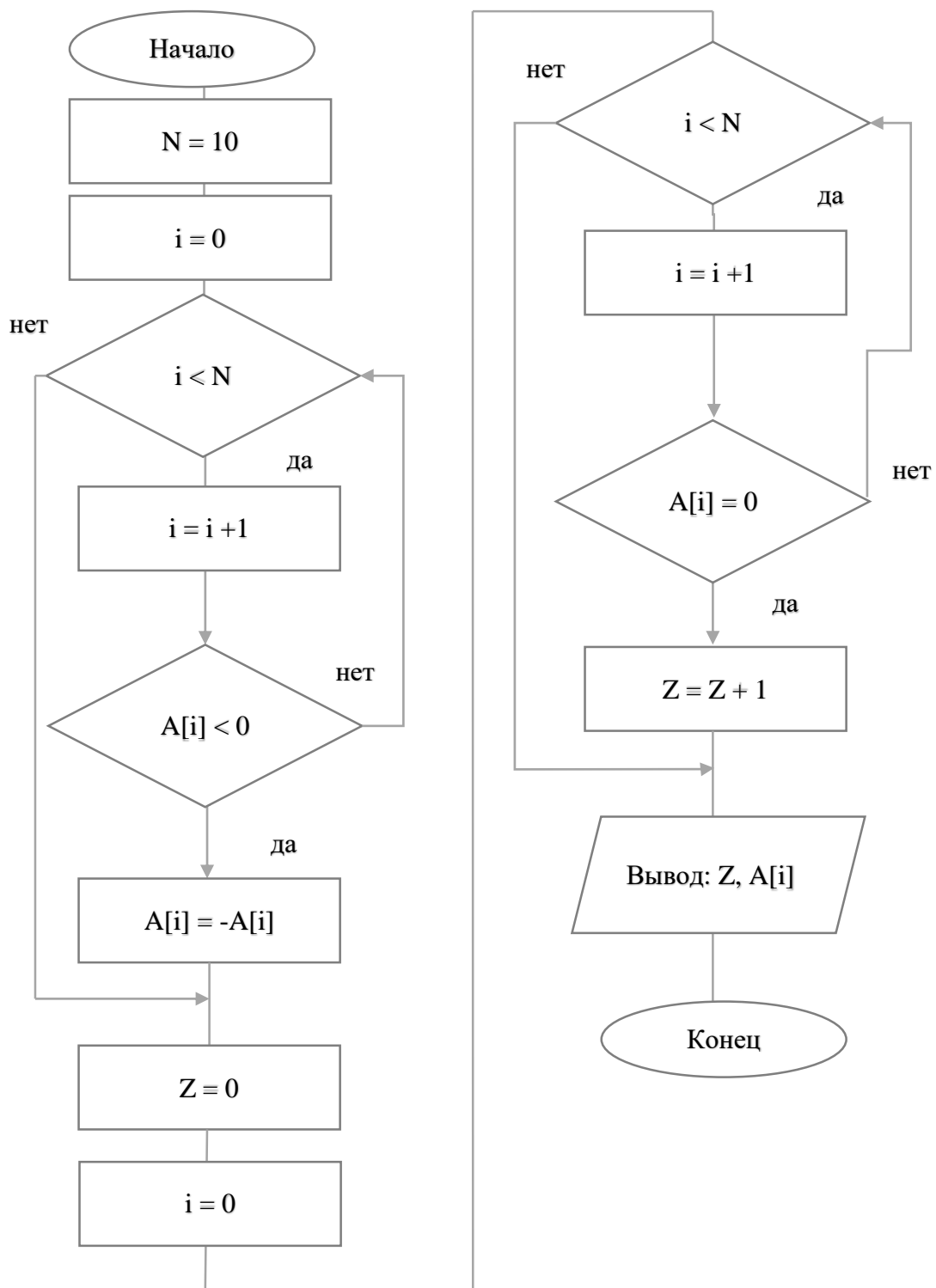
6  int module(int A[])
7  {
8  7.1 cout << "Результат:";
9  7.2 for (int i = 0; i<n; i++)
10 7.3 {
11 7.3.1 if (A[i]<0)
12 7.3.2 {
13 7.3.2.1 A[i] = -(A[i]);
14
15 7.3.3 }
16 7.3.4 cout << A[i]<<" ";
17 7.4 }

18 7.5 return 0;

19 }
20 void zero(int z)
21 {
22 20.1 cout << "Количество элементов с нулевым значением: " << z << endl;
23
24 }
25 int main()
26 {
27 26.1 setlocale(LC_ALL, "rus");
28 26.2 int a;
29 26.3 int z = 0;
30 26.4 int A[n];
31 26.5 srand(time(NULL));
32 26.6 cout << "Массив: ";
33 26.7 for (int i = 0; i<n; i++)
34 26.8 {
35 26.8.1 A[i] = rand() % n -((n/2)-1);
36 26.8.2 cout << A[i] << " ";
37
38 26.9 }
39 26.10 cout << endl;
40
41 26.11 for (int i = 0; i < n; i++)
42 26.12 {
43 26.12.1 if (A[i] == 0)
44 26.12.2 {
45 26.12.2.1 z++;
46 26.12.3 }
47 26.13 }
48
49 26.14 cout << "Для замены всех отрицательных значений элементов на модули их значений наберите
50 1" << endl;
51 26.15 cout << "Для подсчета количества элементов с нулевым значением наберите 2" << endl;
52 26.16 cout << "Для выхода из программы наберите любой другой символ" << endl;
53 26.17.1 cin >> a;
54 26.17.2 switch (a)
55 26.18 {
56 26.18.1 case 1: module(A); break;
57 26.18.2 case 2: zero(z); break;
58 26.18.3 default: exit(0);
59 26.18.4 }

60
61 26.19 system("pause");
62 26.20 return 0;
63 }
```

2.2 Реализующей алгоритм



3. Расчет теоретической пространственной и теоретической временной сложностей алгоритма

Теперь подсчитываем теоретические сложности алгоритма.

Разработанный алгоритм использует следующие данные: – одну константу; – один массив размерностью n ; – одну переменную целого типа.

Значит, пространственная сложность алгоритма определяется следующим образом:

$$v = C_{const} + n * C_{int} + 1 * C_{int}$$

где C_{const} – константа, характеризующая объем памяти, отводимый под константу;

C_{int} – константа, характеризующая объем памяти, отводимый под переменную целого типа.

Теоретическая пространственная сложность алгоритма составляет:

$$V(n) = O(v) = O(\max(O(C_{const}), O(n * C_{int}), O(1 * C_{int}))) = O(\max(O(1), O(n), O(1))) = O(n)$$

Временную сложность алгоритма определяем на основе анализа текста программы, реализующей этот алгоритм.

Согласно заданию на лабораторную работу, необходимо реализовать две функции, теоретические временные сложности которых не превышают заданных. Поэтому необходимо рассчитать теоретическую временную сложность функций, реализующих эти алгоритмы:

$$T_{module}(n) = K_{7.1} + n * K_{7.2} + K_{7.3.1} + K_{7.3.2.1} + K_{7.3.4}$$

$$T_{zero}(1) = K_{10.1}$$

$$T_{main}(n) = K_{13.1} + K_{13.2} + K_{13.3} + K_{13.4} + K_{13.5} + K_{13.6} + n * K_{13.7} + K_{14.1.1} + K_{14.1.2} + K_{14.3} + n * K_{14.4} + K_{14.5.1} + K_{14.5.2.1} + K_{15} + K_{16} + K_{17} + K_{17.1.1} + K_{17.1.2}$$

где K_i – константа, характеризующая время выполнения операций, помеченных i ; t_{sum} , t_{count} и t_{alg} – временные сложности функций `calculationAmount`, `countingDuplicates` и всего алгоритма в целом, соответственно. Теоретическая временная сложность функций составляет:

$$T_{sum}(n) = O(t_{sum}) = O(\max(O(n), O(1), O(n)))$$

4. Выводы по работе

В результате лабораторной работы я изучила методы и получила практические навыки анализа сложности алгоритмов. В задании на лабораторную работу требовалось: используя память, пропорциональную n , хранить массив целых чисел A , содержащий n элементов, где 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 определялся вариантом задания. Который я рассчитала перед началом лабораторной работы и получила значение равное одиннадцати. Ознакомившись с текстом задания пришло время обозначить порядок выполнения лабораторной работы, что было успешно сделано и не вызвало трудностей. Код программы был протестирован и сбоев в работе программы не нашлось. Сложности с выбором алгоритма были связаны с ограничениями в лабораторной работе наложенными на теоретическую временную сложность. Устранено это было с помощью цикла внедрённого в функцию `main()`. В результате, рассчитав теоретические пространственную и временную сложность функций ответы не превышали значений, которые были предоставлены мне в задании к лабораторной работе. На основе этих расчетов можно сделать вывод, что был разработан алгоритм, характеристики которого соответствуют поставленному заданию.

Завершив лабораторную работу мною был оформлен отчёт, который, может быть и не дочитан до конца, но значимости от этого своей не потеряет...