

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**"МИРЭА - Российский технологический университет"**

**РТУ МИРЭА**

Институт информационных технологий (ИТ)

Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1**

**по дисциплине**

**«Структуры и алгоритмы обработки данных»**

Тема. Вычислительная сложность алгоритма

|  |  |  |
| --- | --- | --- |
| Выполнил студент группы ИКБО-65-23 |  | Петров А. |
| Принял старший преподаватель |  | Скворцова Л.А. |

Москва 2023

Оглавление

[1. УСЛОВИЕ ЗАДАЧИ. 2](#_Toc159787803)

[2. РАЗРАБОТКА РЕШЕНИЯ 3](#_Toc159787804)

[2.1. Представление алгоритма на псевдокоде и анализ 3](#_Toc159787805)

[функциональных зависимостей 3](#_Toc159787806)

[2.2. Код алгоритма на языке С++. 6](#_Toc159787807)

[2.3. Тестирование алгоритма. 6](#_Toc159787808)

[3. Выводы 7](#_Toc159787809)

[4. Информационные источники 8](#_Toc159787810)

# **УСЛОВИЕ ЗАДАЧИ.**

**Задание**

Разработать алгоритм задачи варианта (табл. 3). Определить функцию, показывающую зависимость количества выполняемых инструкций от размера задачи, функцию емкостной сложности алгоритма.

**Требования к выполнению практической работы**

1. Выполнить разработку алгоритма задачи варианта, представляя последовательность как массив из n значений, записать алгоритм на псевдокоде. Символы и правила для псевдокода приведены в пособии [2], представленном в списке информационных источников.

2. Определить, для полученного алгоритма, функциональную зависимость (функцию), указывающей зависимость количества выполняемых инструкций от размера задачи.

3. Технологию подсчета количества инструкций алгоритма представить в таблице (Таблица 1). При этом: − псевдокод алгоритма разместить в столбце Инструкции (оператор) алгоритма таблицы (Таблица 1), каждая управляющая инструкция строго в отдельной строке таблицы;

4. Определить функцию (функции: наилучший, наихудший и средний случаи) зависимости количества инструкций алгоритма от размера задачи и от данных. Для этого выполнить суммарный подсчет всех значений столбца Количество выполнений инструкции, учитывая влияние данных на количество выполняемых инструкций.

5. Реализовать алгоритм.

6. Разработать тесты для доказательства корректной работы алгоритма

при n=20.

7. Выполнить отладку и тестирование алгоритма.

8. Определить емкостную сложность алгоритма.

**Вариант задачи № 22**

Дано натуральное число n и последовательность натуральных чисел. Определить количество чисел последовательности, являющихся палиндромами.

# **РАЗРАБОТКА РЕШЕНИЯ**

## **2.1. Представление алгоритма на псевдокоде и анализ**

## **функциональных зависимостей**

Составленный алгоритм на псевдокоде, находящийся в таблице (Таблица 1) разложим по количеству выполняемых инструкций операторами. Обозначения имен объектов алгоритма и их назначение:

posled – массив значений

n – размер массива (n>0)

originalNum– переменная для хранения числа из массива

reverseNum – переменная обратная originalNum

temp – вспомогающая переменная для подсчета

palindromeCount – кол-во чисел в массиве, являющихся палидоромами

Таблица 1. Форма представления алгоритма при получении функции зависимости количества выполняемых инструкций от размера задачи

|  |  |  |
| --- | --- | --- |
| Номер инструкции | Код алгоритма | Количество операций в инструкции |
| 1 | count←0 | 1 |
| 2 | n ←0 | 1 |
| 3 | originalNum ←0 | 1 |
| 4 | palindromeCount ←0 | 1 |
| 5 | For i←0 to n do | 1+3(n+1) |
| 6 | originalNum ← posled[i]; | 2n |
| 7 | originalNum ← originalNum / 10; | 3n |
| 8 | while (originalNum > 0) do | n\*(len(originalNum)+1) |
| 9 | temp ← originalNum % 10 | n\*(len(originalNum)\*2) |
| 10 | reverseNum ← reverseNum \* 10 + temp; | n\*(len(originalNum)\*3) |
| 11 | originalNum ← originalNum / 10 | n\*(len(originalNum)\*2) |
| 12 | if (posled[i] == reverseNum) then | n\*(len(originalNum)\*2) |
| 13 | palindromeCount++; | n\*(len(originalNum)) |
| 14 | endIf | - |
| 15 | od | - |
| 16 | od | - |

Пусть кол-во разрядов в числах будет равно len(originalNum) = t, тогда общая теоретическая временная сложность алгоритма:

T(n) = 1+1+1+1+1+3(n+1)+2n+3n+n(t+1)+n\*2t+n\*3t+n\*2t+n\*2t+n\*t

T(n) = 4+3n+3+5n+nt+n+2nt+3nt+2nt+2nt+nt

T(n) = 7+9n+11nt

Асимптотическая временная сложность линейная – O(n).

Лучший случай: в массиве posled все числа состоят из одного разряда (t = 1)

T(n) = 7+9n+11nt

T(n) = 7+20n

Худший случай: в массиве posled все числа состоят из максимального количества разрядов (t=max) и все они палиндромы, выполняется условие (posled[i]==reverseNum)

T(n) = 7+9n+11nt

Средний случай:

T(n) = (7+20n +7+9n+11nt)/2

T(n) = (14+29n+11nt)/2

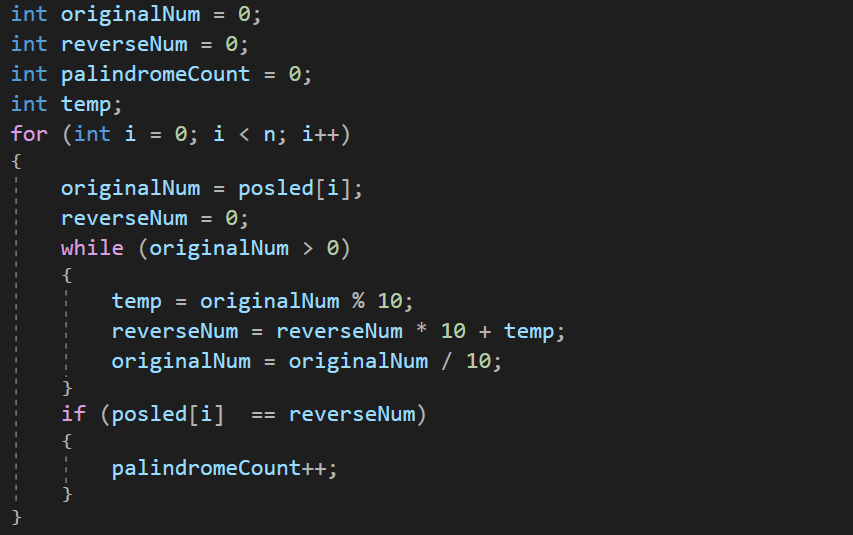
T(n) = 7+15n+6t

Оценим емкостную сложность алгоритма: так как в качестве входных

данных поступает массив размерностью n, дополнительного массива не

требуется, поэтому емкостная сложность алгоритма равна O(n).

## **2.2. Код алгоритма на языке С++.**



## **2.3. Тестирование алгоритма.**

Составленные тесты (Таблица 2), которые обрабатывают базовые и крайние случаи для размерности массива равной 20 (n = 20).

Таблица 2 - Таблица тестирования

|  |  |  |
| --- | --- | --- |
| Номер теста | Входные данные | Эталон результата |
| 1 | posled={63,23,42,23,75,57,34,67,123,43,87,34,87,  12,9,65,22,11,33,44} | palindromeCount = 5 |
| 2 | posled={23,34,78,54,9,87,65,43,21,32,43,65,78,9,  12,22,43,65,87,21} | palindromeCount = 3 |
| 3 | posled={1,2,3,4,5,6,7,8,9,11,1111,111111,111111,  111111,11111,111111,111111,1111,1111,11} | palindromeCount = 20 |
| 4 | posled={12,23,43,54,65,76,78,98,87,76,56,45, 34,23,12,23,34,45,56,67,} | palindromeCount = 0 |









Рисунок 2 - Скриншоты тестирований

Исследования алгоритма при T(n) = 7+9n+11nt и t=4 (кол-ве разрядов в числах) и 3.4\*10^9 Гц процессора на различных объемах данных и получение времени его выполнения занесены в таблицу (Таблица 3).

Таблица 3 - Параметры алгоритма при оценке сложности алгоритма

|  |  |  |  |
| --- | --- | --- | --- |
| Размер задачи(n) | Время выполнения алгоритма (ms) | Количество инструкций по формуле функции (Т) | Время выполнения Т инструкций на компьютере (Т/быстродействие комп.) (ms) |
| 1 | 22 | 60 | 0,000176 |
| 100 | 45 | 5307 | 0,0156 |
| 1000 | 262 | 53007 | 0,156 |
| 5000 | 1244 | 265007 | 0,779 |

# **Выводы**

Сравнивая значения в столбцах 2 и 4 таблицы (Таблица 3), которые указывают время, затраченное на выполнение алгоритм можно заметить явный рост времени при увеличении размера задачи. В процессе разработки алгоритма освоил способ представления в псевдокоде, способы нахождения разрядов чисел в C++, подготовил тестирование простой программы

# 4. **Информационные источники**

1. Приложение к практическим работам – СДО (online-edu.mirea.ru)

2. Структуры и алгоритмы обработки данных – методические указания / Скворцова Л.А., Филатов А.С., Гусев К.В., Трушин С.М. - Москва, МИРЭА - Российский технологический университет, 2023.