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-62-23 Хисамутдинов А.И.

Принял старший преподаватель Скворцова Л.А.

Москва 2024

Оглавление

1. **Условие задачи и задание варианта**

Задание: разработать алгоритм задачи варианта. Определить функцию, показывающую зависимость количества выполняемых инструкций от размера задачи, функцию емкостной сложности алгоритма

Требования к выполнению задания:

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 №27: дано натуральное число n и последовательность натуральных чисел. Определить среднее арифметическое чисел последовательности, сумма цифр которых кратна 7.

2. Разработка решения

2.1. Представление алгоритма на псевдокоде и анализфункциональных зависимостей

Составленный алгоритм на псевдокоде, находящийся в таблице(Таблица 1) разложим по количеству выполняемых инструкций операторами.

Обозначения имен объектов алгоритма и их назначение:arr – массив значений

n – размер массива (n>0)

sum – переменная для хранения суммы элементов массива

count – кол-во чисел в массиве, сумма цифр которых кратна 7

num – переменная для хранения чисел массива

sum_num – переменная для хранения суммы цифр каждого элемента массива

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

Номер строки инструкции алгоритма	Инструкция (оператор) алгоритма	Количество выполнений инструкции
1	count←0	1
2	sum←0	1
3	For i←0 to n-1 do	n+1
4	num←arr[i]	n*2
5	sum_num←0	n
6	While(num > 0) do	n*(len(num))
7	sum_num←sum_num + num % 10	n*3*(len(num)-1)
8	num←num / 10	n*2* (len(num)-1)
9	od	-
10	If (sum_num % 7 = 0) then	n*2
11	sum←sum + arr[i]	n*6
12	count←count + 1	n*4
13	endif	-
14	od	-
15	average←sum / count	2

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

T(n)=2+n+1+n*2+n+n*t+n*3*(t-1)+n*2*(t-1)+n*2+n*6+n*4+2

T(n)=5+n*(6*t+11)

Асимптотическая временная сложность линейная – O(n).

Лучший случай: в массиве arr все числа состоят из одного разряда (t =1) и ни одно число не кратно 7

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

T(n) = 5+n(1+6*t)

Худший случай: в массиве arr все числа состоят из максимального количества разрядов (t=max) и сумма цифр каждого числа в массиве кратна 7

T(n)=5+n*(6*t+11)

Средний случай:

T(n)=(5+n*(6*t+11)+5+n(1+6*t))/2

T(n)=5+6*n(1+t)

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. Алгоритм на языке C++

При реализации алгоритма (Рисунок 1) удалось избежать инициализации счетной переменной цикла for путем использования в нем значения n, которое копируется в память. После нахождения элементов, удовлетворяющих условиям задачи, возвращается их кол-во, при обратном исходе возвращается нулевое значение

```cpp
float arithmetic_mean(int n, int arr[])
{
    float sum = 0, count = 0;
    for (int i = 0; i < n; ++i)
    {
        int num = arr[i];
        int sum_num = 0;
        while (num > 0)
        {
            sum_num += num % 10;
            num /= 10;
        }
        if (sum_num % 7 == 0)
        {
            sum += arr[i];
            count++;
        }
    }
    float average = sum / count;
    return average;
}
```

Рисунок 1 – Алгоритм на языке C++

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

Среднее арифметическое чисел последовательности, сумма цифр которых кратна 7 float arithmetic_mean(int n, int arr[])		
Номер теста	Входные данные	Эталон результата
1	n = 20, arr = { 7, 16, 34, 879, 231, 5342, 6543, 3241, 43, 76, 435, 809, 312, 876, 4234, 54, 65, 786, 98, 76}	average = 1014.86
2	n = 20, arr = {7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7}	average = 7
3	n = 20, arr = { 7, 16, 34, 25, 124, 2221, 331, 133, 4111, 511, 7, 61, 70, 52, 16, 1111111, 43, 2221, 133, 16}	average = 56062.1
4	n = 20, arr = { 1, 2, 3, 4, 5, 6, 8, 9, 12, 23, 45, 67, 87, 90, 132, 465, 798, 987, 765, 5434}	average = nan

Тестирования успешно пройдены (Рисунок 2), программа соответствует своей спецификации.

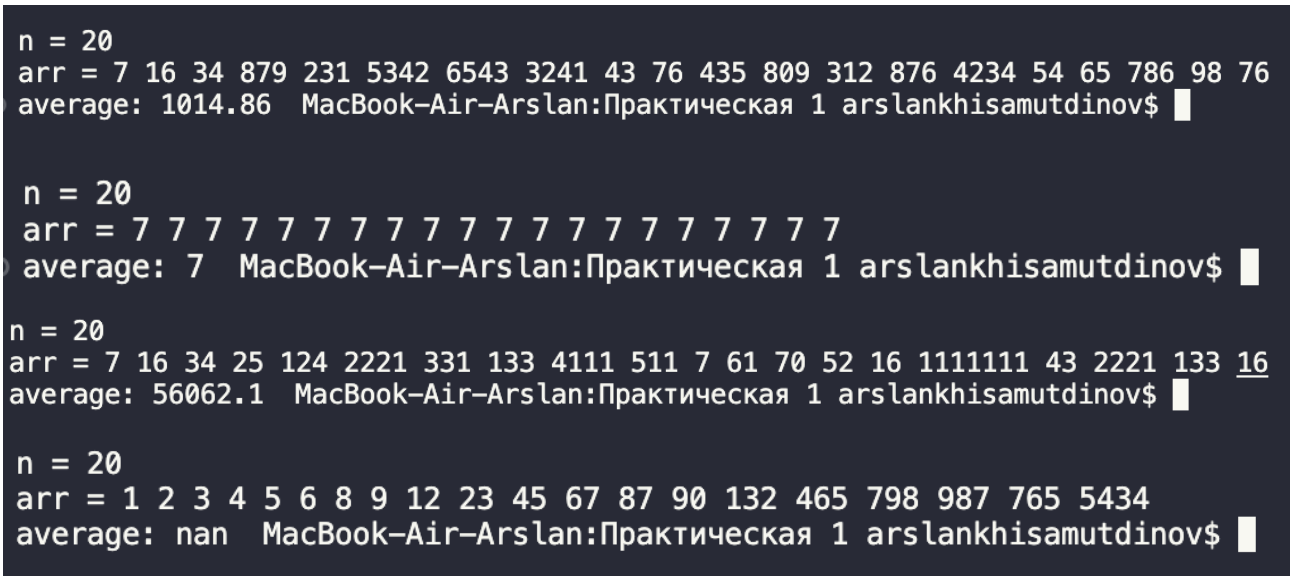

Рисунок 2 - Скриншот тестирований

Исследования алгоритма при T(n)=5+n*(6*t+11) и t=4 (кол-ве разрядовв числах) и 3480 МГц 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 (T)	Время выполнения T инструкций на компьютере (T/быстродействие комп.) (ms)
1	0.0006	40	0.000001
100	0.0069	3505	0.0001
1000	0.0634	35005	0.001
5000	0.2957	175005	0.005

3. 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.