ОТЧЕТ

по лабораторной работе № 3

«Экспериментальное определение количества элементарных операций языка высокого уровня в программной реализации алгоритма»

по дисциплине «Основы теории алгоритмов»

Выполнил студент группы ИС/б-22о

Горбенко К.Н.

Проверил:

Карлусов В.Ю.

* 1. ЦЕЛЬ РАБОТЫ

Экспериментальная проверка теоретически полученной функции трудоемкости для алгоритма точного решения задачи о сумме методом полного перебора. Получение практических навыков расстановки счетчика операций в программе на языке высокого уровня.

* 1. ВАРИАНТ ЗАДАНИЯ

Таблица 1 – Вариант задания

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Размерность вектора случайных чисел | Максимальное случайное число в векторе | Значение суммы (V) |
| 17 | 12 | 90 | 29 |

* 1. ТЕКСТ ПРОГРАММЫ

using System;

public static class SumTask

{

public static bool Compose(int[] array, int V, out int[] counter, out int operations)

{

var N = array.Length;

int maxN = (int)Math.Pow(2, N) – 1;

counter = new int[N];

operations = 0;

for (int cnt = 1; cnt < maxN; cnt++)

{

int sum = 0; operations += 4;

for (int i = 0; i < N; i++)

{

sum = sum + counter[i] \* array[i];

operations += 8;

}

operations += 1;

if (sum == V)

{

operations += 2;

return true;

}

operations += 1;

var j = N - 1; operations += 2;

while (counter[j] == 1 && j != 0)

{

counter[j] = 0;

j = j - 1;

operations += 8;

}

operations += 1;

counter[j] = 1;

cnt = cnt + 1;

operations += 4;

}

return false;

}

}

using System;

using System.Linq;

using System.IO;

class Program

{

static void Main(string[] args)

{

var random = new Random();

var arrayLength = 12;

var maxValue = 90;

var sum = 29;

var path = "file.txt";

for (int i = 1; i <= 10; i++)

{

var array = new int[arrayLength].Select(x => random.Next(maxValue)).ToArray();

using (var streamWriter = new StreamWriter(path, append:true))

{

foreach (var value in array)

streamWriter.Write($"{value} ");

if (SumTask.Compose(array, sum, out var counter, out var operations))

streamWriter.WriteLine("\nSum has been composed");

else streamWriter.WriteLine("\nSum has not been composed");

foreach (var value in counter)

streamWriter.Write($"{value} ");

streamWriter.WriteLine($"\noperations - {operations}\n");

}

}

}

}

* 1. СТРУКТУРНАЯ СХЕМА АЛГОРИТМА

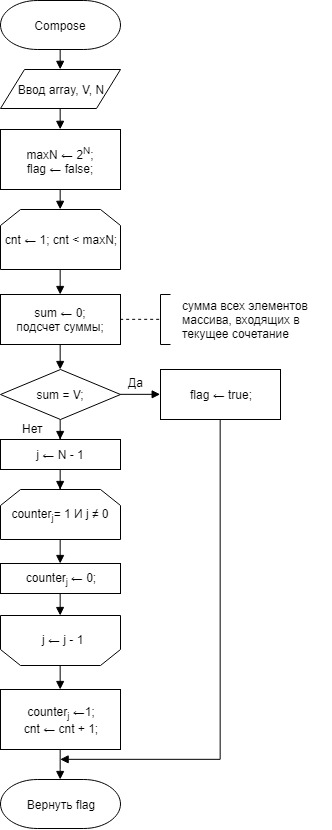


Рис. 1 – Блок схема алгоритма решения задачи о сумме

* 1. ОПИСАНИЕ ГЕНЕРАТОРА СЛУЧАЙНЫХ ЧИСЕЛ

В лабораторной работе используется стандартный генератор случайных чисел языка C# - класс Random. Он инициализируется начальным значением генератора. При одинаковых начальных значениях генератор будет возвращать одинаковые случайные числа.

* 1. ПОЛУЧЕНИЕ ФУНКЦИИ ТРУДОЕМКОСТИ

Построчно подсчитаем количество операций алгоритма:

var N = array.Length; - 2

int maxN = (int)Math.Pow(2, N) - 1; N + 2

counter = new int[N]; 2N

for (int cnt = 1; cnt < maxN; cnt++) 3\*2^N + 1

{

int sum = 0; 1

for (int i = 0; i < N; i++) 3N + 1

{

sum = sum + counter[i] \* array[i]; 5

}

if (sum == V) 0.00025 \* 1 + N

return true;

var j = N - 1; 2

while (counter[j] == 1 && j != 0) 4N + 1

{

counter[j] = 0; 2

j = j – 1; 2

}

counter[j] = 1; 2

cnt = cnt + 1; 2

}

return false;

Получим следующую функцию оценки трудоемкости для худшего случая:

.

Для лучшего случая:

.

* 1. РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

Таблица 2 – Результат работы программы

|  |  |  |
| --- | --- | --- |
| № эксперимента | Последовательность чисел/  включенные в сумму сочетания | Количество операций |
| 1 | 76 2 13 70 8 53 5 46 59 37 22 51 | 123865 |
| 0 1 0 0 0 0 1 0 0 0 1 0 |
| 2 | 50 66 62 24 35 18 82 36 49 64 33 73 | 239411 |
| 0 1 1 1 1 1 1 1 1 1 1 1 |
| 3 | 29 64 6 4 13 18 37 27 49 84 58 58 | 239411 |
| 0 1 1 1 1 1 1 1 1 1 1 1 |
| 4 | 73 49 42 9 62 6 1 18 87 41 85 56 | 239411 |
| 0 1 1 1 1 1 1 1 1 1 1 1 |
| 5 | 66 72 63 42 63 14 76 78 68 40 47 15 | 7692 |
| 0 0 0 0 0 1 0 0 0 0 0 1 |
| 6 | 16 58 76 2 6 35 13 33 25 73 0 68 | 239411 |
| 0 1 1 1 1 1 1 1 1 1 1 1 |
| 7 | 73 35 49 29 53 35 80 59 36 50 36 59 | 30047 |
| 0 0 0 1 0 0 0 0 0 0 0 0 |
| 8 | 88 47 49 62 80 6 50 83 72 85 11 24 | 239411 |
| 0 1 1 1 1 1 1 1 1 1 1 1 |
| 9 | 30 28 69 58 63 66 53 70 20 11 67 21 | 239411 |
| 0 1 1 1 1 1 1 1 1 1 1 1 |
| 10 | 48 89 57 18 32 12 82 10 86 49 66 29 | 212 |
| 0 0 0 0 0 0 0 0 0 0 0 1 |

* 1. ВЫВОД

В ходе лабораторной работы была теоретически и практически оценена сложность алгоритма поиска суммы. В худшем случае оценка сложности принимала следующий вид , в лучшем .

Практическая оценка сложности включала подсчет количества элементарных операций с помощью счетчика, включенного в программу. В худшем случае количество элементарных операций составило 239411, в лучшем – 212.