ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Направление: 09.03.04 – «Программная инженерия»
Профиль: Современная разработка программного обеспечения

Семестровая работа по дисциплине Алгоритмы и Структуры Данных на тему:

«Сортировка и сумма трёх»

Работу выполнила:
Студентка 1 курса 11-305 группы очного отделения контрактной формы обучения Бережная Светлана Сергеевна

Руководитель:
Доцент Цивильский Илья Владимирович

Казань
2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Задачей данной семестровой работы является реализация программного класса SumOfThree, который для трех заданных множеств A, B и C(каждое из которых содержит не более N целых чисел) определяет, существует ли такая тройка (a, b, c), где a принадлежит A, b принадлежит B, и c принадлежит C, что a + b + c = 0. А также оценить и обосновать сложность алгоритма.

РЕАЛИЗАЦИЯ

Для **реализации** поставленной задачи был создан булевый метод CheckSumm в классе SumOfThree (см. рис. 1) , в котором используется метод двух указателей, что позволило уменьшить асимптотическую сложность до O(n*m), где n - размер массива A, m - размер наибольшего из массивов B и C. Такая сложность обуславливается тем, что программа содержит два вложенных цикла, первый из которых содержит n итераций, а второй - m итераций, на каждой из которых мы переопределяем указатель.

В случае решения задачи с помощью трёх вложенных циклов сложность алгоритма составила бы O(n*m*t) , где n - размер массива A, m - размер массива B и t - размер массива C.

```
Checks a sum of three numbers from three arrays

Params: A – is first array
        B – is second array
        C – is third array

Returns: True if the sum do match (false if sum do not match)
```

```java
public static boolean checkSumm(int[] A, int[] B, int[] C) {  4 usages   fewalthel *
    if (A == null || B == null || C == null || A.length == 0 || B.length == 0 || C.length == 0) {
        System.out.println("поиск невозможен, если какой-то из массивов пуст"); return false;
    }
    //сортировка в порядке возрастания массивов B и C
    Arrays.sort(B); Arrays.sort(C);

    //определяем больший по размеру массив из B и C
    int[] third, second;
    if (C.length >= B.length) {third = C; second = B;
    } else {second = C; third = B;}

    boolean result = false;
    for (int j : A) {
        int k = 0; //первый указатель (индекс в меньшем массиве)
        int n = third.length - 1; //второй указатель(индекс в большем массиве)

        while (k <= n) {
            if (j + second[k] + third[n] < 0) { //если сумма < 0, то необходимо рассмотреть число > second[k]
                k++;
            } else if (j + second[k] + third[n] > 0) { //если сумма > 0, то необходимо рассмотреть число < third[n]
                n--;
            } else { //если сумма = 0, прерываем цикл
                result = true;
                f = j; s = second[k]; t = third[n];
                break;
            }
        }
        if (result) { break; }
    } return result;
}
```

Рисунок 1 - Реализация функционала метода проверки суммы

```java
public class SumOfThree {  7 usages   fewalthel *

    public static int f; //первое число в тройке
    public static int s; //второе число в тройке
    public static int t; //третье число в тройке
```

Рисунок 2 - Глобальные переменные для тройки найденных чисел

Для проверки времени выполнения программы использован метод long nanoTime(), который измеряет время в наносекундах. В методе main класса Main был создан цикл из 10 итераций, в каждой из которых создаются массивы рандомных размеров, а затем для каждой тройки замеряется время работы метода boolean checkSumm(int[] A, int[] B, int[] C) класса SumOfThree.

```
public static void main(String[] args) throws IOException {  fewalthel *
    //выбираем числовой диапазон для рандомных чисел в массиве
    Scanner scan = new Scanner(System.in);
    System.out.println("Выберите минимальное число в массиве");
    int minNum = scan.nextInt();
    System.out.println("Выберите максимальное число в массиве");
    int maxNum = scan.nextInt();

    int minSize = 10, maxSize = 100;

    //замеряем время работы программы для 10 рандомных массивов
    for (int i = 0; i < 10; i++) {
        long start = System.nanoTime();

        //создаем и заполняем рандомные массивы
        int[] A = createAndFillArray(minSize, maxSize, minNum, maxNum);
        int[] B = createAndFillArray(minSize, maxSize, minNum, maxNum);
        int[] C = createAndFillArray(minSize, maxSize, minNum, maxNum);

        boolean result = SumOfThree.checkSumm(A, B, C);

        long finish = System.nanoTime();
        long elapsed = finish - start;
        String filename = "./info" + i + ".csv";

        Write.writeArraysToCSV(filename, A, B, C);
        Write.writeNumbersToCSV(filename, SumOfThree.f, SumOfThree.s, SumOfThree.t, result);
        Write.writeTimeToCSV(filename, elapsed);
    }
}
```

Рисунок 3 - логика программы в методе main

Числовой диапазон чисел в рандомных массивах выбирает сам пользователь. Результат выполнения каждой программы в цикле в наносекундах, а также входные данные для каждой итерации записываются в файлы info0.csv - info9.csv (см. рис. 3).

Для фиксирования найденной тройки чисел в трёх множествах созданы глобальные переменные в классе SumOfThree, которые будут переопределяться после каждого нового использования метода checkSumm (см. рис. 1 и рис. 2)

Создание массивов и заполнение их рандомными числами осуществляется с помощью метода createAndFillArray класса Main (см.рис.4).

```
A method to initialize an array of random size.

Params: minSize – is minimum quantity of elements
        maxSize – is maximum quantity of elements

Returns: an array of generated integers
```

```java
public static int[] createAndFillArray(int minSize, int maxSize, int minNum, int maxNum) {
    if (minSize > maxSize || minSize < 0) {
        System.out.println("неподходящие параметры для размера массива"); return null;
    }

    //генерация рандомного размера для массива
    int size = (int) (Math.random() * (maxSize - minSize) + minSize);
    int[] array = new int[size];

    //заполнение массива рандомными числами в указанном диапазоне
    if (minNum > maxNum || minNum < 0) {
        System.out.println("неподходящие параметры для чисел массива"); return null;
    }
    for (int j = 0; j < array.length; j ++) {
        array[j] = (int) (Math.random() * maxNum- minNum) + minNum;
    }
    return array;
}
```

Рисунок 4 - Реализация функционала метода для создания и заполнения массива рандомными числами

Для записи различных данных в csv файл реализован класс Write, содержащий три метода:

-writeTimeToCSV для записи времени работы программы (см. рис. 5)

-writeArraysToCSV для записи входных данных (см. рис. 6).

-writeNumbersToCSV для записи найденной тройки чисел (если тройка не была найдена, метод записывает в файл строку "Тройка чисел не была найдена") (см. рис. 7)

```
A method for writing time data to csv file

Params: fileName – is of the file to which the data is being written
        result – is data is being written

public static void writeTimeToCSV(String fileName, long result) throws IOException {  1 usage
    BufferedWriter writer = new BufferedWriter(new FileWriter(fileName,  append: true));
    try {
        writer.write( str: "\nВремя работы программы: " + result + " нс.");
    } catch (IOException e) {
        System.out.println(e);
    } finally {
        writer.flush();
        writer.close();
    }
}
```

Рисунок 5 - Реализация функционала метода записи времени работы метода checkSumm

```
A method for writing arrays data to csv file

Params: fileName – is of the file to which the data is being written
        array1 – is first array from where the data is read
        array2 – is second array from where the data is read
        array3 – is third array from where the data is read

public static void writeArraysToCSV(String fileName, int[] array1, int[] array2, int[] array3) throws IOException {
    BufferedWriter writer = new BufferedWriter(new FileWriter(fileName,  append: true));
    try {
        writer.write( str: "Для массивов: \n[");
        // Записываем элементы первого массива
        for (int num : array1) { writer.write( str: num + " ");}
        writer.write( str: "]\n[");

        // Записываем элементы второго массива
        for (int num : array2) { writer.write( str: num + " ");}
        writer.write( str: "]\n[");

        // Записываем элементы третьего массива
        for (int num : array3) { writer.write( str: num + " ");}
        writer.write( str: "]");

    } catch (IOException e) {
        System.out.println(e);
    } finally {
        writer.flush();
        writer.close();
    }
}
```

Рисунок 6 - Реализация функционала метода записи входных данных для метода checkSumm

```
A method for writing time data to csv file

Params: fileName – is of the file to which the data is being written
        f – is first number is being written
        s – is second number is being written
        t – is third number is being written
        result – is result of the method checkSumm execution
```

```java
public static void writeNumbersToCSV(String fileName, int f, int s, int t, boolean result) throws IOException {
    BufferedWriter writer = new BufferedWriter(new FileWriter(fileName,  append: true));
    try {
        if (result) {
            writer.write( str: "\nДля чисел: " + f + " " + s + " " + t);
        } else {
            writer.write( str: "\nТройка чисел не была найдена");
        }
    } catch (IOException e) {
        System.out.println(e);
    } finally {
        writer.flush();
        writer.close();
    }
}
```

Рисунок 7 - Реализация функционала метода найденной тройки в методе checkSumm

РЕЗУЛЬТАТЫ

Результатом работы программы является булево значение в зависимости от входных данных. Для демонстрации результата работы программы при разных входных данных использованы unit тесты в классе Test (см.рис.8):

-в случае, если какой-то из входных массивов пуст, тест возвращает false

-в случае, если все массивы не пусты и существует тройка чисел из трёх множеств, дающая в сумме 0, тест возвращает true.

-в случае, если все массивы не пусты и не существует такой тройка чисел из трёх множеств, дающей в сумме 0, тест возвращает false.

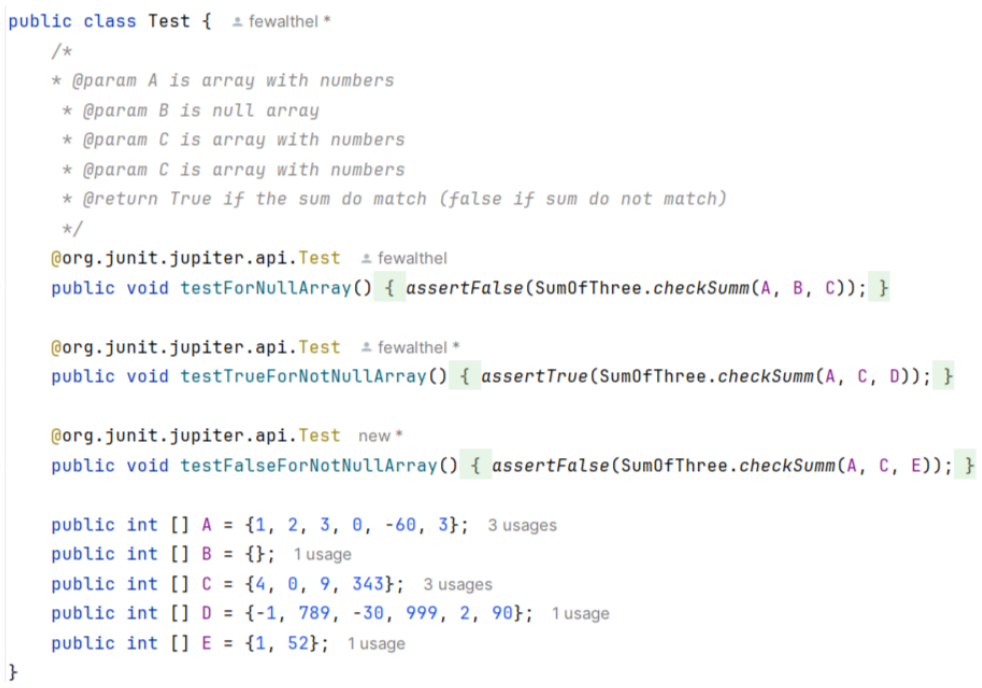

Рисунок 8 - Реализация unit тестов для разных входных данных

Время работы программы при рандомных входных данных можно просмотреть в файлах info0.csv - info9.csv, находящихся в папке проекта. В этих файлах указаны входные данные, результат работы программы, а также время работы программы в наносекундах.

ПРИЛОЖЕНИЯ

Приложение 1 - листинг класса Main

```
/**
* File: Main.java
* Description: General class in project
* Author: Berezhnaya Svetlana
* Date: 2.05.2024
*/

package org.example;
import java.util.Scanner;
import java.io.IOException;
import java.lang.Math;
```

```
public class Main {
    public static void main(String[] args) throws IOException {
        //выбираем числовой диапазон для рандомных чисел в массиве
        Scanner scan = new Scanner(System.in);
        System.out.println("Выберите минимальное число в массиве");
        int minNum = scan.nextInt();
        System.out.println("Выберите максимальное число в массиве");
        int maxNum = scan.nextInt();

        int minSize = 10, maxSize = 100;

        //замеряем время работы программы для 10 рандомных массивов
        for (int i = 0; i < 10; i++) {
            long start = System.nanoTime();

            //создаем и заполняем рандомные массивы
            int[] A = createAndFillArray(minSize, maxSize, minNum, maxNum);
            int[] B = createAndFillArray(minSize, maxSize, minNum, maxNum);
            int[] C = createAndFillArray(minSize, maxSize, minNum, maxNum);

            boolean result = SumOfThree.checkSumm(A, B, C);

            long finish = System.nanoTime();
            long elapsed = finish - start;
            String filename = "./info" + i + ".csv";

            Write.writeArraysToCSV(filename, A, B, C);
            Write.writeNumbersToCSV(filename, SumOfThree.f, SumOfThree.s,
SumOfThree.t, result);
            Write.writeTimeToCSV(filename, elapsed);
        }
```

```
    }


    /**
     * A method to initialize an array of random size.
     * @param minSize is minimum quantity of elements
     * @param maxSize is maximum quantity of elements
     * @return an array of generated integers
     */
    public static int[] createAndFillArray(int minSize, int maxSize, int minNum,
int maxNum) {
        if (minSize > maxSize || minSize < 0) {
            System.out.println("неподходящие параметры для размера массива");
return null;
        }


        //генерация рандомного размера для массива
        int size = (int) (Math.random() * (maxSize - minSize) + minSize);
        int[] array = new int[size];


        //заполнение массива рандомными числами в указанном диапазоне
        if (minNum > maxNum ) {
            System.out.println("неподходящие параметры для чисел массива"); return
null;
        }
        for (int j = 0; j < array.length; j ++) {
            array[j] = (int) (Math.random() * maxNum- minNum) + minNum;
        }
        return array;
    }
}
```

Приложение 2 - листинг класса Write

```java
/**
* File: Write.java
* Description: Class contains a methods for writing different data to files
* Author: Berezhnaya Svetlana
* Date: 2.05.2024
*/

package org.example;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

public class Write {

   /**
    * A method for writing arrays data to csv file
    * @param fileName is of the file to which the data is being written
    * @param array1 is first array from where the data is read
    * @param array2 is second array from where the data is read
    * @param array3 is third array from where the data is read
    */
   public static void writeArraysToCSV(String fileName, int[] array1, int[]
array2, int[] array3) throws IOException {
       BufferedWriter writer = new BufferedWriter(new FileWriter(fileName,
true));
       try {
           writer.write("Для массивов: \n[");
           // Записываем элементы первого массива
           for (int num : array1) { writer.write(num + " ");}
           writer.write("]\n[");
```

```
            // Записываем элементы второго массива
            for (int num : array2) { writer.write(num + " ");}
            writer.write("]\n[");

            // Записываем элементы третьего массива
            for (int num : array3) { writer.write(num + " ");}
            writer.write("]");

        } catch (IOException e) {
            System.out.println(e);
        } finally {
            writer.flush();
            writer.close();
        }
    }

    /**
     * A method for writing time data to csv file
     * @param fileName is of the file to which the data is being written
     * @param result is data is being written
     */
    public static void writeTimeToCSV(String fileName, long result) throws
IOException {
        BufferedWriter writer = new BufferedWriter(new FileWriter(fileName,
true));
        try {
            writer.write("\nВремя работы программы: " + result + " нс.");
        } catch (IOException e) {
            System.out.println(e);
        } finally {
            writer.flush();
            writer.close();
```

```
        }
    }


    /**
     * A method for writing time data to csv file
     * @param fileName is of the file to which the data is being written
     * @param f is first number is being written
     * @param s is second number is being written
     * @param t is third number is being written
     * @param result is result of the method checkSumm execution
     */
    public static void writeNumbersToCSV(String fileName, int f, int s, int t,
boolean result) throws IOException {
        BufferedWriter writer = new BufferedWriter(new FileWriter(fileName,
true));
        try {
            if (result) {
                writer.write("\nДля чисел: " + f + " " + s + " " + t);
            } else {
                writer.write("\nТройка чисел не была найдена");
            }
        } catch (IOException e) {
            System.out.println(e);
        } finally {
            writer.flush();
            writer.close();
        }
    }
}
```

Приложение 3 - листинг класса Test

```
/**
* File: Test.java
```

```
* Description: Class contains a tests for method checkSumm
* Author: Berezhnaya Svetlana
* Date: 2.05.2024
*/

import org.example.*;
import static org.junit.jupiter.api.Assertions.*;

public class Test {
    /*
    * @param A is array with numbers
     * @param B is null array
     * @param C is array with numbers
     * @param C is array with numbers
     * @return True if the sum do match (false if sum do not match)
     */
    @org.junit.jupiter.api.Test
    public void testForNullArray() {
        assertFalse(SumOfThree.checkSumm(A, B, C));
    }

    @org.junit.jupiter.api.Test
    public void testTrueForNotNullArray() {
        assertTrue(SumOfThree.checkSumm(A, C, D));
    }

    @org.junit.jupiter.api.Test
    public void testFalseForNotNullArray() {
        assertFalse(SumOfThree.checkSumm(A, C, E));
    }
```

```
    public int [] A = {1, 2, 3, 0, -60, 3};
    public int [] B = {};
    public int [] C = {4, 0, 9, 343};
    public int [] D = {-1, 789, -30, 999, 2, 90};
    public int [] E = {1, 52};
}
```

Приложение 4 - листинг класса SumOfThree

```
/**
* File: SumOfThree.java
* Description: Class contains a method that checks for the presence of numbers
* in three arrays such that their sum is zero
* Author: Berezhnaya Svetlana
* Date: 2.05.2024
*/
package org.example;
import java.util.Arrays;

public class SumOfThree {
    public static int f; //первое число в тройке
    public static int s; //второе число в тройке
    public static int t; //третье число в тройке

    /**
     * Checks a sum of three numbers from three arrays
     * @param A is first array
     * @param B is second array
     * @param C is third array
     * @return True if the sum do match (false if sum do not match)
     */
    public static boolean checkSumm(int[] A, int[] B, int[] C) {
```

```
        if (A == null || B == null || C == null || A.length == 0 || B.length == 0
|| C.length == 0) {

            System.out.println("поиск невозможен, если какой-то из массивов
пуст"); return false;

        }

        //сортировка в порядке возрастания массивов B и C

        Arrays.sort(B); Arrays.sort(C);

        //определяем больший по размеру массив из B и C

        int[] third, second;

        if (C.length >= B.length) {third = C; second = B;

        } else {second = C; third = B;}


        boolean result = false;

        for (int j : A) {

            int k = 0; //первый указатель (индекс в меньшем массиве)

            int n = third.length - 1; //второй указатель(индекс в большем
массиве)

            while (k <= n) {

                if (j + second[k] + third[n] < 0) { //если сумма < 0, то
необходимо рассмотреть число > second[k]

                    k++;

                } else if (j + second[k] + third[n] > 0) { //если сумма > 0, то
необходимо рассмотреть число < third[n]

                    n--;

                } else { //если сумма = 0, прерываем цикл

                    result = true;

                    f = j; s = second[k]; t = third[n];

                    break;

                }

            }

            if (result) { break; }

        } return result;

    }

}
```