Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

Институт ИВТ

Кафедра «Программное обеспечение»

Отчет

по лабораторной работе №1 по теме

«Структурное тестирование»

по дисциплине

«Тестирование программного обеспечения»

Выполнил:

ст-т гр. Б08-191-2 Хасанов М.П.

Принял преподаватель: Старыгина Е.В.

Ижевск 2019

СОДЕРЖАНИЕ

1. Содержание задания3

2. Блок-схема, описание алгоритма4

3. Тестирование базового пути7

3. 1. Потоковый граф7

3. 2. Цикломатическая сложность8

3. 3. Базовое множество независимых линейных путей9

3. 4. Тестовые варианты9

4. Тестирование потоков данных11

4. 1. Информационный граф11

4. 2. Формирование полного набора DU-цепочек12

4. 3. Формирование полного набора отрезков путей в управляющем графе13

4. 4. Построение маршрутов14

5. Области эквивалентности15

6. Контрольный пример16

7. Текст программы18

1. СОДЕРЖАНИЕ ЗАДАНИЯ

Разработать нижеприведенную программу и провести ее структурное тестирование.

Вариант 20: Даны целые числа а0,...,а15. Получить новый массив по правилу (а0+а8, а1+а9, ..., а7+а15). Найти минимальный элемент полученного массива.

2. БЛОК-СХЕМА, ОПИСАНИЕ АЛГОРИТМА

Представим алгоритм в виде блок-схемы (рис.1).



Рис.1



Продолжение Рис.1

Далее опишем алгоритм работы программы без учета валидации входных данных:

Программа

1 начало программы, получить массив аргументов;

2 выполнить для каждого элемента в массиве

3 преобразовать элемент в число;

2 конец выполнять;

4 обозначим за len размер массива, i – текущий индекc = 0, minimum – минимальный элемент = Infinity;

5 выполнять пока i+8<len

6 обозначим за sum сумму i и i+8 элемент массива, если sum меньше minimum

7 то положить в minimum значение sum

6 конец если;

5 конец выполнять;

8 вывести minimum, конец программы;

3. ТЕСТИРОВАНИЕ БАЗОВОГО ПУТИ

3. 1. Потоковый граф

|Построим по описанию программы потоковый граф, представленный на рис. 2. Видим, что этот потоковый граф имеет четыре региона.



Рис.2

3. 2. Цикломатическая сложность

Определим цикломатическую сложность потокового графа по каждой из трех формул:

1) V(G) = 4 региона;

2) V(G) = 10 дуг - 8 узла + 2 = 4;

3) V(G) = 3 предикатных узлов + 1 = 4.

3. 3. Базовое множество независимых линейных путей

Базовое множество независимых линейных путей:

Путь 1: 1-2-4-5-8; /ввести пустой массив элементов

Путь 2: 1-2-3-2-4-5-8;/ввести массив меньше 9 элементов

Путь 3: 1…5-6-5…5-8; /ввести массив значений Infinity

Путь 4: 1…5-6-7-5…5-8; /режим нормальной обработки

3. 4. Тестовые варианты

Тестовый вариант для пути 1 ТВ1:

ИД: отсутствие аргументов

ОЖ.РЕЗ.: вывод сообщения об ошибке

Тестовый вариант для пути 2 ТВ2:

ИД: аргументы: 1 2 3 4 5 6 7 8

ОЖ.РЕЗ.: вывод сообщения об ошибке

Тестовый вариант для пути 3 ТВ3:

ИД: аргументы = Infinity Infinity Infinity Infinity Infinity Infinity Infinity Infinity Infinity Infinity Infinity

ОЖ.РЕЗ.: Infinity

Тестовый вариант для пути 4 ТВ4:

ИД: аргументы = 1 2 3 4 5 6 7 8 9 10 11 12 13

ОЖ.РЕЗ.: 10

4. ТЕСТИРОВАНИЕ ПОТОКОВ ДАННЫХ

4. 1. Информационный граф

Построим граф программы с управляющими и информационными связями



Рис.3

4. 2. Формирование полного набора DU-цепочек

В нашем примере существуют следующие DU-цепочки:

[numbers, 2,6], [minimum, 5,7], [minimum, 5,8], [sum, 6, 7]

4. 3. Формирование полного набора отрезков путей в управляющем графе

Фрагмент отрезка для первого пути



Рис.4

Фрагмент отрезка для второго пути



Рис.5

Фрагмент отрезка для третьего пути



Рис.6

Фрагмент отрезка для четвертого пути



Рис.7

4. 4. Построение маршрутов

Можно построить маршрут, который полностью покрывает набор отрезков путей управляющего графа: 1-2-3-4-5-6-7-5…5-8 (Ввести 9 чисел).

5. ОБЛАСТИ ЭКВИВАЛЕНТНОСТИ

Для разработанной программы выделим две области эквивалентности: когда смогли определить минимум, и когда его нет. Также для входных данных обозначим еще две области: когда ввели менее 9 чисел и от 9 и более.

Область эквивалентности количества введенных значений

|  |  |
| --- | --- |
| Меньше 9 | Больше или равно 9 |

Рис.8

Область эквивалентности значений входных данных

|  |  |
| --- | --- |
| Целочисленные | Дробные |

Рис.9

Таблица 1

Области эквивалентности

|  |  |  |  |
| --- | --- | --- | --- |
| **Последовательность** | | **Ключевые элементы** | |
| 8 элементов | | Нет в последовательности | |
| 9 элементов | | Первый элемент последовательности | |
| Больше 9 элементов | | Первый элемент последовательности | |
| Больше 9 элементов | | Последний элемент последовательности | |
| Больше 9 элементов | | Средний элемент последовательности | |
|  | | | |
| **Входная последовательность** | **Минимальная сумма** | | **Выходные данные** |
| 0 1 2 3 4 5 6 7 | нет | | Сообщение об ошибке |
| 0 1 2 3 4 5 6 7 8 | 0 + 8 = 8 | | Сумма найдена |
| 1.5 7 13 2 4 5 10 6 7 8 9 | 1.5 + 8 = 9.5 | | Сумма найдена |
| 9 1 4 3 0 5 6 7 12 9 2 | 4 + 2 = 6 | | Сумма найдена |
| 8 7 1 10 0 6 3 5 6 2 8 7 1 5 0 -6 3 5 6 2 | 5 + (-6) = -1 | | Сумма найдена |

6. КОНТРОЛЬНЫЙ ПРИМЕР

Первый тестовый вариант

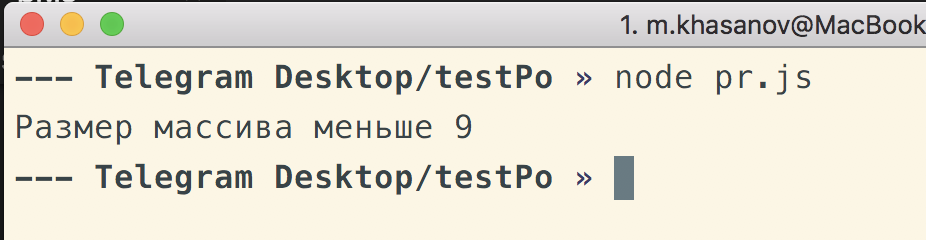


Рис.10

Второй тестовый вариант

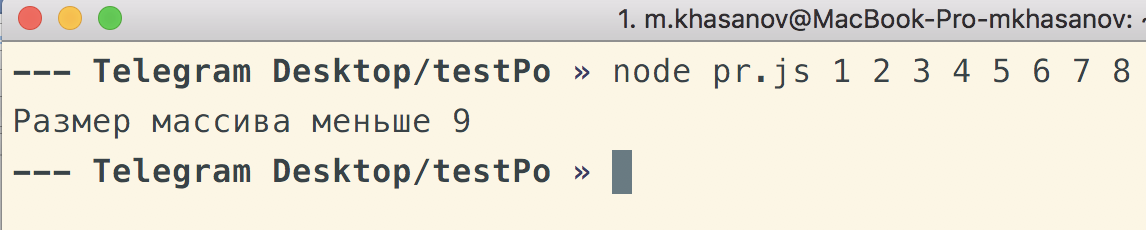


Рис.11

Третий тестовый вариант

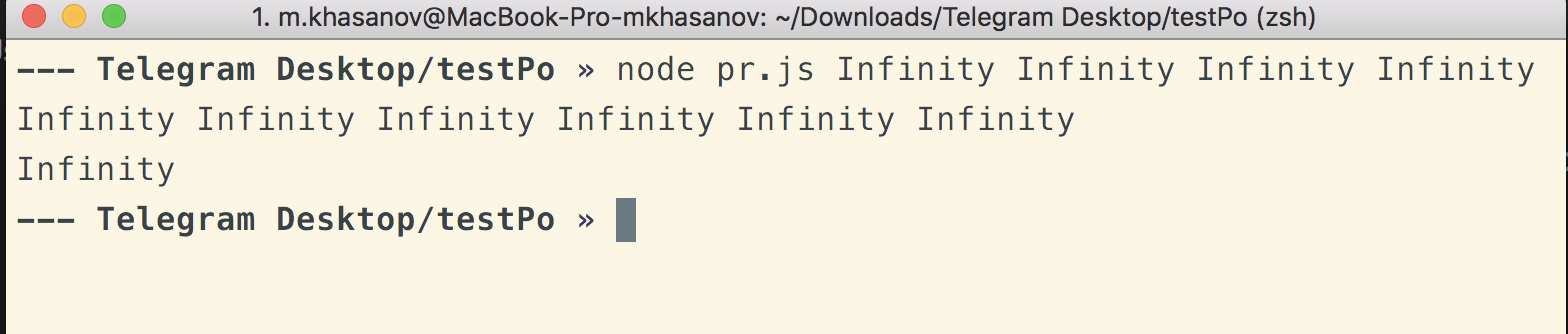


Рис.11

Четвертый тестовый вариант

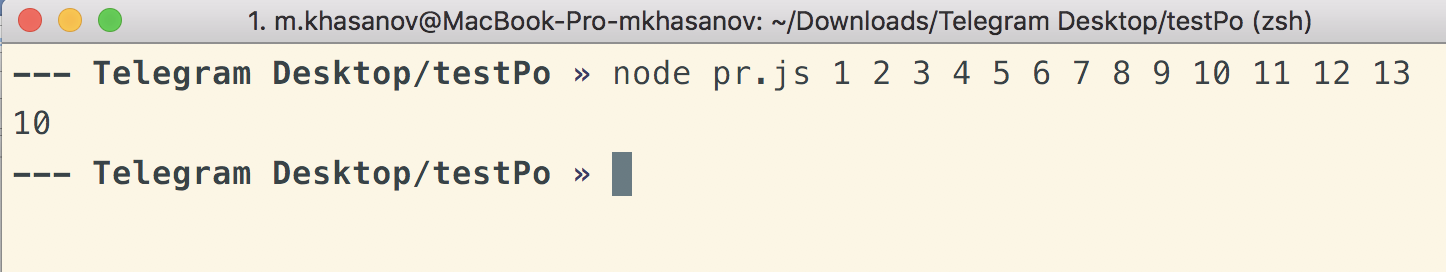


Рис.12

Тестовый вариант с отрицательными значениями

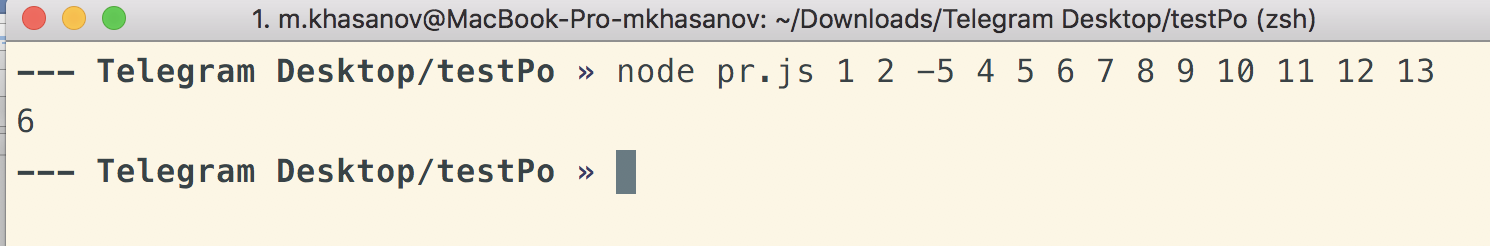


Рис.13

Тестовый вариант с элементами в середине

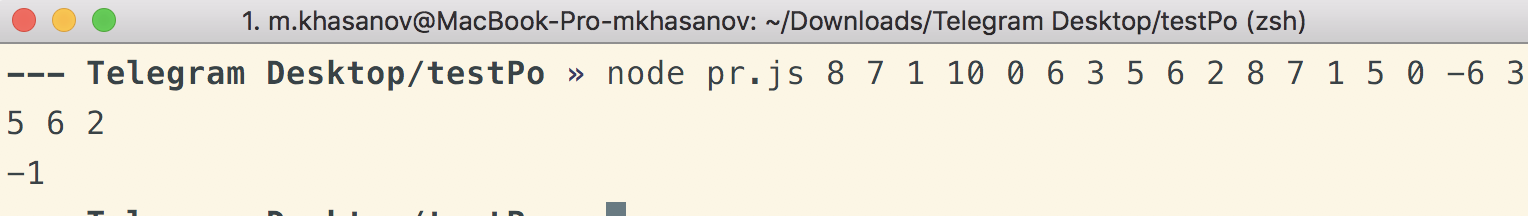


Рис.14

Тестовый вариант с дробными числами

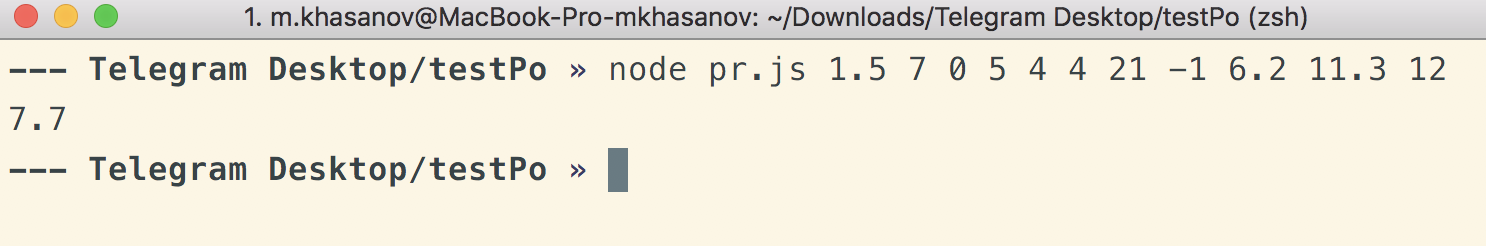


Рис.14

7. ТЕКСТ ПРОГРАММЫ

const SHIFT = 8;

function program(shift = SHIFT) {

const numbers = process.argv.slice(2).map(i => parseFloat(i));

if (numbers.length < shift + 1) {

console.error("Размер массива меньше 9");

return;

}

if (numbers.includes(NaN)) {

console.error("Введены невалидные значения");

return;

}

let minimum = Infinity;

for (let i = 0; i + shift < numbers.length; i++) {

minimum = Math.min(numbers[i] + numbers[shift + i], minimum);

}

console.log(minimum);

}

program();