**Черкасов А. А-06-19 Вариант 24**

**Задание:**

Для заданного одномерного массива Y из N элементов проверить, что существует по крайней мере один элемент *Y*i, для которого выполняется условие . В рекурсивной функции каждый раз отделять первую треть от остальных двух третей рассматриваемой части массива и применять эту же функцию к одной или обеим частям при необходимости. Рекурсивные вызовы заканчивать, когда останется только один элемент в рассматриваемой части массива.

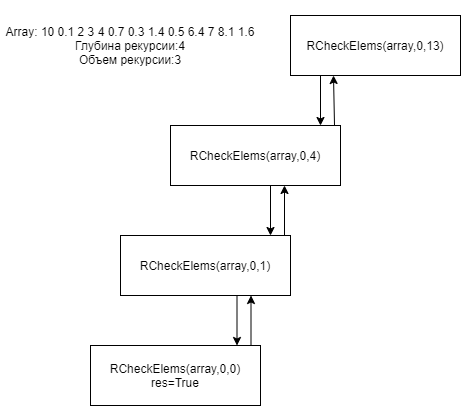
Например, для N=9: , (∨ – «или»)

**Функциональные тесты**

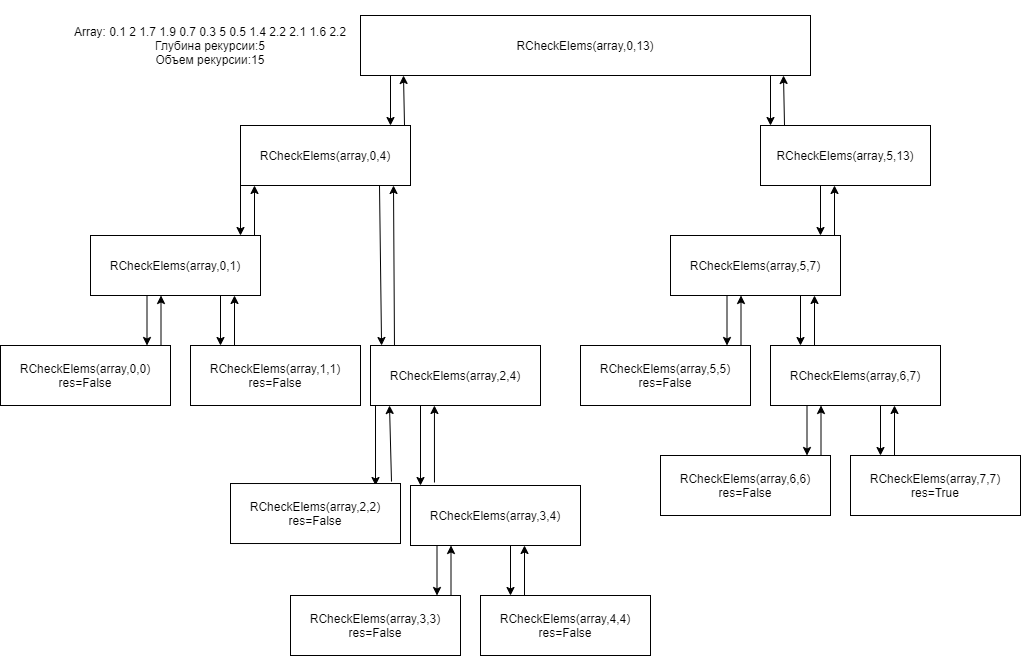
|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Исходные данные** | **Ожидаемый результат** | **Смысл теста** |
| **1** | 5 | Найден элемент (#1), удовлетворяющий условию поиска | Проверка поведения программы, только при одном элементе |
| **2** | 5 0.1 | Найден элемент (#1), удовлетворяющий условию поиска | Проверка разделения массива при двух элементах (левая ветка) |
| **3** | 0.1 5 | Найден элемент (#2), удовлетворяющий условию поиска | Проверка разделения массива при двух элементах (правая ветка) |
| **4** | 3 0.1 0.5 | Найден элемент (#1), удовлетворяющий условию поиска | Проверка разделения массива при трех элементах (первый элемент) |
| **5** | 0.1 3 0.5 | Найден элемент (#2), удовлетворяющий условию поиска | Проверка разделения массива при трех элементах (элемент по середине) |
| **6** | 0.1 0.5 3 | Найден элемент (#3), удовлетворяющий условию поиска | Проверка разделения массива при трех элементах (последний элемент) |
| **7** | 0.1 1.1 1.3 5 1.7 1.9 | Найден элемент (#4), удовлетворяющий условию поиска | Среднее количество элементов, четное, элемент по середине массива |
| **8** | 0.1 4 1.3 5 1.7 1.9 | Найден элемент (#2), удовлетворяющий условию поиска | Среднее количество элементов, два возможных элемента, найден только первый возможный (#2) |
| **9** | 0.1 2 1.7 1.9 0.7 0.3 1.7 0.5 1.9 1.7 2.1 2.3 1.8 | Не найдено элементов, удовлетворяющих условию | Полная проверка массива – отсутствие нужного элемента |
| **10** | 0.1 2 1.7 1.9 0.7 0.3 5 0.5 1.4 2.2 2.1 1.6 2.2 | Найден элемент (#7), удовлетворяющий условию поиска | Среднее количество элементов, нечетное, элемент по середине массива |
| **11** | 0.1 2 1.7 1.9 0.7 0.3 2.9 0.5 3.7 1.15 4.1 1.6 14 | Найден элемент (#13), удовлетворяющий условию поиска | Среднее количество элементов, нечетное, элемент в конце массива – полная проверка массива |
| **12** | 1 2 | Не найдено элементов, удовлетворяющих условию |  |
| **13** |  | Пустой массив | Проверка поведения программы, при пустом массиве |

**Деревья вызовов для N=13:**

**Минимально возможное:**

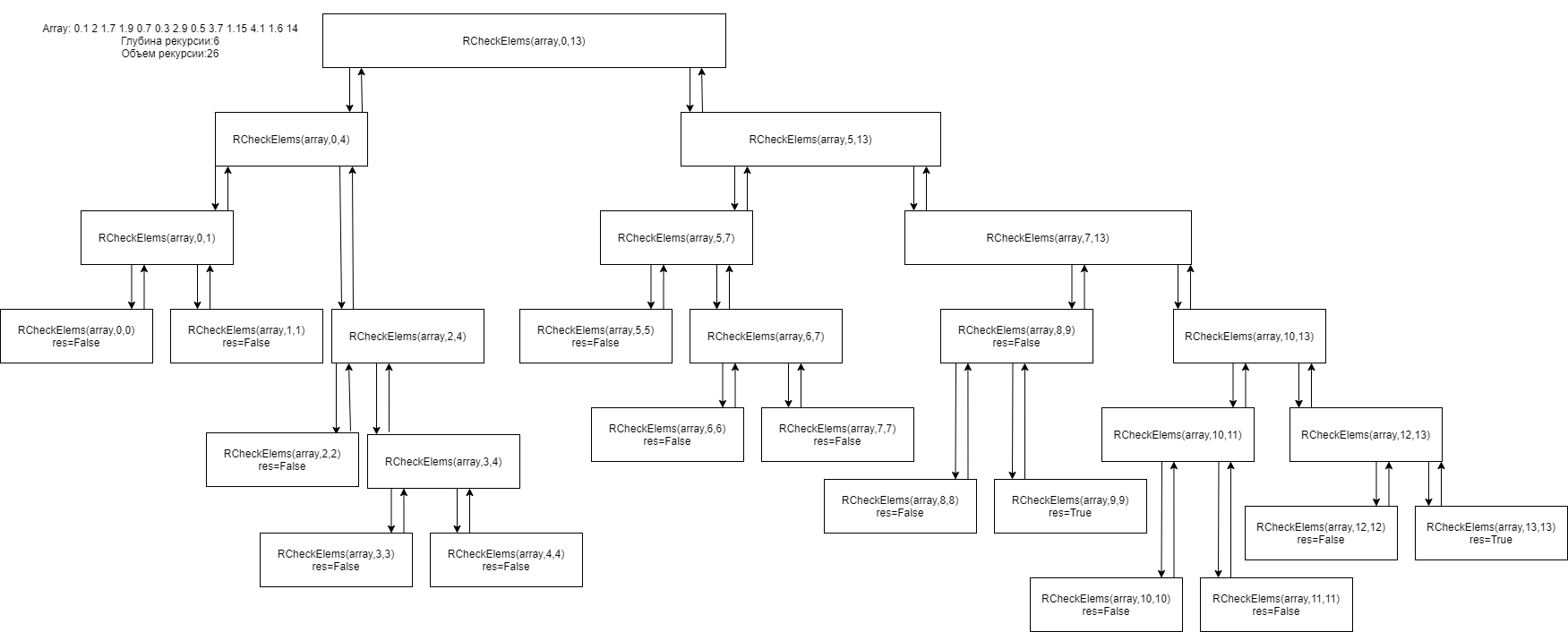
****

**Среднее**

****

**Максимальное**

Array: 0.1 2 1.7 1.9 0.7 0.3 2.9 0.5 3.7 1.15 4.1 1.6 14  
Глубина рекурсии:6  
Объем рекурсии:26

****

**Код:**

**CMakeLists:**

cmake\_minimum\_required(VERSION 3.15)

project(Lab4)

set(CMAKE\_CXX\_STANDARD 17)

add\_executable(Lab4 main.cpp)

**main.cpp**

#include <cstdio>

#include <cmath>

#include <cstdlib>

**bool RCheckElems**(double \*array, int start, int end) //Рекурсия (передаем массив, начало и конец)

{

bool res; //Наш флаг на условие

if (start==end) { //Достигли одного элемента

res = pow((int)array[start], 2) / (end + 1) > 2; //Проверка условия

if (res) {

printf("Найден элемент (#%d), удовлетворяющий условию поиска\n",start+1);

}else{}

}

else

{

int oot = trunc((end-start)/3.0)+start; //отделяем треть

//Проверка отделенных частей (start - oot) первая треть (oot+1 - end) оставшиеся 2 трети

res = RCheckElems(array, start, oot) or RCheckElems(array, oot+1, end);

}

return res;

}

int main() {

int N;

system("chcp 65001");

printf("Лабораторная работа #4\n");

printf("Введите количество элементов массива N=");

scanf("%d",&N);

fflush(stdin);

double \*array = new double[N];

for (int i=0;i<N;i++) scanf("%lf",&array[i]);

fflush(stdin);

if (N!=0) if (!RCheckElems(array, 0, N)) printf("Не найдено элементов, удовлетворяющих условию\n"); else{}

else printf("Пустой массив\n");

delete[]array;

printf("Press ENTER");

getc(stdin);

return 0;

}