|  |  |
| --- | --- |
| Министерство науки и высшего образования Российской Федерации  Санкт-Петербургский политехнический университет Петра Великого  Институт компьютерных наук и технологий  Высшая школа киберфизических систем и управления | |
| ОТЧЁТ  по дисциплине «Теория и технология программирования»  по лабораторной работе №2(7) | |
| **Выполнил**: |  |
| студент гр. 3530902/10003 | Сурьянинова В. В. |
|  |  |
|  |  |
| **Проверил:**  ассистент |  |
|  | Журавская А. |
|  |  |
|  | |

1. Задание

Составить программу для поиска в массиве данных последовательным и бинарным (по возможности) методами. В случае невозможности (из условий индивидуального задания) поиска бинарным методом, провести поиск бинарным методом одной буквы (цифры). Придумать тестовые примеры, для которых были бы эффективными каждый из методов.

Вариант №23: Дана матрица чисел, заданная случайным образом. Найти все простые чиста (до 50).

1. Текст программы

#include <iostream>

#include <ctime>

#include <math.h>

#include <string>

**using** **namespace** std;

**void** swap(**int**& a, **int**& b){

**int** t = a;

a = b;

b = t;

}

**int** enter(string& k, **char** j){

**bool** f = **true**;

**int** count, t;

**while** (f == **true**) {

cout << "Enter the " << j << ": ";

k = "";

cin >> k;

count = 0;

**for**(**int** i = 0; i < k.length(); i++){

**if** ((k[i] >= '0')&&(k[i] <= '9')){

count++;

}

}

**if** (count == k.length()){

f = **false**;

}

}

t = 0;

**for**(**int** i = 0; i < k.length(); i++){

t = 10 \* t + k[i] - 48;

}

**return** t;

}

**void** shell(**int** a[], **int** n){

**int** l, j;

**for**(**int** t = n/2; t > 0; t = t / 2){

**for**(**int** i = t; i < n; i++){

l = a[i];

**for**(j = i; j >= t; j -= t){

**if** (l < a[j - t]){

a[j] = a[j - t];

}

**else** **break**;

}

a[j] = l;

}

}

}

**int** main() {

srand(time(0));

**int** n, m;

string q;

n = enter(q, 'N');

m = enter(q, 'M');

**int** a[n][m], b[n][m], ro[m], res[n\*m];

**int** s[15] = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47};

**for**(**int** i = 0; i < n; i++){

**for**(**int** j = 0; j < m; j++){

a[i][j] = rand() % 50;

}

}

cout << "Original array:\n";

**for**(**int** i = 0; i < n; i++){

**for**(**int** j = 0; j < m; j++){

printf(" %3i ", a[i][j]);

}

cout << "\n";

}

**for**(**int** i = 0; i < n; i++){

**for**(**int** j = 0; j < m; j++){

b[i][j] = a[i][j];

}

}

//сортировка массива

**int** k = 0;

**for** (**int** i = 0; i < n; i ++){

**for** (k = 0; k < m; k ++){

ro[k] = a[i][k];

}

shell(ro, k);

**for** (k = 0; k < m; k ++){

a[i][k] = ro[k];

ro[k] = 0;

}

}

/\*cout << "Отсортированный массив:\n";

for(int i = 0; i < n; i++){

for(int j = 0; j < m; j++){

printf(" %3i ", a[i][j]);

}

cout << "\n";

}\*/

cout << "Binary search algorithm\n";

**bool** flag = **false**;

**int** l = 0, r = m, key = 0, mid = 0, count = 0;

**for** (**int** k = 0; k < 15; k ++){

key = s[k];

**for** (**int** i = 0; i < n; i ++){

**for** (**int** j = 0; j < m; j ++){

ro[j] = a[i][j];

}

l = 0, r = m;

flag = **false**;

**while** ((l <= r) && (flag != **true**)) {

mid = (l + r) / 2; // считываем срединный индекс отрезка [l,r]

**if** (ro[mid] == key){

res[count] = key;

count++;

flag = **true**;//проверяем ключ со серединным элементом

}

**if** (ro[mid] > key) r = mid - 1; // проверяем, какую часть нужно отбросить

**else** l = mid + 1;

}

}

}

cout << "Simple nums: ";

**if** (count == 0){

cout << "no simple nums.\n";

cout << "\n";

cout << "Consistent algorithm\n";

cout << "Simple nums: no simple nums.\n";

}

**else** cout << "\n";

**for** (**int** i = 0; i < count; i ++){

cout << res[i] << " ";

}

cout << "\n";

**if** (count != 0){

cout << "Consistent algorithm\n";

cout << "Simple nums: \n";

**for** (**int** i = 0; i < n; i ++){

**for** (**int** j = 0; j < m; j ++){

**for** (**int** t = 0; t < 15; t ++){

**if** (a[i][j] == s[t]){

cout << a[i][j] << " ";

}

}

}

}

cout << "\n";

}

}

1. Входные и выходные данные

Входные данные: две целочисленные переменные − размеры матрицы.

Выходные данные:

1. Исходная матрица, заполненная случайными числами,
2. Все простые числа до 50, находящиеся в матрице, найденные бинарным поиском,
3. Все простые числа до 50, находящиеся в матрице, найденные последовательно.
4. Пример работы программы

На рисунке 3 представлен пример выполненной программы, где введены нормальные значения размеров матрицы. На рисунке 4 представлен примеры моей выполненной программы, когда вводятся не числа, а неправильные значения, к примеру буквы.

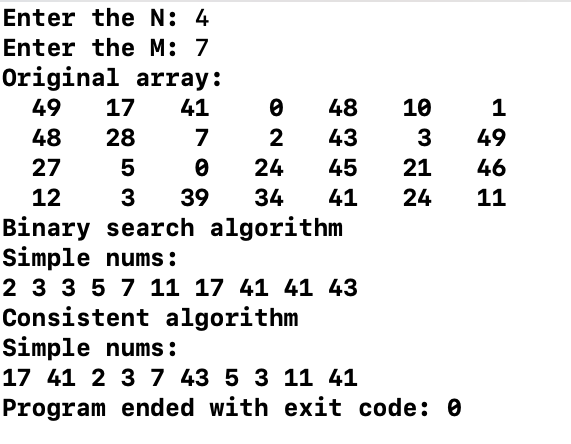


Рисунок 4

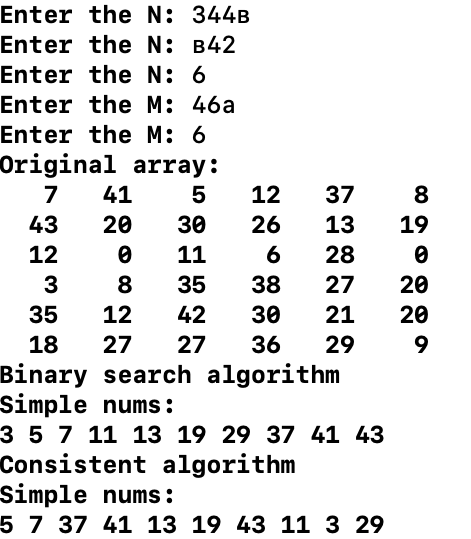


Рисунок 5

1. Вывод

В результате проделанной мною работы я написала программу, которая находит в матрице все простые числа до 50, найденные бинарным и последовательным поисками. Во время выполнения данной лабораторной работы я ознакомилась с основами написания программ на C++, а также смогла понять различие в эффективности различных способов поиска элементов в массиве.

Бинарный поиск в моем случае не совсем эффективен, так как для его использования вначале нужно отсортировать весь массив, и только затем производить поиск элемента. Так как в моем варианте нужно искать не один определенный элемент, а множество (все простые числа до 50), то мне эффективнее использовать последовательный поиск элементов.

Эти знания могут помочь мне в написании более сложных программ в будущем.