

Алгоритмизация и программирование

Программирование на C/C++

(ч.6 – алгоритмы работы с массивами)



```
#include <stdio.h>
int main(void)
{
    printf("Hello, World!\n");
    return 0;
}
```

C/C++

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
<http://www.berkut.mk.ua>

Линейный поиск в массиве

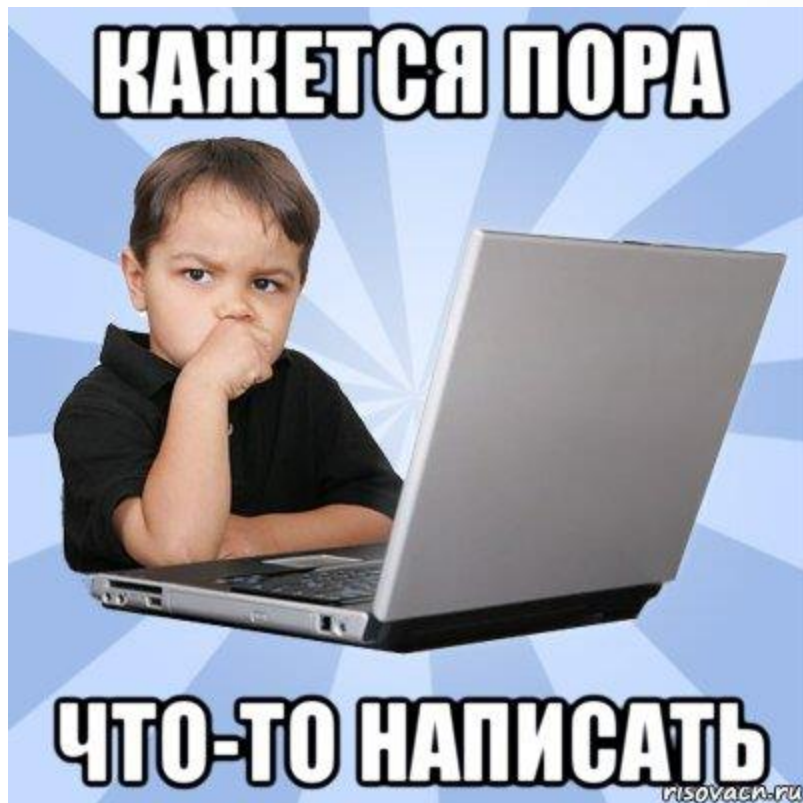
- Линейный поиск в массивах, или как его ещё называют, поиск в ЛОБ эффективен в массивах, с небольшим количеством элементов, причём элементы в таких массивах никак не отсортированы и не упорядочены.
- Алгоритм линейного поиска в массивах последовательно проверяет все элементы массива и сравнивает их с ключевым значением.
- Таким образом, в среднем необходимо проверить половину значений в массиве, чтобы найти искомое значение. Чтобы убедиться, в отсутствии искомого значения необходимо проверить все элементы массива.

Линейный поиск в массиве

```
#include <iostream>
#include <ctime>
#include <cstdlib>
using namespace std;

int main() {
    srand(time(0));
    const int arr_size = 25;
    int arr[arr_size];
    for (int i = 0; i < arr_size; i++) {
        arr[i] = rand() % 50 - 25;
        cout << arr[i] << " ";
    }
    int key;
    cout << "\n key to find: ";
    cin >> key;
    int i = 0;
    while (i < arr_size) {
        if (arr[i] == key) break;
        i++;
    }
    if (i == arr_size) cout << "not found\n";
    else cout << "index=" << i << "\n";
    return 0;
}
```

Демонстрація



Еще один пример:

- Разработаем программу, которая ищет максимальное значение в массиве.
- Поиск в программе реализован согласно алгоритму линейного поиска в массиве.



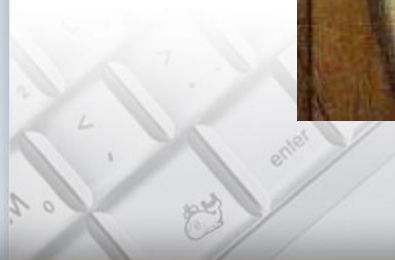
Линейный поиск в массиве

```
#include <iostream>
#include <ctime>
#include <cstdlib>
using namespace std;

int main()
{
    srand(time(0));
    const int arr_size = 25;
    int arr[arr_size];
    for (int i = 0; i < arr_size; i++)
    {
        arr[i] = rand() % 50 - rand() % 50;
        cout << arr[i] << " ";
    }
    cout << "\n";
    int max = arr[0];
    for (int i=1; i<arr_size; i++) {
        if (arr[i] > max) max = arr[i];
    }
    cout << "max = " << max << "\n";
    return 0;
}
```



Демонстрація

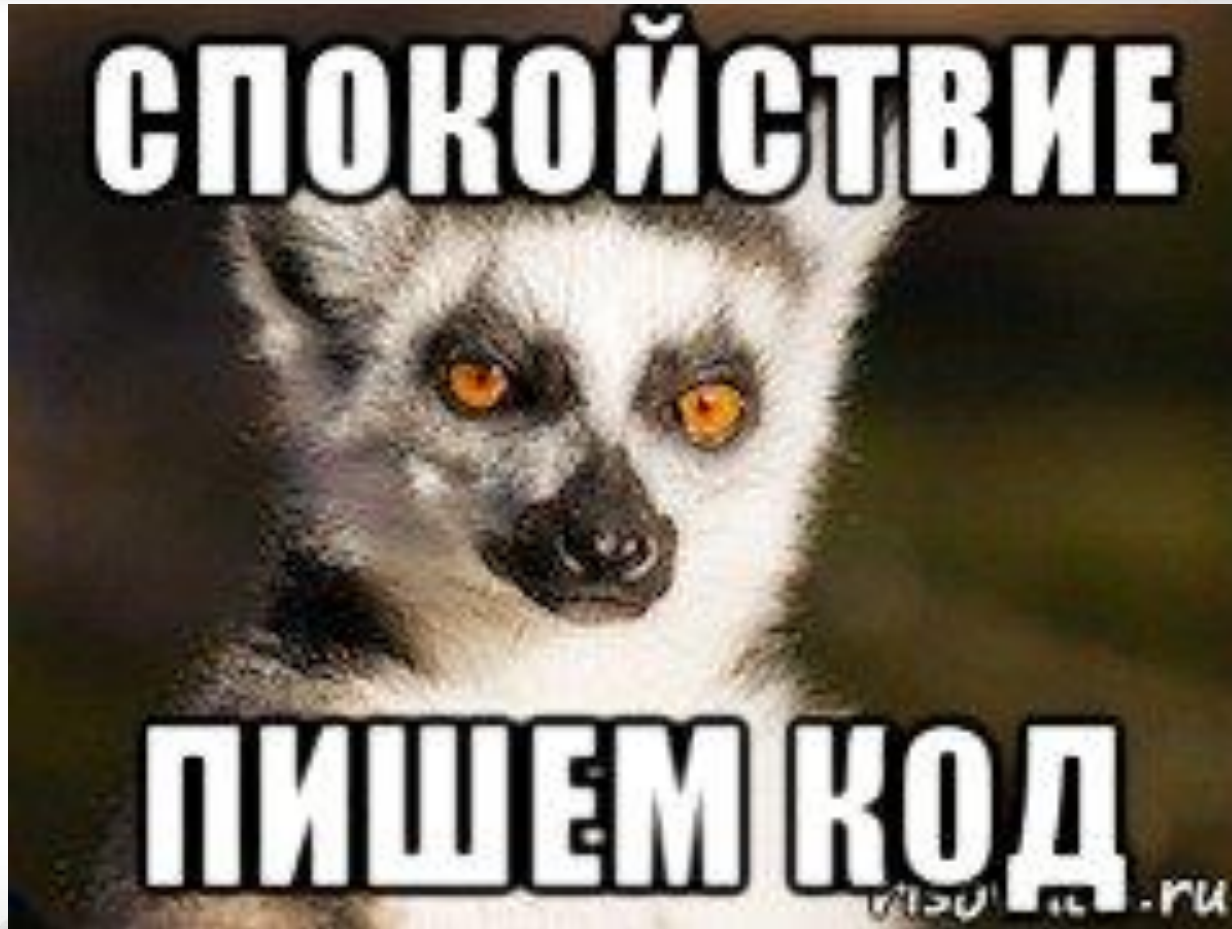


Линейный поиск в массиве

- В двумерных массивах алгоритм линейного поиска не изменится.
- Рассмотрим фрагмент кода — алгоритм линейного поиска в двумерном массиве.

```
int min = arr[0][0];
for (int i = 0; i < rows; i++)
{
    for (int j = 0; j < columns; j++)
    {
        if ( min > arr[i][j] )
            min = arr[i][j];
    }
}
```


Демонстрація



Бинарный поиск в массиве

- Если ваш массив отсортирован, то есть лучший способ поиска – бинарный (или двоичный) поиск
- Рассмотрим этот алгоритм подробнее
- Двоичный(бинарный) поиск — алгоритм поиска элемента в отсортированном массиве.
- Бинарный поиск нашел себе применение в математике и информатике.
- Двоичный поиск можно использовать только в том случае, если есть массив, все элементы которого упорядочены (отсортированы).

Бинарный поиск в массиве

- Алгоритм бинарного поиска применим, если необходимо найти некоторый ключевой элемент в массиве.
- То есть организовать поиск по ключу, где ключ — это определённое значение в массиве.
- Разработаем программу, в которой объявим одномерный массив, и организуем двоичный поиск.
- Объявленный массив нужно инициализировать некоторыми значениями, причём так, чтобы эти значения были упорядочены.

Бинарный поиск в массиве (1/2)

```
#include <iostream>
using namespace std;

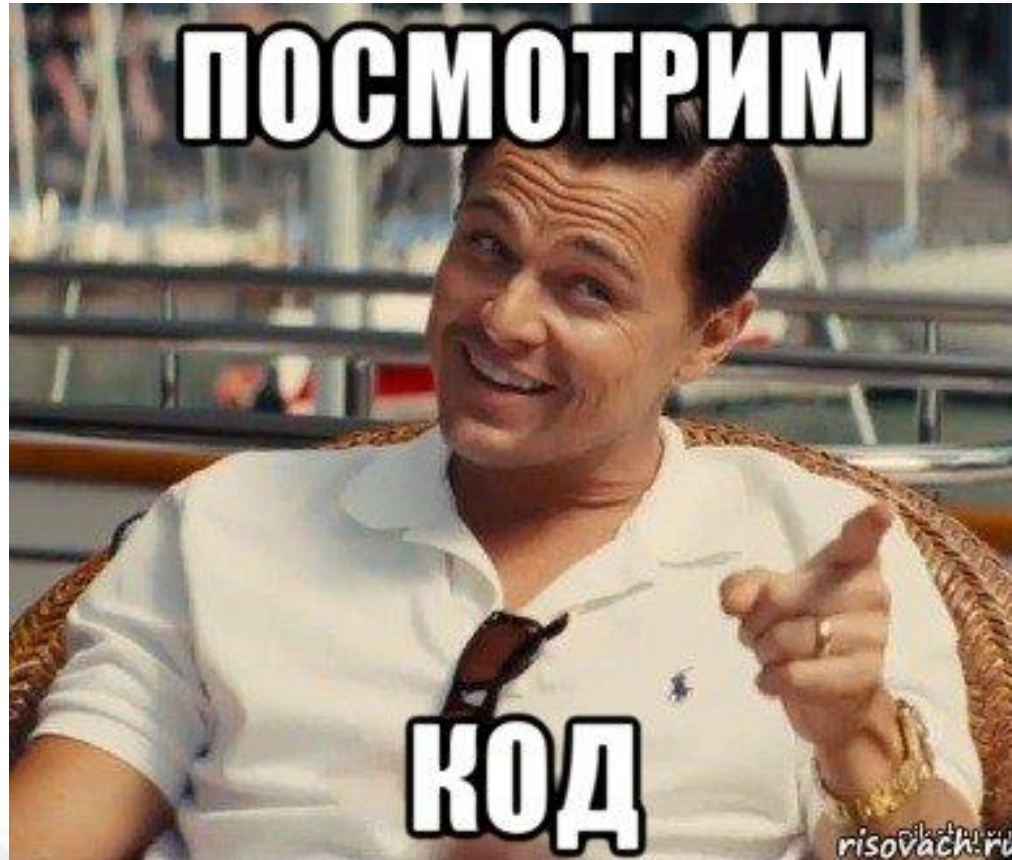
int main()
{
    const int size_array = 10;
    int arr[size_array] = {-8,-7,-6,-6,-4, 2, 6, 7, 8, 15};
    cout << "array[" << size_array << "] = { ";
    for (int i = 0; i < size_array; i++)
    {
        cout << arr[i] << " ";
    }
    cout << " }\n";
    int avg_index = 0;
    int first_index = 0;
    int last_index = size_array - 1;
```

Бинарный поиск в массиве (2/2)

```
//-----  
    int search_value = 15; // искомое (ключевое) значение  
//-----  
    if (last_index == -1) cout << "array is empty" << endl;  
    while (first_index < last_index)  
    {  
        avg_index = (last_index + first_index) / 2;  
        if (search_value <= arr[avg_index])  
            last_index = avg_index ;  
        else first_index = avg_index + 1;  
    }  
    if ( arr[last_index] == search_value)  
        cout << "value is found" << "index = "  
            << last_index << endl;  
    else cout << "value is not found" << endl;  
    return 0;  
}
```



Демонстрація





НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА



Алгоритмизация и программирование

Программирование на C/C++

(ч.6 – алгоритмы работы с массивами)



```
#include <stdio.h>
int main(void)
{
    printf("Hello, World!\n");
    return 0;
}
```

C/C++

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
<http://www.berkut.mk.ua>