

Алгоритмизация и программирование

Программирование на Kotlin

(ч.6 – алгоритмы работы с массивами)



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

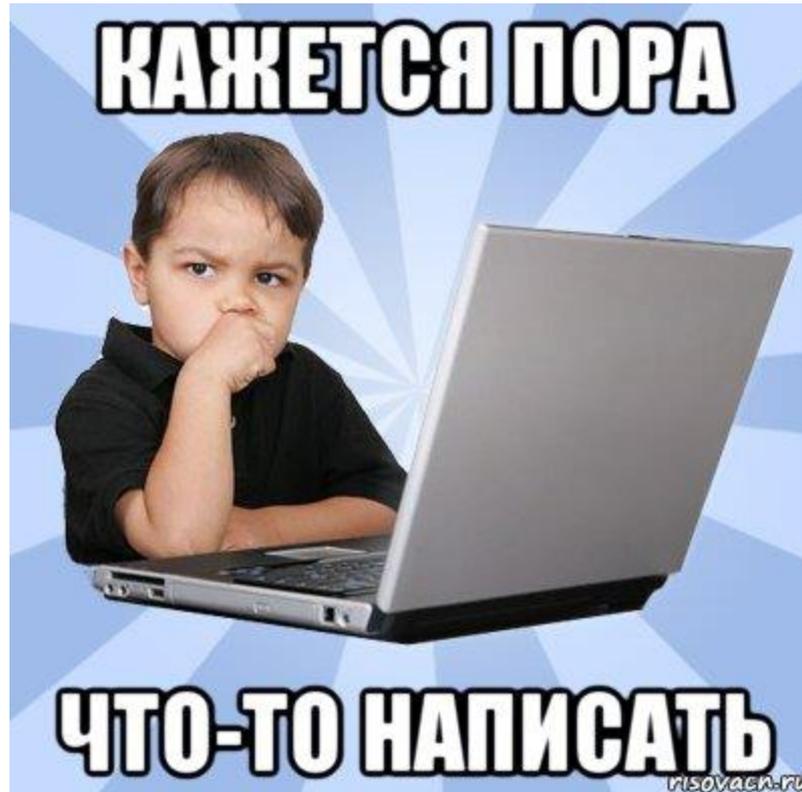
Линейный поиск в массиве

- 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

```
fun main() {  
    val a = Array(25) { (-25..25).random() }  
    print("Key to find:")  
    val key = readLine()!!.toInt()  
  
    val index = find(a, key)  
    if (index<0) println("not found")  
    else println("found index = $index")  
  
    println(a.toList())  
}  
  
fun find(a: Array<Int>, key :Int) :Int {  
    for (i in a.indices) {  
        if (a[i]==key) return i  
    }  
    return -1  
}
```

Демонстрація



Еще один пример:

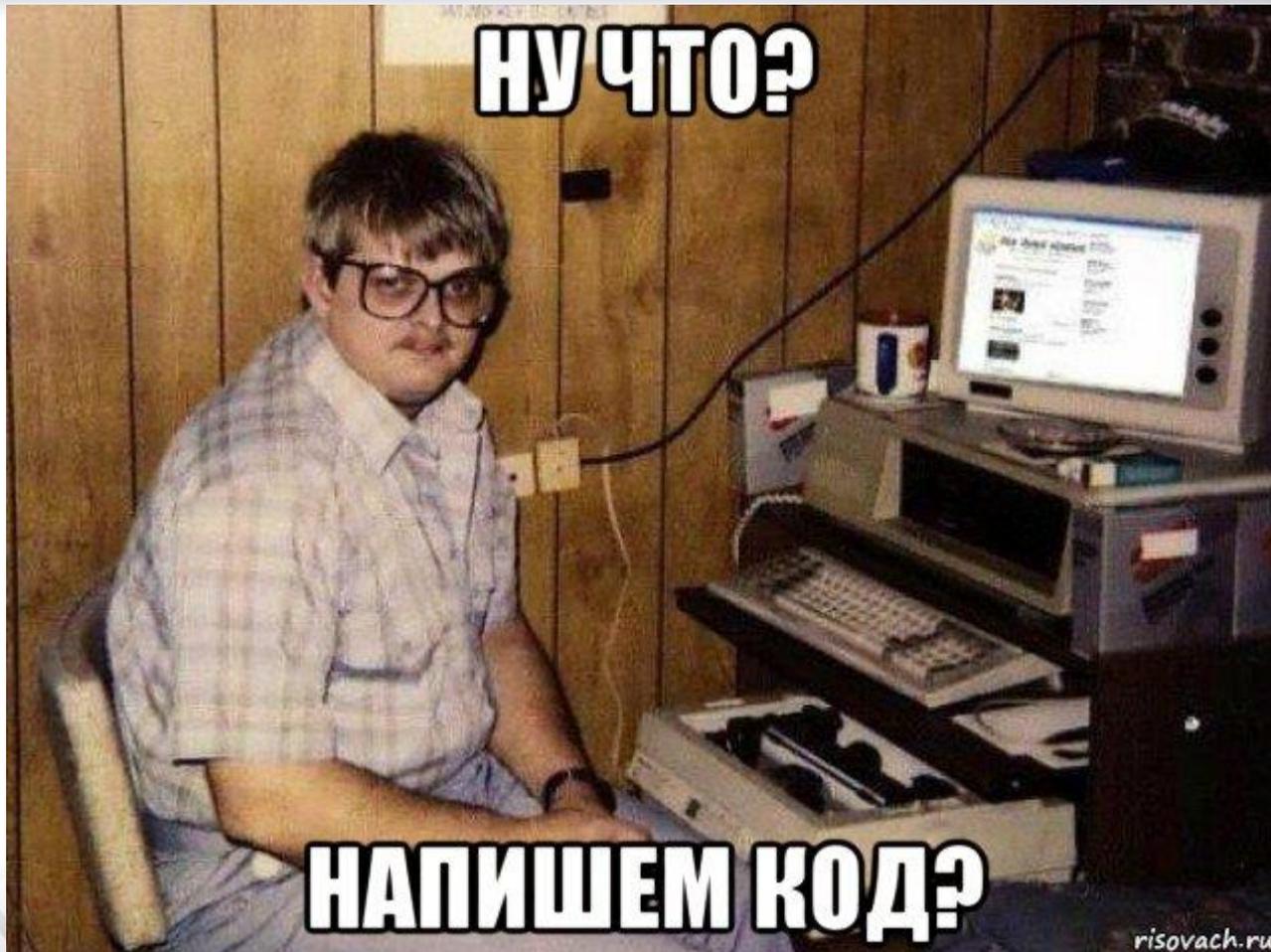
- Разработаем программу, которая ищет **максимальное значение** в массиве.
- Поиск в программе реализован согласно **алгоритму линейного поиска** в массиве.

Линейный поиск в массиве

```
fun main() {  
    val a = Array(25) { (-25..25).random() }  
  
    val indexMax = findNumOfMax(a)  
    println("index of max = $indexMax")  
  
    println(a.toList())  
}  
  
fun findNumOfMax(a: Array<Int>) :Int {  
    var num = 0  
    for (i in a.indices) {  
        if (a[i] > a[num]) num = i  
    }  
    return num  
}
```



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

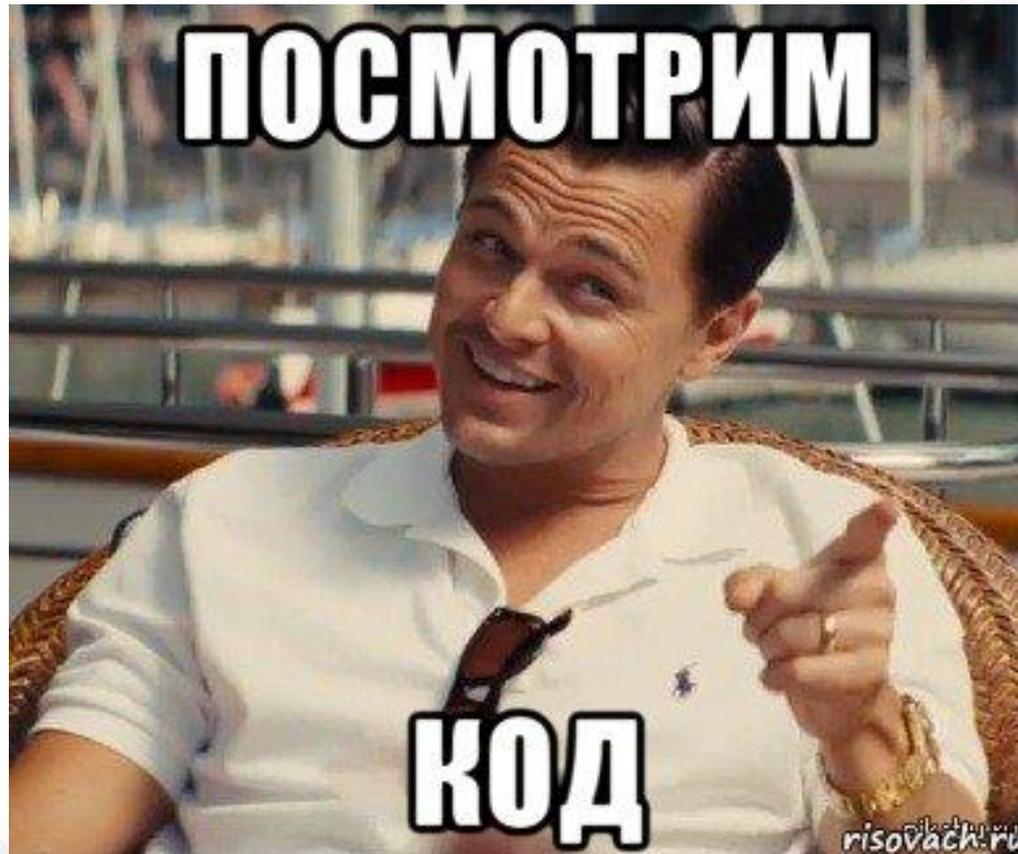
(1/2)

```
fun main() {  
    val a = Array(25) { (-25..25).random() }.sortedArray()  
    print("Key to find:")  
    val key = readLine()!!.toInt()  
  
    val index = find(a, key)  
    if (index < 0) println("not found $index")  
    else println("found index = $index")  
  
    println(a.toList())  
}
```

Бинарный поиск в массиве (2/2)

```
fun find(a: Array<Int>, key :Int) : Int {
    var firstIndex = 0
    var lastIndex = a.size-1
    if (lastIndex== -1) return -1
    while (firstIndex < lastIndex) {
        val avgIndex = (lastIndex + firstIndex) / 2
        if (key < a[avgIndex])
            lastIndex = avgIndex-1
        else
            if (key > a[avgIndex]) firstIndex = avgIndex + 1
            else return avgIndex
    }
    return -(firstIndex+1)
}
```

Демонстрація





НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА



Алгоритмизация и программирование

Программирование на Kotlin

(ч.6 – алгоритмы работы с массивами)



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