In [42]:

```
import numpy as np
import random
import time

# Бинарный поиск
def binarySearch(array, item):
    first = 0
    last = len(array) - 1

    middle = last // 2

    while array[middle] != item and first <= last:
        if item > array[middle]:
            first = middle + 1
        else:
            last = middle - 1
        middle = (first + last) // 2

    if first > last:
        return -1
    else:
        return middle

def interpolationSearch(array, item):
    left = 0
    right = len(array) - 1

    while array[left] < item and item < array[right]:
        middle = left + ((item - array[left]) * (right - left)) // (array[right] - array[left])
        current = array[middle]

        if current < item:
            left = middle + 1
        else:
            if current > item :
                right = middle - 1
            else:
                return middle;

    if array[left] == item:
        return left
    else:
        if array[right] == item:
            return right
        else:
            return -1
```

In [43]:

```
# PROCESSING ARRAYS

ARRAY_MAX_ITEM = 100

def findIndex(arr, p, r):
    pivot = arr[(p + r) // 2] # Округление
    i = p
    j = r

    while i <= j:
        while arr[i] < pivot:
            i = i + 1

        while arr[j] > pivot:
            j = j - 1

        if i <= j:
            arr[i], arr[j] = arr[j], arr[i]
            i = i + 1
            j = j - 1

    return i

def quickSort(arr, p, r):
    if p < r:
        index = findIndex(arr, p, r)

        if p < index - 1:
            quickSort(arr, p, index-1)

        if r > index:
            quickSort(arr, index, r)

    return arr

def makeSortedArray(length):
    array = []

    for i in range(length):
        array.append(int((random.random() * (ARRAY_MAX_ITEM+1))))

    array = quickSort(array, 0, len(array)-1)

    return array
```

In [60]:

```
# LOGIC

N = 50
ARRAYS_SIZE = 100

for n in range(2, N):
    # Number for all searches
    numberToFind = int(random.randint(1, ARRAY_MAX_ITEM))

    # Make N arrays for searches
    arraysList = []
    for i in range(ARRAYS_SIZE):
        arraysList.append(makeSortedArray(n))

    # ------------------------------------------

    print("----- For length = {} and number = {} -----".format(n, numberToFind))

    # BindarySearch for all arrays
    allBinaryTime = 0

    for i in range(ARRAYS_SIZE):
        arr = arraysList[i]
        item = numberToFind

        startTime = time.time()
        binarySearch(arr, item)
        allBinaryTime += time.time() - startTime

    avgBinaryTime = allBinaryTime / ARRAYS_SIZE;

    # print("Binary search = {}".format(avgBinaryTime))

    # ------------------------------------------

    # Interpolation Search
    allInterpolationTime = 0

    for i in range(ARRAYS_SIZE):
        arr = arraysList[i]
        item = numberToFind

        startTime = time.time()
        binarySearch(arr, item)
        allInterpolationTime += time.time() - startTime

    avgInterpolationTime = allInterpolationTime / ARRAYS_SIZE

    # print("Interpolation search = {}".format(avgInterpolationTime))

    # ------------------------------------------

    sign = ""
    if(avgBinaryTime < avgInterpolationTime):
        sign = ">"
        print("B {} I for {}".format(sign, avgInterpolationTime - avgBinaryTime))
        break
    else:
        sign = "<"
        print("B {} I".format(sign))
```

```
----- For length = 2 and number = 55 -----
B < I
----- For length = 3 and number = 71 -----
B < I
----- For length = 4 and number = 80 -----
B < I
----- For length = 5 and number = 91 -----
B < I
----- For length = 6 and number = 68 -----
B < I
----- For length = 7 and number = 63 -----
B < I
----- For length = 8 and number = 33 -----
B < I
----- For length = 9 and number = 64 -----
B < I
----- For length = 10 and number = 96 -----
B < I
----- For length = 11 and number = 29 -----
B < I
----- For length = 12 and number = 7 -----
B > I for 2.145767211914059e-08
```

```
In [ ]:
```