

Bisection method 二分法

BinarySearch 二分查找

When and How?

When: search an index or an element in a regular range

Time complexity: $\log(n)$

Template: https://github.com/MagicienDeCode/LeetCode_Solution/blob/master/src/BinarySearch.md

- ⇒ left right ?
- ⇒ `while(left + 1 < right)` // avoid dead loop
- ⇒ conditions
- ⇒ right first or left first ?
- ⇒ `left = mid / right = mid`

1. search the (exact) (position / element)
2. search the (first / last) (position / element)
3. search with complex conditions
4. binary search the result range with a verification function

34. Find First and Last Position of Element in Sorted Array

```
fun searchFirstPosition(nums: IntArray, target: Int): Int {  
    if (nums.isEmpty()) {  
        return -1  
    }  
    var left = 0  
    var right = nums.size - 1  
    while (left + 1 < right) {  
        val mid = left + (right - left) / 2  
        when {  
            nums[mid] == target -> right = mid  
            nums[mid] < target -> left = mid  
            nums[mid] > target -> right = mid  
        }  
    }  
    return when (target) {  
        nums[left] -> left  
        nums[right] -> right  
        else -> -1  
    }  
}
```

```
fun searchLastPosition(nums: IntArray, target: Int): Int {  
    if (nums.isEmpty()) {  
        return -1  
    }  
    var left = 0  
    var right = nums.size - 1  
    while (left + 1 < right) {  
        val mid = left + (right - left) / 2  
        when {  
            nums[mid] == target -> left = mid  
            nums[mid] < target -> left = mid  
            nums[mid] > target -> right = mid  
        }  
    }  
    return when (target) {  
        nums[right] -> right  
        nums[left] -> left  
        else -> -1  
    }  
}
```

1. search the (exact) (position / element)

704. Binary Search (exact element)

167. Two Sum II - Input array is sorted (exact element)

35. Search Insert Position ([first index] <= target)

374. Guess Number Higher or Lower

392. Is Subsequence (follow up using a hashmap to store index of char, then binary search in list)

852. Peak Index in a Mountain Array

162. Find Peak Element

74. Search a 2D Matrix (binary search x 2)

240. Search a 2D Matrix II (foreach then search)

475. Heaters

2. search the (first / last) (position / element)

69. Sqrt(x) (last element $\wedge^2 \leq target$)

34. Find First and Last Position of Element in Sorted Array

278. First Bad Version

350. Intersection of Two Arrays II (first & last then count the total)

1337. The K Weakest Rows in a Matrix (first or last then count)

1351. Count Negative Numbers in a Sorted Matrix

300. Longest Increasing Subsequence (LIS)

441. Arranging Coins

454. 4Sum II

528. Random Pick with Weight (transform then binary search)

497. Random Point in Non-overlapping Rectangles

911. Online Election

981. Time Based Key-Value Store

315. Count of Smaller Numbers After Self

327. Count of Range Sum

354. Russian Doll Envelopes (LIS)

862. Shortest Subarray with Sum at Least K (Deque)

887. Super Egg Drop (Math, also DP)

3. search with complex conditions

33. Search in Rotated Sorted Array

81. Search in Rotated Sorted Array II

153. Find Minimum in Rotated Sorted Array & 154

275. H-Index II

658. Find K Closest Elements

710. Random Pick with Blacklist

4. binary search the result range with a verification function

[378. Kth Smallest Element in a Sorted Matrix](#)

[718. Maximum Length of Repeated Subarray \[DP\]](#)

[875. Koko Eating Bananas](#)

[1011. Capacity To Ship Packages Within D Days](#)

[1482. Minimum Number of Days to Make m Bouquets](#)

[1201. Ugly Number III](#)

[1283. Find the Smallest Divisor Given a Threshold](#)

[1300. Sum of Mutated Array Closest to Target](#)

[1292. Maximum Side Length of a Square with Sum Less than or Equal to Threshold](#)

[540. Single Element in a Sorted Array](#)

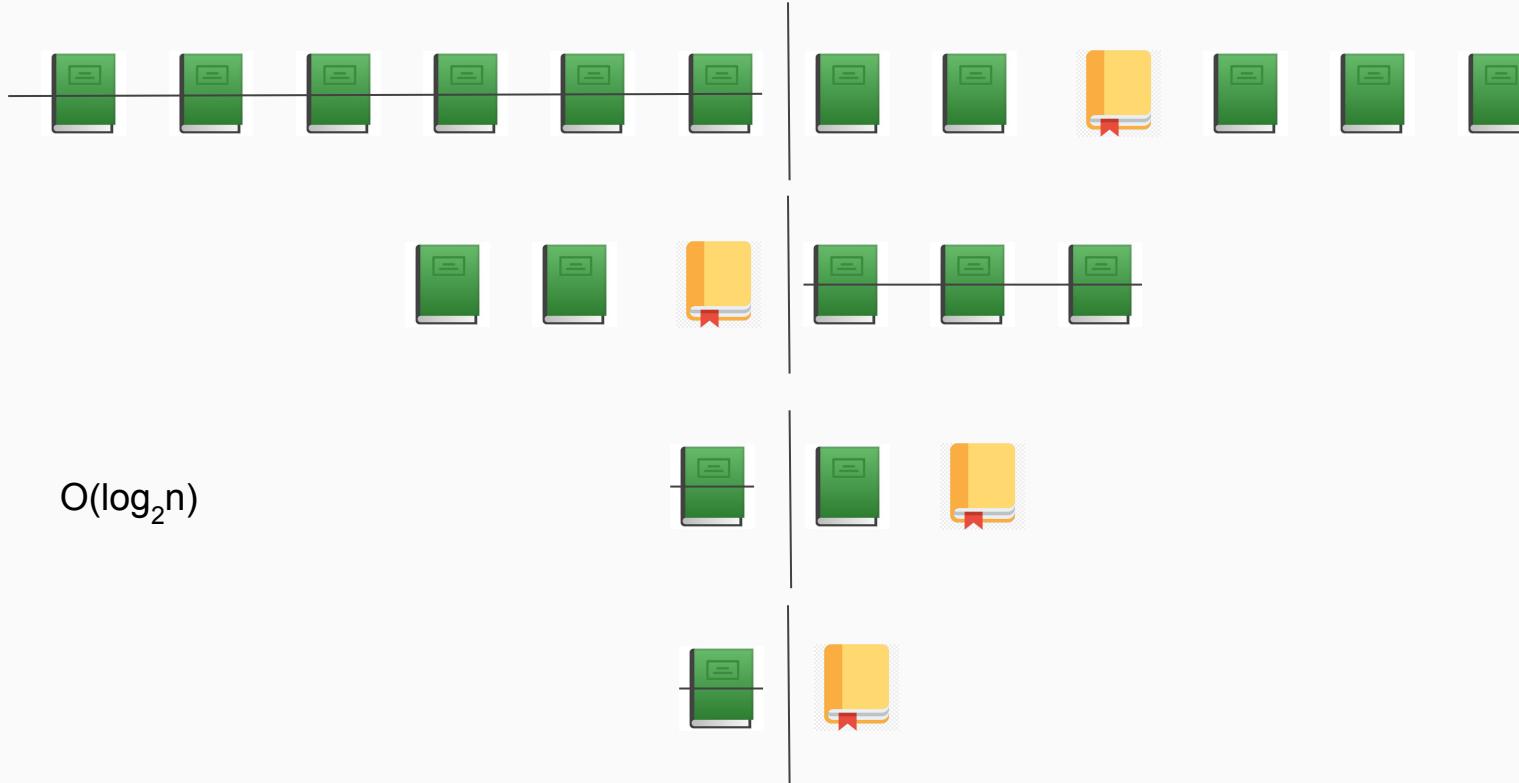
[410. Split Array Largest Sum](#)

[668. Kth Smallest Number in Multiplication Table](#)

[778. Swim in Rising Water \(DFS\)](#)

[1044. Longest Duplicate Substring \(Rabin-Karp\)](#)

Bisection 二分法



BinarySearch 二分查找

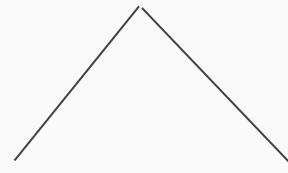
- Given collection has a regular sequence
- Find a position that satisfy some conditions

-1, 2, 5, 7, 11, 100, 150 ...



1, 1, 1, 1, 1, 0, 0, 0, 0

-9, -5, 1, 8, 17, 6, 5, 2, 0, -3, -10



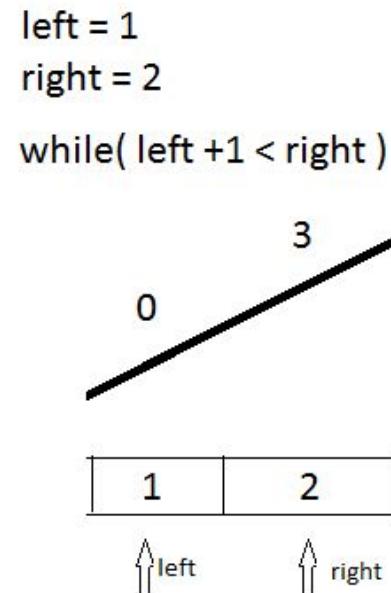
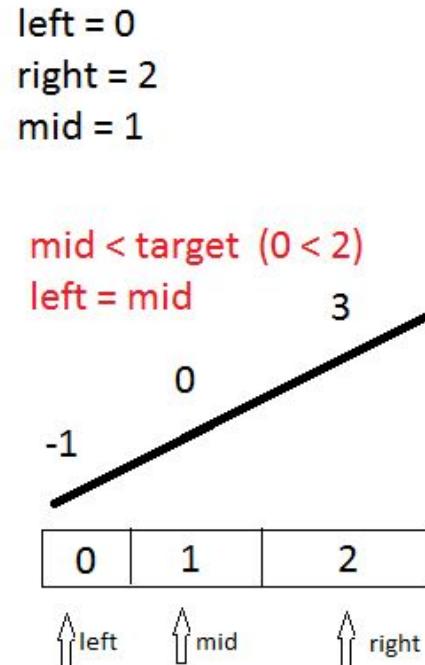
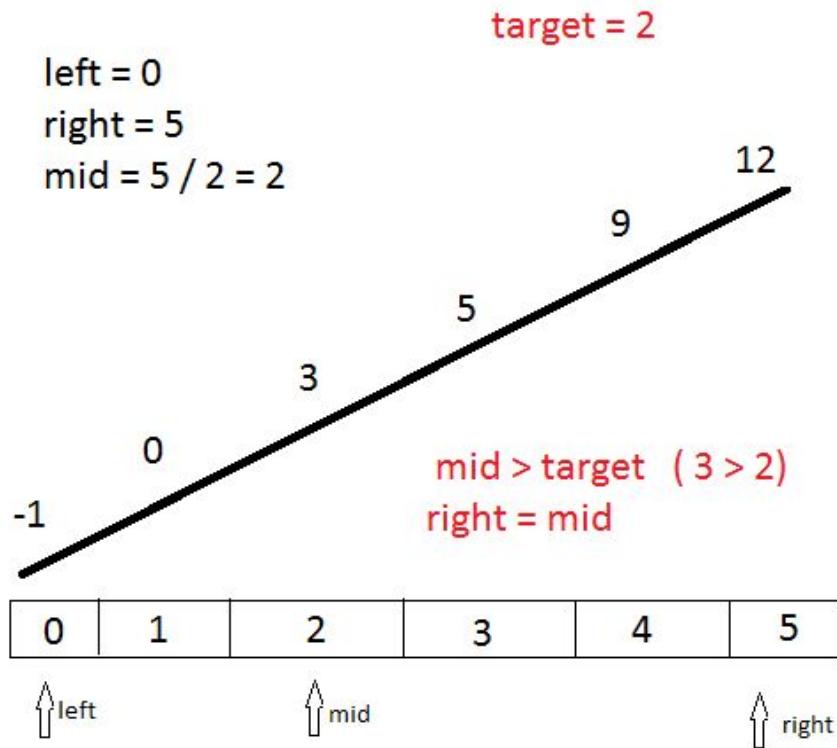
Test Driven Development (TDD)

三板斧

1 ⇒ Write some unit tests

2 ⇒ Copy binarysearch template

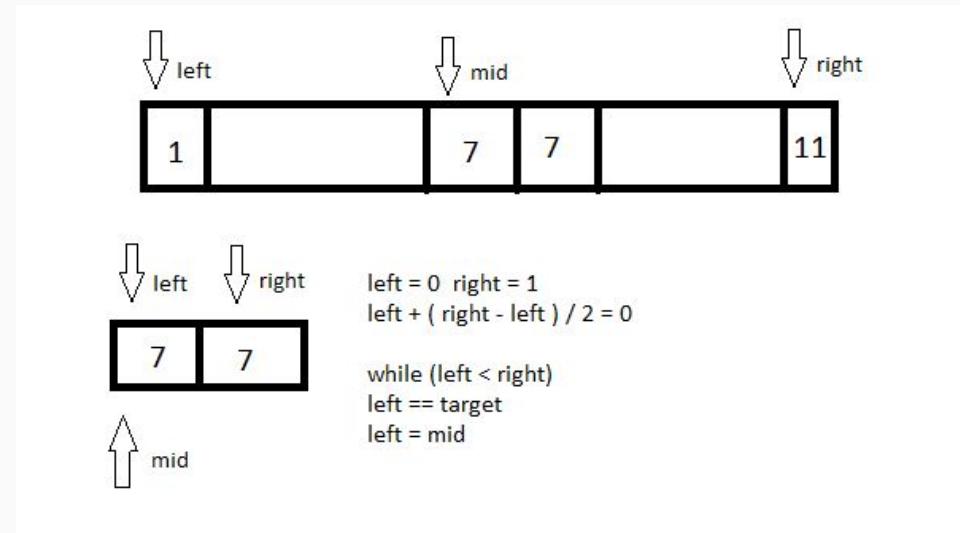
3 ⇒ Modify the binarysearch template



Template Java / Kotlin

1. `while (left +1 < right)`
2. `mid = left + (right - left) / 2`
3. `mid == target ? mid > target ? mid < target
left = mid or left = mid +1`
4. `left? right?`

`Int.MAX_VALUE = 2147483647`

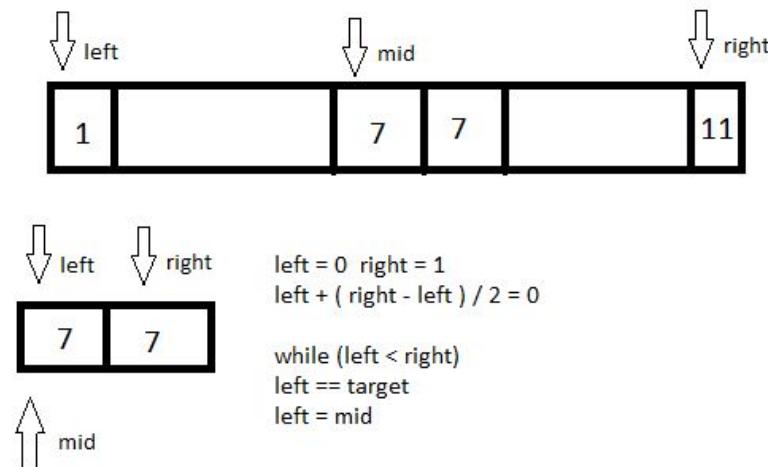


<https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/>

<https://leetcode.com/tag/binary-search/>

34. Find first and last position

<https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/>



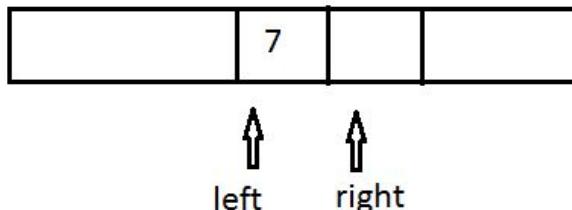
First position:
if (mid == target) right = mid

Last position:
if (mid == target) left = mid

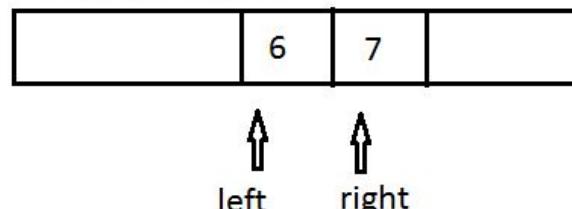
35. Search Insert Position

<https://leetcode.com/problems/search-insert-position/>

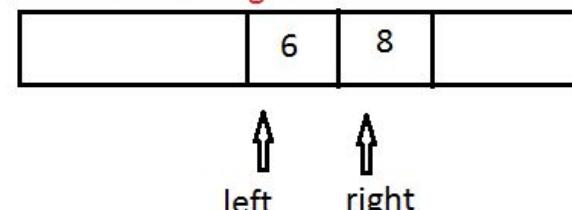
target \leq [left]
==> left



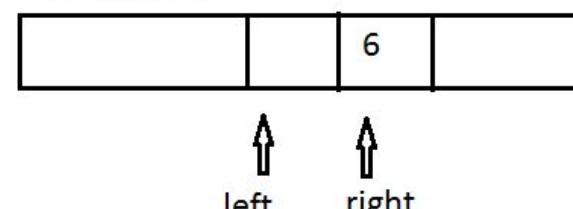
target == [right]
==> right = left + 1



target = 7 [Left] $<$ target $<$ [right]
==> left + 1 = right



target \geq [right]
==> right + 1



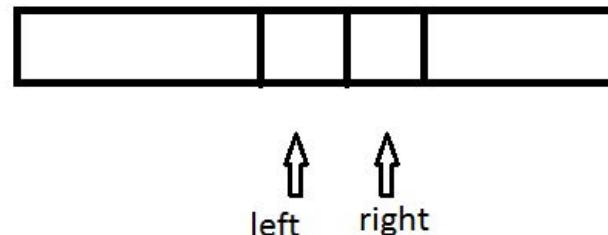
69.Sqrt(x)

<https://leetcode.com/problems/sqrtx/>

```
4 = 2 * 2  
8 = 2.8... * 2.8... ==> 2
```

```
left = 0  
// Int.MAX_VALUE = 2147483647 Sqrt(~) = 46340.95...  
right = 46340
```

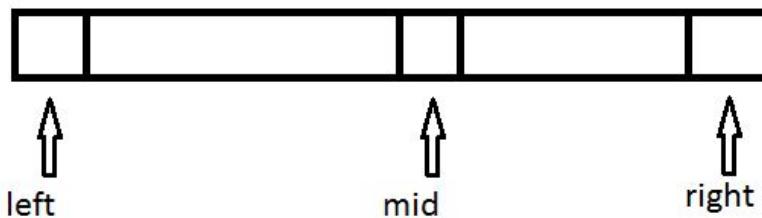
```
target = 8  
left = 2 , right = 3
```



```
target = 8  
left = 1 , right = 2  
right * right <= target  
==> right <= target / right  
==> right
```

167. Two Sum II - Input array is sorted

<https://leetcode.com/problems/two-sum-ii-input-array-is-sorted/>



Case 1: $[\text{left}] + [\text{right}] < \text{target} \Rightarrow \text{left} = \text{mid}$
 $\text{mid?} \Rightarrow$

$[\text{mid}] + [\text{right}] \geq \text{target}$

~~1 3 5 7 9 11 13 14 15~~

$\text{target} = 25 (14 + 11)$

$[\text{right}] = 15 \ [\text{mid}] = 11$

Case 2: $[\text{left}] + [\text{right}] > \text{target} \Rightarrow \text{right} = \text{mid}$
 $\text{mid?} \Rightarrow$

$[\text{mid}] + [\text{left}] \leq \text{target}$

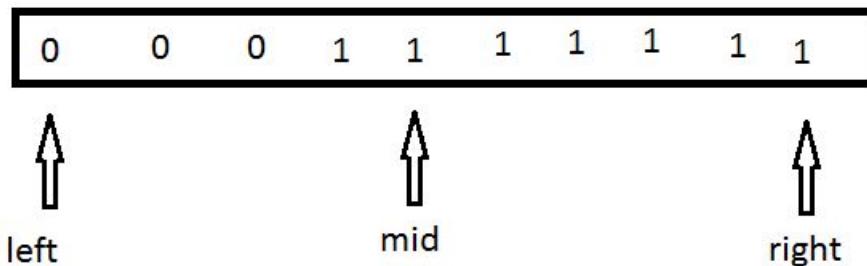
~~0 3 5 7 9 11 13 14 15~~

$\text{target} = 12 (3+9)$

$[\text{left}] = 0 \ [\text{mid}] = 11$

278. First Bad Version

<https://leetcode.com/problems/first-bad-version/>

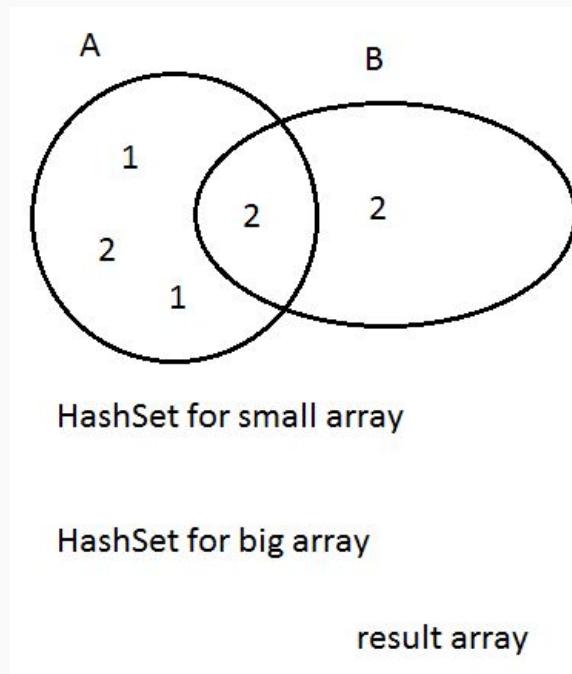


mid isBadVersion ==> right = mid

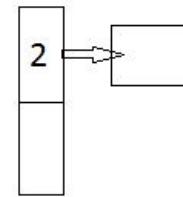
mid ! isBadVersion ==> left = mid

349. Intersection of Two Arrays & 350. Intersection of Two Arrays II

https://github.com/MagicienDeCode/LeetCode_Solution

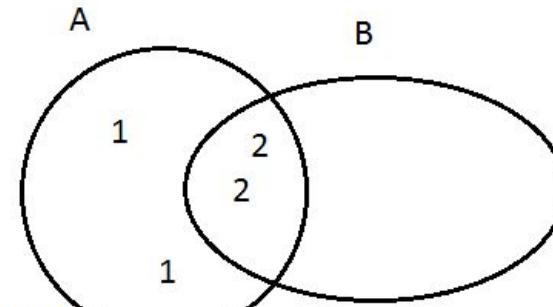


HashMap < Int, Int >



Key = element in small array

value = total number the element in small array



Foreach element in big array, search if element present in map, if yes and the count > 0, add it to result.

$O(n + m)$

367. Valid Perfect Square

<https://leetcode.com/problems/valid-perfect-square/>

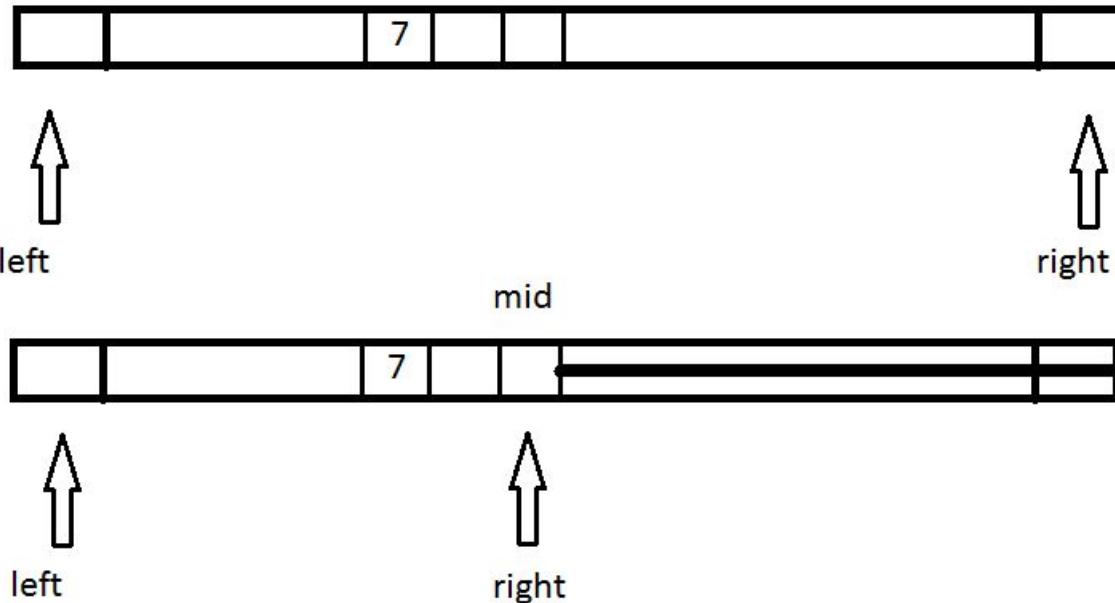


`mid * mid == target`

`right * right == target`

374. Guess Number Higher or Lower

<https://leetcode.com/problems/guess-number-higher-or-lower/>



392. Is Subsequence

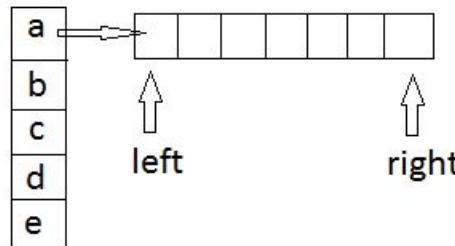
<https://leetcode.com/problems/is-subsequence/>

a h b g d c
↑ ↑ ↑

```
indexOfS = 0
foreach char in t:
    if (char == s[indexOfS]){
        indexOfS++
    }

    indexOfS ==? s.length
O(t)
```

Follow up $O(k*t)$

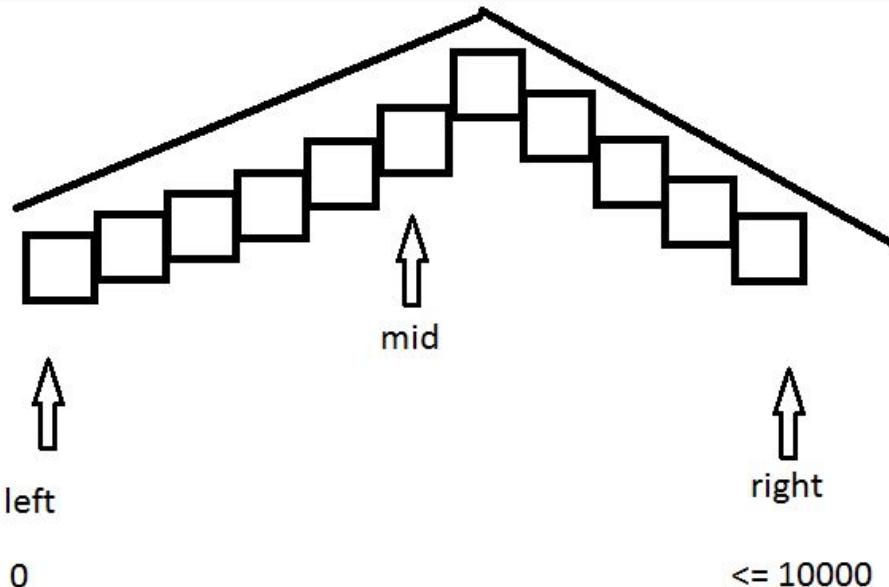


$O(k*100*\log(500\ 000 / 26))$

```
foreach S in S1...Sk:
    foreach char in S:
        binarysearch in mans.values
        a {100,323,4434,32333...}
        b {23,56,136,456,7237...}
        C {2,65,154,232,1511,6251...}
```

852. Peak Index in a Mountain Array & 162. Find Peak Element

<https://leetcode.com/problems/peak-index-in-a-mountain-array/>



$[\text{mid}-1] < [\text{mid}] > [\text{mid}+1] \Rightarrow \text{return mid}$

$[\text{mid}-1] < [\text{mid}] < [\text{mid}+1]$ $\Rightarrow \text{left} = \text{mid}$

$[\text{mid}-1] > [\text{mid}] > [\text{mid}+1]$ $\Rightarrow \text{right} = \text{mid}$

$[\text{left}-1] < [\text{left}] > [\text{left}+1]$ $\Rightarrow \text{left}$

else right

$[\text{mid}-1] < [\text{mid}] \Rightarrow \text{left} = \text{mid}$

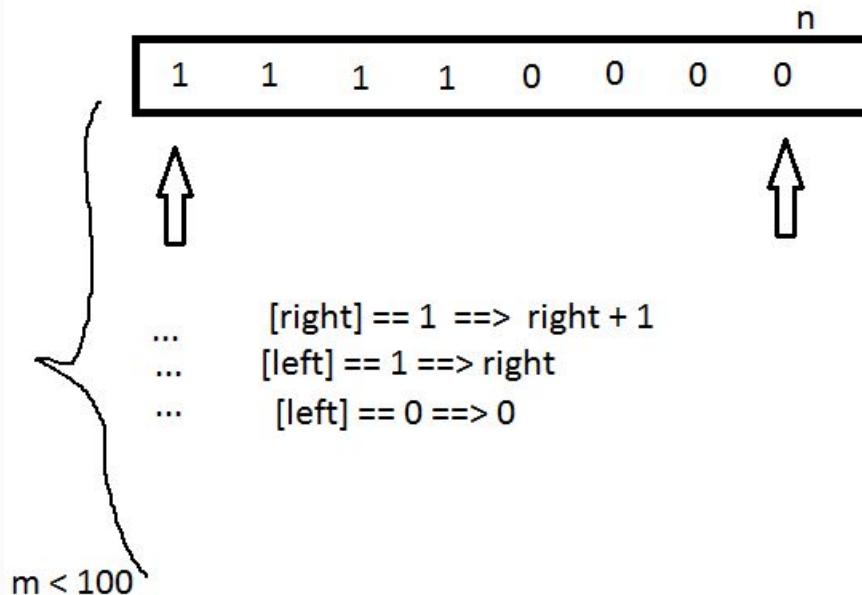
else $\rightarrow \text{right} = \text{mid}$

$[\text{left}] > [\text{right}] \Rightarrow \text{left}$

else $\Rightarrow \text{right}$

1337. The K Weakest Rows in a Matrix

<https://leetcode.com/problems/the-k-weakest-rows-in-a-matrix/>

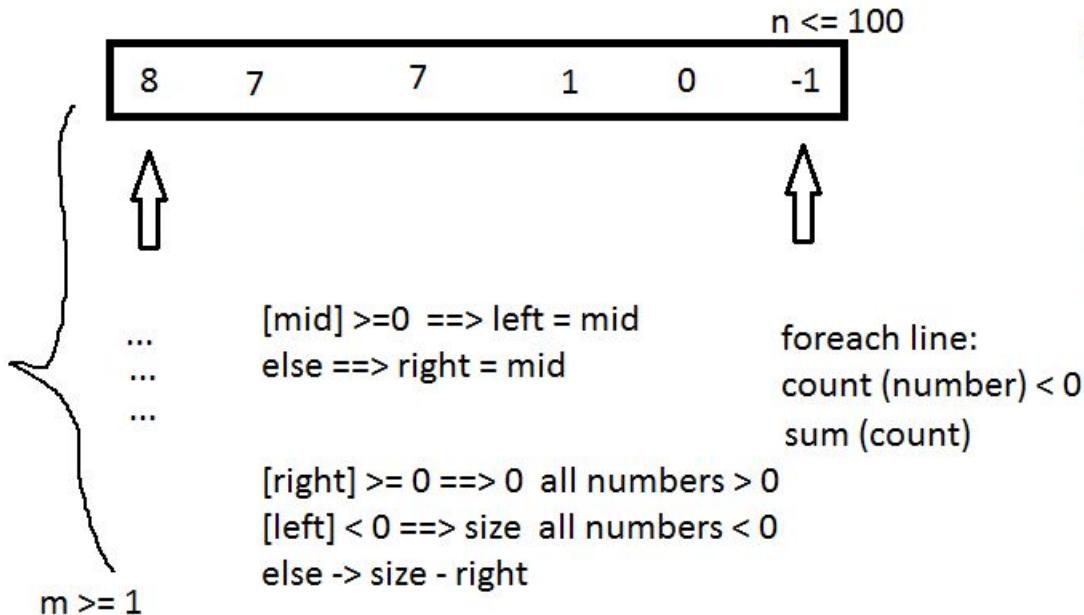


list < Pair < Int, Int > >
↑ count (1) : total 1 in this
line
index: 0 ~ m

sort by Pair.second
get subList 0,k then return Pair.first

1351. Count Negative Numbers in a Sorted Matrix

<https://leetcode.com/problems/count-negative-numbers-in-a-sorted-matrix/>



functional programming:
is a programming paradigm
that treats computation as the
evaluation of mathematical
functions and avoids changing
state and mutable data.

29. Divide Two Integers

<https://leetcode.com/problems/divide-two-integers/>

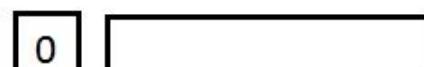
1 字节Byte = 8 位bit

Int ==> 4 字节 * 8 == 32 位

最高位要表示正负

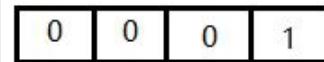


负数



正数

31 位



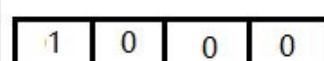
$$1 \ll 0 = 1$$



$$1 \ll 1 = 2$$



$$1 \ll 2 = 4$$



$$1 \ll 3 = 8$$

$$1 * 2^3$$

$$1*8 + 0*4 + 0*2 + 0*1$$

0 111 1111 1111 1111 ==> 2147483647 MAX_VALUE

1 111 1111 1111 1111 ==> -2147483647

0 000 0000 0000 0000 ==> 0

1 000 0000 0000 0000 ==> -2147483648 MIN_VALUE

total= 10, divisor= 3, count = 0

$$3 \ll 0 = 3$$

$$3 \ll 1 = 6$$

$$3 \ll 2 = 12 > 10$$

count = 2

$$\text{total} - 3 \ll 1 = 4$$

总共 $1 \ll 1 = 2$ 个 3

$$\text{total} = 4, \text{divisor} = 3$$

$$3 \ll 0 = 3 \quad \text{那么需要 } 2*3 + 1*3$$

$$3 \ll 1 = 6 \quad ==> 2 + 1 = 3$$

$$10 / 3 = 3 ==> 3 * 2^1 + 3 * 2^0$$

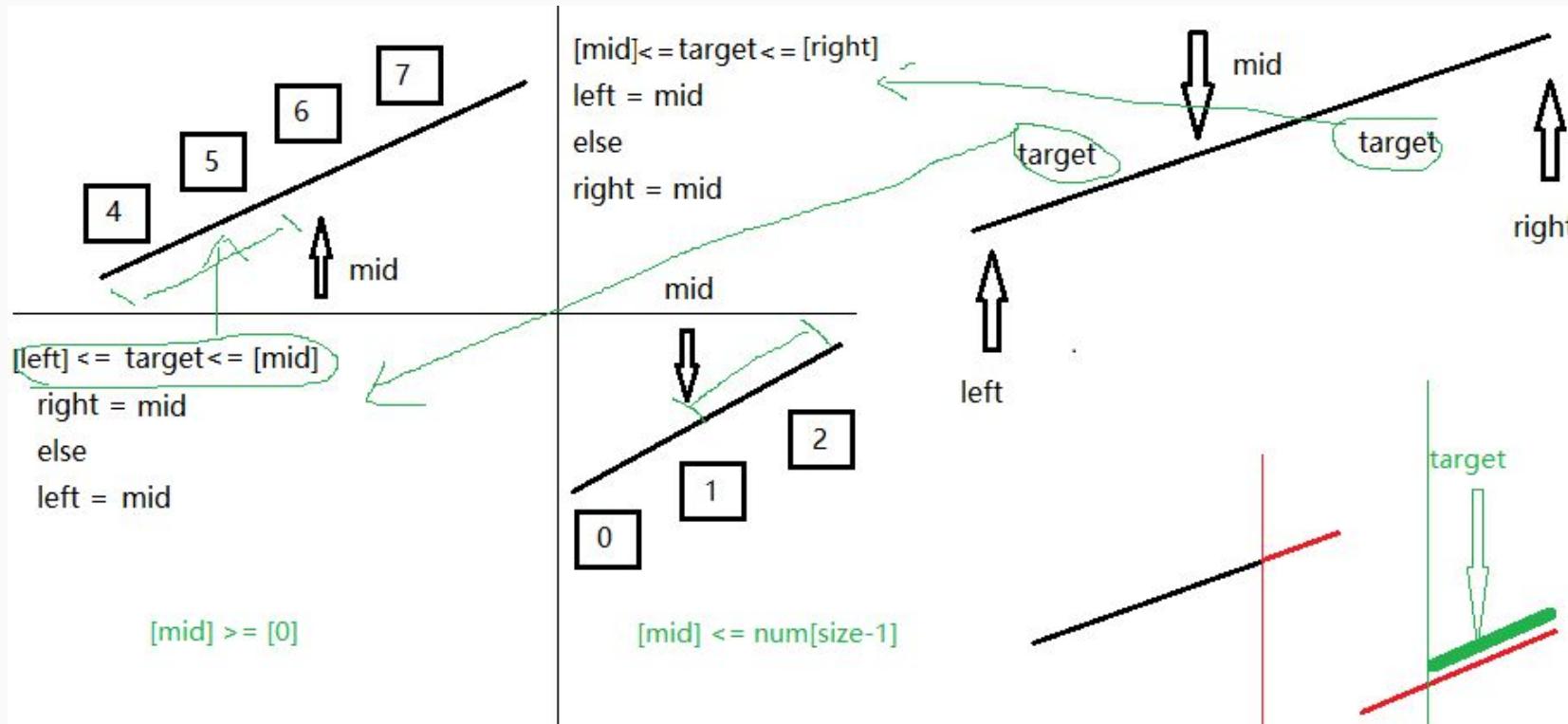
$$23 / 3 = 7 ==> 3 * 2^2 + 3 * 2^1 + 3 * 2^0$$

O(nlogn) > O(n) > O(log(n)+log(n/2)+log(n/4)) ... > O(log(n))

O(nlogn) > O(n) > O(log(n)+log(n)) ... > O(log(n))

33. Search in Rotated Sorted Array

<https://leetcode.com/problems/search-in-rotated-sorted-array/>



50. Pow(x, n)

<https://leetcode.com/problems/powx-n/>

`n % 2 == 0`

$$x^n = x^{n/2 + n/2} = x^{n/2} * x^{n/2}$$

`n % 2 == 1`

$$\begin{aligned} n &= x^{(n-1)/2 + (n-1)/2 + 1} \\ &= x^{(n-1)/2} \cdot x^{(n-1)/2} \cdot x \end{aligned}$$

$3^{16} =$

$$3^{16} = 3 \cdot 3$$

$$= 3 \cdot 3$$

$$= 3 \cdot 3$$

$$= 3 \cdot 3$$

$O(n) ==> O(\log(n))$

$O(n) \Rightarrow O(\log(n))$

74. Search a 2D Matrix

<https://leetcode.com/problems/search-a-2d-matrix/>

```
matrix = [  
    [ 1 ,   3,      5,      7 ],  
    [ 10 ,  11,     16,     20 ],  
    [ 23 ,  30,     34,     50 ]  
]
```

1. find target is in which row:
binary search
 $O(\log(m))$



2. find target in row:
binary search

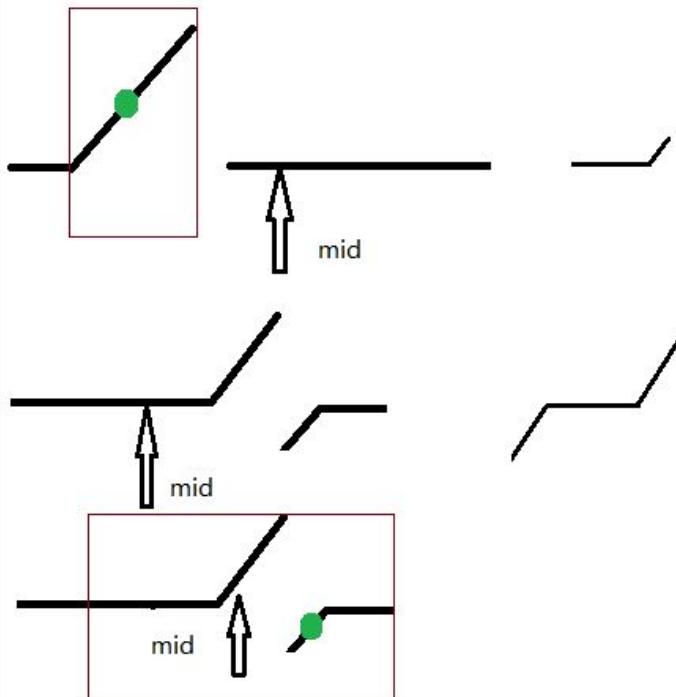


$O(\log(n))$

1. verify the given array is not null or empty
2. relation between mid / target
3. at the end, $\text{left} + 1 == \text{right}$,
return left or right or -1

81. Search in Rotated Sorted Array II

<https://leetcode.com/problems/search-in-rotated-sorted-array-ii/>



Can't use template "Search in Rotated Sorted Array",
because we do not know should ignore which part if the value of
mid is equal to left or right.

So the first step is to pass the duplicate value,
when the value of mid is equal to left, left ++
or it's equal to right, right --

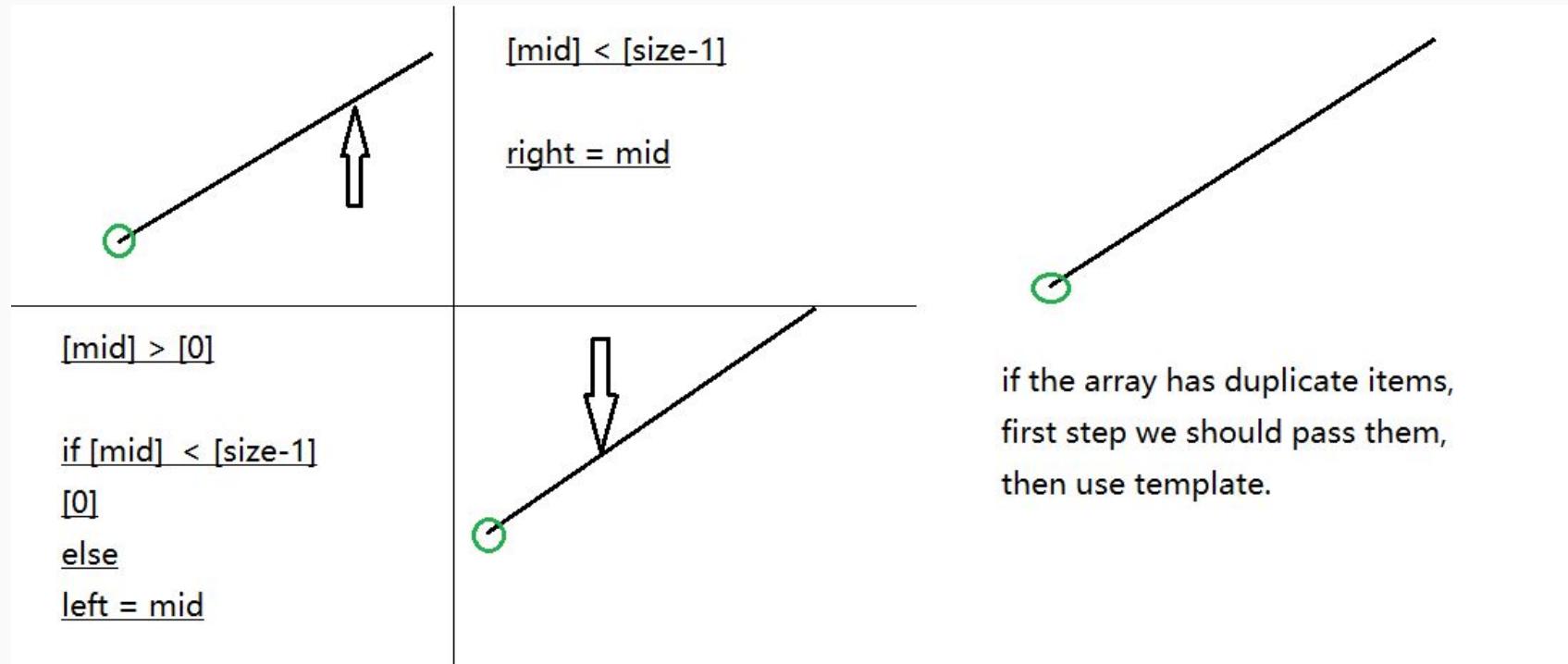
magiciendecode.fr

<https://magiciendecode.fr>

153. Find Minimum in Rotated Sorted Array & 154 II

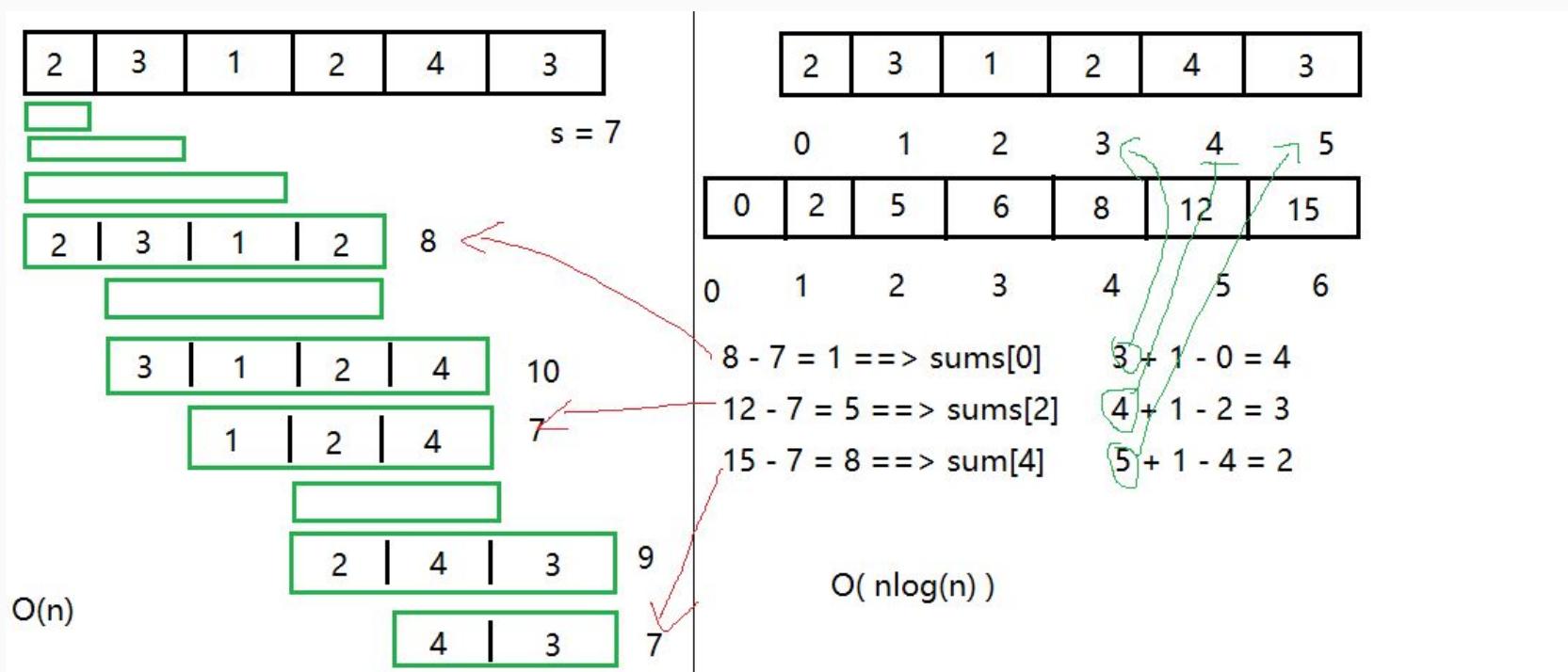
<https://leetcode.com/problems/find-minimum-in-rotated-sorted-array/>

<https://leetcode.com/problems/find-minimum-in-rotated-sorted-array-ii/>



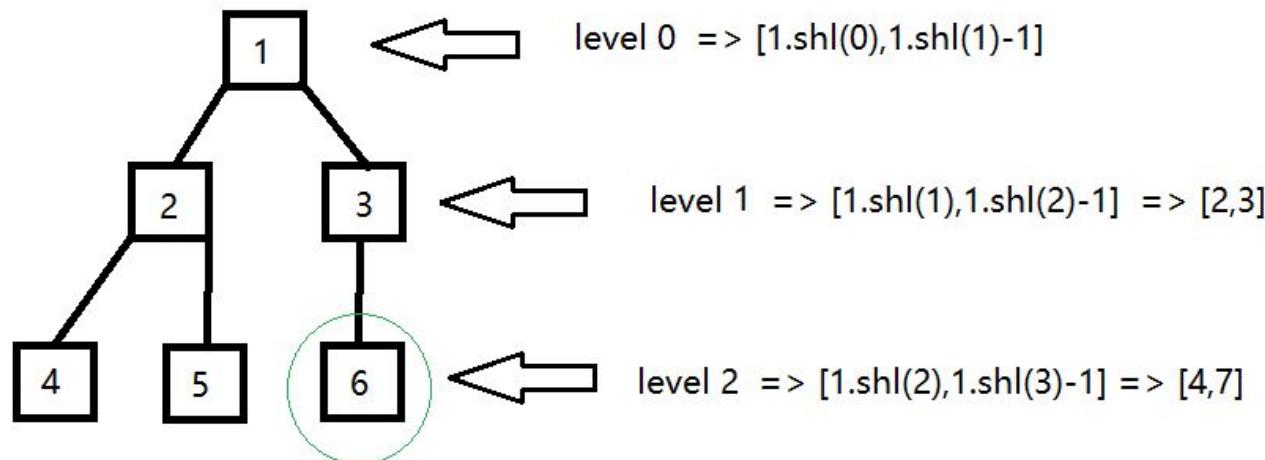
209. Minimum Size Subarray Sum

<https://leetcode.com/problems/minimum-size-subarray-sum/>



222. Count Complete Tree Nodes

<https://leetcode.com/problems/count-complete-tree-nodes/>



$$\text{left} = 1.\text{shl}(2) = 4$$

$$\text{right} = 1.\text{shl}(3) - 1 = 7$$

$$\text{mid} = 11 / 2 = 5$$

$$\text{target} = 6$$

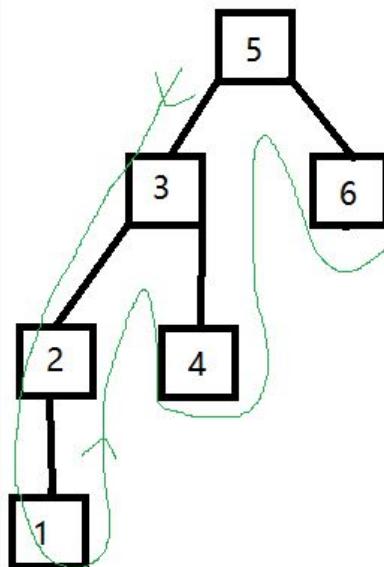
target > mid ==> RightNode

else ==> LeftNode

Binary search in the last level of tree.
verify the last present node.

230. Kth Smallest Element in a BST

<https://leetcode.com/problems/kth-smallest-element-in-a-bst/>



iterate in order: [1, 2, 3, 4, 5, 6]

S(n) O(n)

search the most left node and then add it in result array.

same idea, iterate in order but return the value if it is the kth smallest element.

S(1)
O(n)

count the sum of left nodes then add itself .

O(nlog(n))

if $k == \text{total}$, return current

if $k > \text{total}$, it means kth element in the right side of current node

else $k < \text{total}$, result in left side.

240. Search a 2D Matrix II

<https://leetcode.com/problems/search-a-2d-matrix-ii/>

```
matrix = [  
    [1, 7, 15],  
    [3, 9, 22],  
    [18, 23, 30]  
]
```

from the bottom left, verify the current and target
if current > target ==> move to previous line
if current < target ==> move to next column

$O(m+n)$

Kotlin

fold : Accumulates value starting with [initial] value and applying [operation] from left to right to current accumulator value and each element.

reduce : Accumulates value starting with the first element and applying [operation] from left to right to current acc and each element.

274. H-Index && 275. H-Index II

<https://leetcode.com/problems/h-index/>

<https://leetcode.com/problems/h-index-ii/>

[3, 0, 6, 1, 5]

for each [0 ... size]: binary search $O(\log(n))$
verify if it is a valid target.

从0到数组的长度(包括) , 找到一个值target满足
在这个数组中比target大的至少有target个。

$O(n \log(n))$

validate target: iterate array, count how many
element \geq target, if total \geq target means it's n
a valid h-index.

[0, 1, 3, 5, 6]

validate target use binary search $\Rightarrow O(\log(n))$

$O(\log(n))$

as the citations are in ascending order,
citations[length - mid] means at least mid times, then verify it \geq mid?
if yes, left = mid and find the largest h-index, if not right = mid

287. Find the Duplicate Number

<https://leetcode.com/problems/find-the-duplicate-number/>

[1, 3, 4, 2, 2]
[3, 1, 3, 4, 2]

$O(n^2)$: foreach element in array: iterate array find if there exists another element equals to itself.

$O(n \log n)$: binary search from 0 to size-1,

if the number of element ($\leq mid$) $\leq mid$, means no duplicate $\leq mid$, so left = mid
else right = mid.

[1, 2, 2]

$O(n)$: two pointers: fast pointer and slow pointer will meet two times, the second time is the duplicate value. After first meet, fast pointer will go to the begining and move as slow pointer.

300. Longest Increasing Subsequence

<https://leetcode.com/problems/longest-increasing-subsequence/>

[10, 9, 2, 5, 3, 7, 101, 18] 2 -> 5 -> 7 -> 101 2 -> 3 -> 7 -> 101 2 -> 3 -> 7 -> 18

[10, 9, 2, 5, 3, 4] 2 → 5 2 -> 3 -> 4



this is an array to record longest increasing subsequence.

maxLIS

foreach element in original array, try to find the first element \geq current,
then replace it.

10

9

2

2 -> 5

2 -> 3

2 -> 3 -> 4

if we use two foreach nested, $O(n^2)$

if we use foreach and binray search to find the first element in maxLIS

$O(n \log n)$

378. Kth Smallest Element in a Sorted Matrix

<https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix/>

```
matrix = [           create a descending PriorityQueue.  
    [ 1,  5,   9 ]  
    [ 10, 11,  13 ]  
    [ 12, 13,  15 ]  
]           foreach element in matrix, add it in queue, if size of queue > k, poll an element.  
           as it's in descending, this queue always keeps k element, so at the end, poll the result  
           that is kth smallest element.  
           O( n*n*log(k) )  
  
[           Binary search ==> O( log(max-min)*n*log(n) )  
    [ 1, 2 ]           left = matrix[0][0] right = matrix[max][max]  
    [ 1, 3 ]           every time count the element that <= mid, if the count < k means kth smallest is  
]           between mid and right, so left = mid  
  
           how to count? foreach line, binary search the last element <= target
```

436. Find Right Interval

<https://leetcode.com/problems/find-right-interval/>

```
[  
    [ 3,  4 ]           foreach line in matrix: find  
    [ 2,  3 ]           foreach line in matrix: compare  
    [ 1,  2 ]           O ( n*n )  
]  
]
```

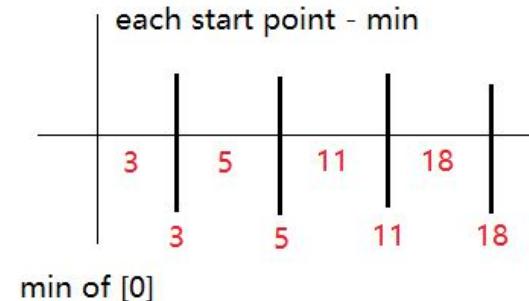
对于当前矩阵的每一行 I，找到是否存在其他行 J，J的第一个元素[0]大于等于当前行I的最后一个元素[size-1]，找到差值最小的那一行 J。
对于每一行，找到则返回J在矩阵中的下标，否则-1。

[3,4] doesn't exist one row that $[0] \geq 4$, so return -1

[2,3] ==> index 0 [3,4] $3 \geq 3$, so return 0

[1,2] ==> index 1 [2,3] $2 \geq 2$, so return 1

The result of example is [-1,0,1]



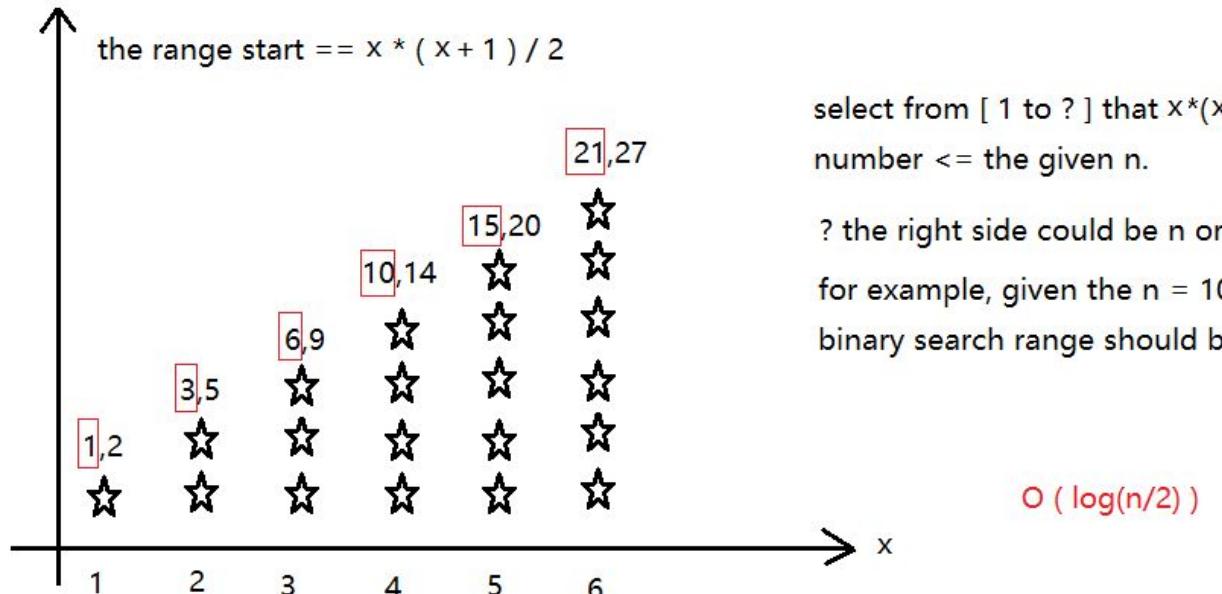
Then just get $[size-1] - min$

$O(3n + max - min)$

$S(max-min)$

441. Arranging Coins

<https://leetcode.com/problems/arranging-coins/>



454. 4Sum II

<https://leetcode.com/problems/4sum-ii/>

- | | |
|---|---|
| <p>1. use cartesian product of A,B,C,D that will make time complexity $n^*n^*n^*n$</p> <p>2. sort D, use cartesian product of A,B,C, search the start and end of target in D.
$n*\log(n) + n^*n^*n^*2\log(n)$</p> <p>3. store the cartesian product of C,D , then sort it. $n^*n + n*\log(n)$
foreach element in cartesian product of A,B, find the start and end of target in sortedCD. $n^*n^*2\log(n^*n)$</p> <p>4. use hashMap but this will take more memory. $n^*n + n^*n$</p> | <p>cartesian product of A,B,C,D
each array has two elements</p> <p>A[0] + B[0] + C[0] + D[0]
A[0] + B[0] + C[0] + D[1]
A[0] + B[0] + C[1] + D[0]
A[0] + B[0] + C[1] + D[1]
A[0] + B[1] + C[0] + D[0]
A[0] + B[1] + C[0] + D[1]
A[0] + B[1] + C[1] + D[0]
A[0] + B[1] + C[1] + D[1]
A[1] + ...</p> |
|---|---|

475. Heaters

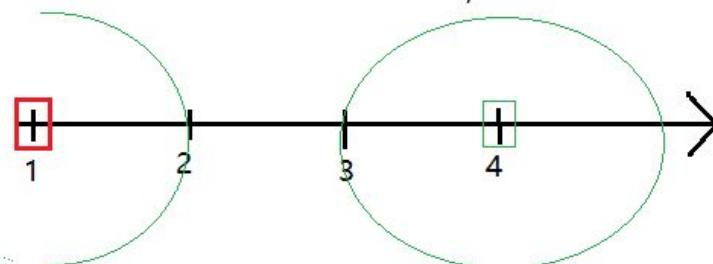
<https://leetcode.com/problems/heaters/>

array A [1, 2, 3, 4]

array B [1, 4]

foreach element in array A find an element in B that make the minimum difference value between them.

Then for all these minimum differences, find the maximum.



binary search in array B

if B empty, return 0

if house < [0] , return [0] - house

if house > [max], return house - [max]

[mid] = house means distance 0

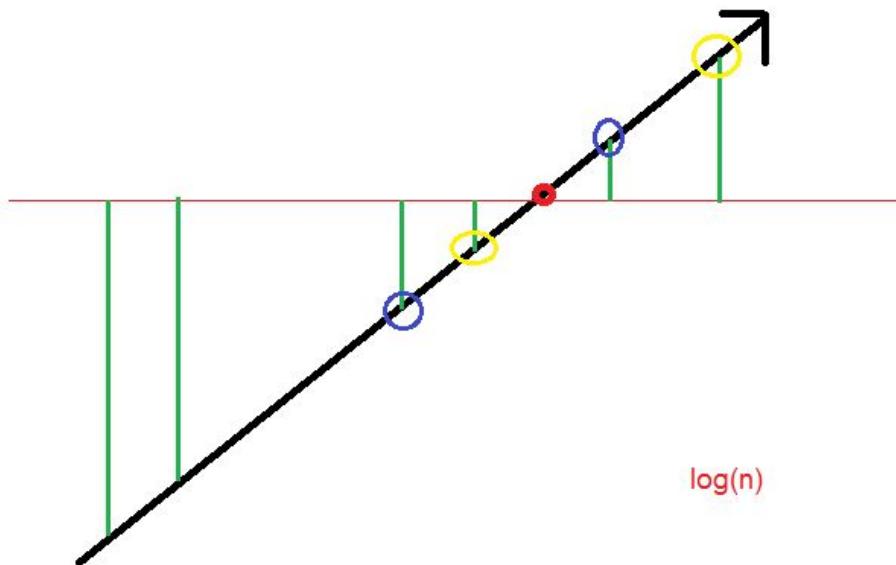
at the end, we will find

[left] < house < [right]

return minOf house - [left] and [right] - house

658. Find K Closest Elements

<https://leetcode.com/problems/find-k-closest-elements/>



binary search in array an element that most close to target, then base on the index, take index - 1 or index + 1 according to the distance between the value and target.

$\log(n)*k$

binary search in array the start index

if $\text{target} - [\text{mid}] > [\text{mid}+k-1] - \text{target}$, $\text{left} = \text{mid}$
else $\text{right} = \text{mid}$

we use our template, at the end we find left,right
take an example, the blue point and yellow point.

just compare the distance between $\text{target} - [\text{left}]$ and $[\text{right}+k-1] - \text{target}$

718. Maximum Length of Repeated Subarray

<https://leetcode.com/problems/maximum-length-of-repeated-subarray/>

A [1, 2, 3, 2, 1]
B [4, 3, 2, 1, 7]

```
foreach in array A:  
    foreach in array B:  
        if(a==b)  
            while  
                count if next elements equals
```

```
foreach in A:  
    1, 2, 3, 2, 1  
    4, 3, 2, 1, 7 ==> 0  
    2, 3, 2, 1  
    4, 3, 2, 1, 7 ==> 3  
    3, 2, 1  
    4, 3, 2, 1, 7 ==> 0  
    3, 2, 1  
    4, 3, 2, 1, 7 ==> 0  
    4, 3, 2, 1, 7 ==> 0
```

foreach in B do the same things with A

$n * n * n$

$2 * n * n$

dynamic programming, using a memory-based data structure to avoid repeated computing.

$S(n * n)$
 $O(n * n)$

binary search in [1,1000]
but how to verify mid is a valid length?

convert A and B to string,
first you should map it to char, why?
we want to use substring and hashSet
to verify if exists a length of string that
make a subarray.

think about 12,4,3, if we use int, 1243

	1	2	3	2	1
4	0	0	0	0	0
3	0	0	1	0	0
2	0	1	0	2	0
1	1	0	0	0	3
7	0	0	0	0	0

$O((2n) * \log 1000)$

then we should create a hashSet, then put
substrng(index, index + targetLength) of A in this hashSet, $O(n)$
foreach substrng(index, index + targetLength) of B, verify if hashSet
contains it. $O(n)$

875. Koko Eating Bananas

<https://leetcode.com/problems/koko-eating-bananas/>

1 1000000000

$O(n * \log(1000\ 000\ 000))$

binary search to find the first valid target
XXXXXXXXXXOooooooooo

A valid target means
foreach element in array
calculate $i / \text{target} + (1 \text{ or } 0, \text{ according to } i \% \text{target})$
then sum the result, compare with H, if result $\leq H$, it's a valid target.

```
int left = (int)(sum / H)
left = left == 0 ? 1 : left
int right = (int)(sum / (H - piles.length + 1)) + 1
```

missing testcase:

$1 \leq \text{piles.length} \leq 10\ 000$
 $\text{piles.length} \leq H \leq 1\ 000\ 000\ 000$
 $1 \leq \text{piles}[i] \leq 1\ 000\ 000\ 000$

wrong code:

```
int left = (int)(sum / H) + 1
```

Input:
piles = [2,2]
H = 2

expected k = 2

but answer is 3

528. Random Pick with Weight

<https://leetcode.com/problems/random-pick-with-weight/>

weight [1, 3, 6]

it means if you select from index 0,1,2
that you have 10% fall on index 0,
30% fall on index 1
60% fall on index 2

This question can be transformed as

given an array and a target in [1,array.max]
return the first index that array[index] >= target

1. calculate sum [1,4,10]

XXXXXXXXOoooooooo

1 2 3 4 5 6 7 8 9 10

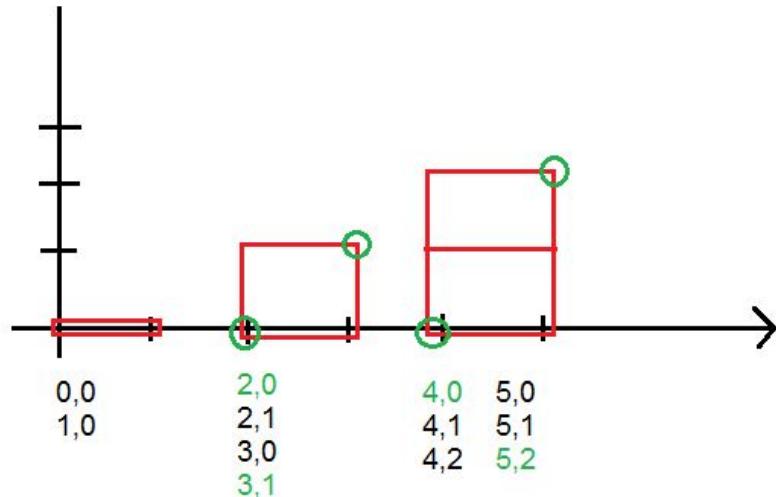
in kotlin

init{...} will be executed after the
main constructor

if there are several init bloc, they
will be executed in order.

497. Random Point in Non-overlapping Rectangles

<https://leetcode.com/problems/random-point-in-non-overlapping-rectangles/>



[2,4,6]

weight [1, 3, 6]

it means if you select from index 0,1,2
that you have 10% fall on index 0,
30% fall on index 1
60% fall on index 2

given an array and a target in [1,array.max]
return the first index that array[index] >= target

XXXXXXO OOOOOOO

if we find know the point should at which rectangle,
then randomly return a [x,y] in it.
for example the rectangle[4,0,5,2]
the x of point could be 4 + random.nextInt(5-4+1)
the y of point could be 0 + random.nextInt(2-0+1)

911. Online Election

<https://leetcode.com/problems/online-election/>

index	0	1	2	3
persons	[3	0	3	1]
times	[5	12	39	100]

leaders[persons[0], persons[1], persons[0], persons[0]]	times[0]	times[1]	times[2]	times[3]
---	----------	----------	----------	----------

The question is given a time (for example 7)
in [times[0], times[max]] ==> [5,100]
return the person that at time 7 get the most vote.

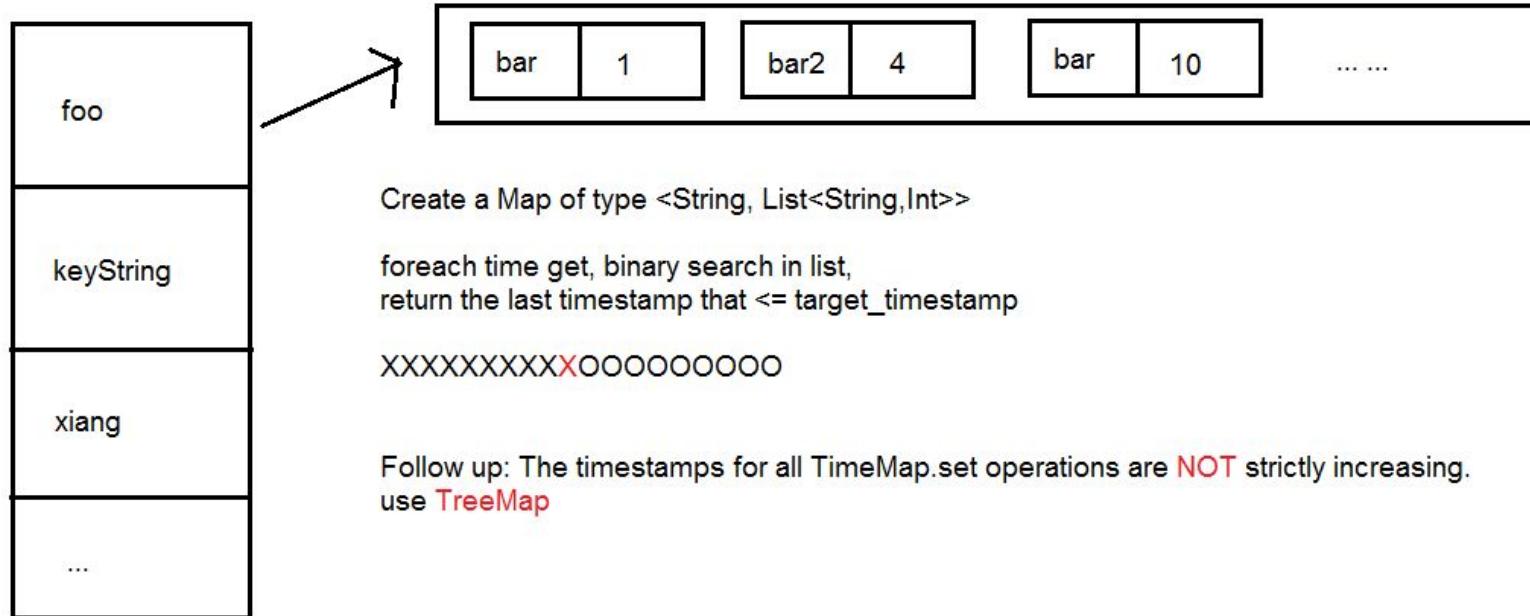
counts int[] max length is persons.length as $0 \leq i \leq persons.length$
// at index i, the value means person[i]'s count

leaders int[] max length is times.length **S(n)**
// at index i, time[i]'s leader

given a random time, use binary search return the last index that $time[index] \leq target$ **10 000 * log(n)**
then return leaders[index] which represent the leader at time[index]

981. Time Based Key-Value Store

<https://leetcode.com/problems/time-based-key-value-store/>



1011. Capacity To Ship Packages Within D Days

<https://leetcode.com/problems/capacity-to-ship-packages-within-d-days/>

binary search, first we should set left and right,

let's search the max weight in array, then set left = max_weight

the right won't be exceed max_weight * max_weights_length, so right = max_weight * 500 ~~50000~~

Then this question transform to select the first valid capacity from [max_weight, ~~500~~ * max_weight]

XXXXXXXXX~~L~~LLLLLLLLL

days = 1, count = target

```
foreach w in weights
    count = count - w
    if( count < 0 ){
        days++
        count = target - weight
    }
```

weights [3, 2, 2, 4, 1, 4] D = 3, target = 5

day 1 [3,2]	count = 0
day 2 [2]	0-2 < 0 ==> count = 5 - 2 = 3
day 3 [4]	3 - 4 < 0 ==> count = 5 - 4 = 1
[1]	1 - 1 = 0
day 4 [4]	

1300. Sum of Mutated Array Closest to Target

<https://leetcode.com/problems/sum-of-mutated-array-closest-to-target/>

```
sum = 0 [ 4, 9, 3 ] , target = 30  
foreach element in array:  
    sum += minOf(element,?) [ 4, 9, 3 ] ==> 16  
[ 4, 9, 3 ] , target = 10 [ 2, 2, 2 ] ==> 6  
[ 3, 3, 3 ] ==> 9 The maximum available value is target  
[ 2, 3, 5 ] , target = 10 [ 0, 0, 0 ] ==> 0  
[ 2, 3, 5 ] ==> 10 The minimum available value is 0
```

```
Binary search from 0 to target  
if sum == target, return it  
if sum < target  
    left = mid  
if sum > target  
    right = mid
```

$O(n * \log(\text{target}))$

1201. Ugly Number III

<https://leetcode.com/problems/ugly-number-iii/>

left = minOfabc
right = minOfabc * n

binary search : calculate the mid is ?th ugly number

mid / a + mid / b + mid / c
- mid / lcmab - mid / lcmac - mid / lcmbc
+ mid / lcmabc

n = 1000000000
a = 2
b = 217983653
c = 336916467

1999999983 ==> 999999999
1999999984 ==> 10000000000
1999999985 ==> 10000000000

LCM, Least Common Multiple 最小公倍数

a * b / gcd(a,b)

GCD Greatest Common Divisor 最大公约数

辗转相除法

given a,b each time ensure a > b, then if a%b == 0 return b
else gcd(b,remainder)

319,377

1 ==> 377 % 319 ... 58
2 ==> 319 % 58 ... 29
3 ==> 58 % 29 ... 0
return 29

更相减损法

63,98
98 - 63 = 35
63 - 35 = 28
35 - 28 = 7
28 - 7 = 21
21 - 7 = 14
14 - 7 = 7
return 7

1283. Find the Smallest Divisor Given a Threshold

<https://leetcode.com/problems/find-the-smallest-divisor-given-a-threshold/>

[10] , threshold = 11

left = 1

[2, 20] , threshold = 1

right = maxOfArray

sum = 0

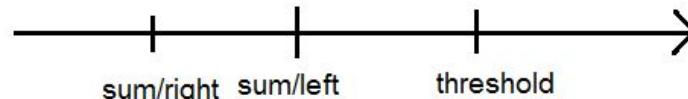
foreach element in array:

 sum += element / target + (1 or 0)

sum == threshold -> return mid ?

sum > threshold -> left = mid

else -> right = mid



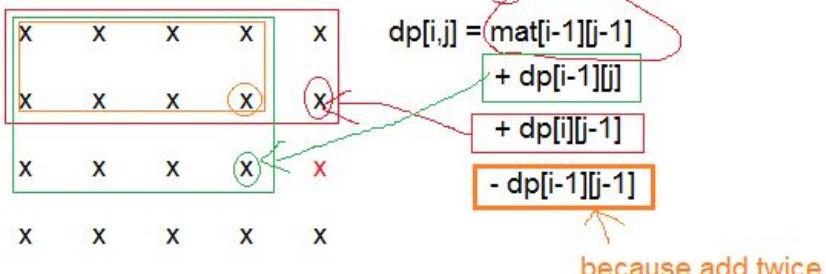
$$\text{sum} / \text{target} \longleftrightarrow \text{threshold}$$

$$\begin{matrix} \text{target} \\ \text{sum} / \text{target} \end{matrix}$$

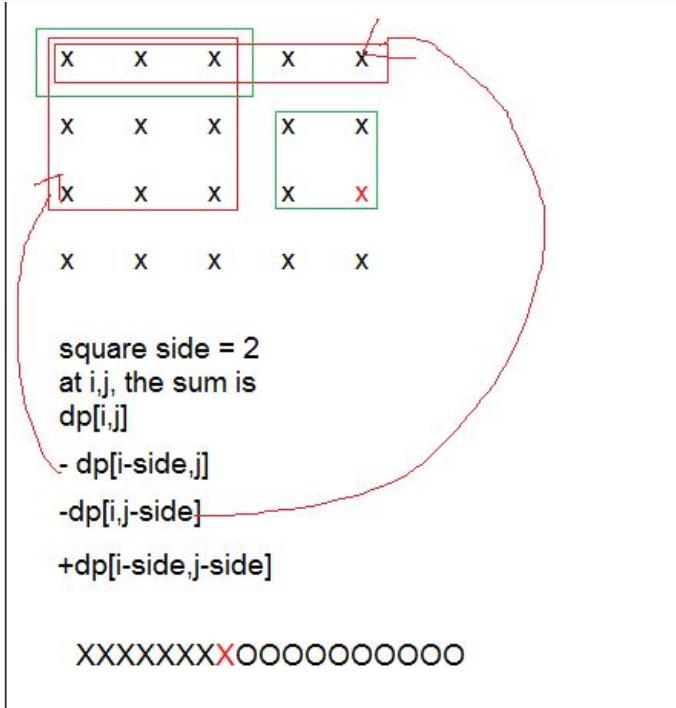
1292. Maximum Side Length of a Square with Sum Less than or Equal to Threshold

<https://leetcode.com/problems/maximum-side-length-of-a-square-with-sum-less-than-or-equal-to-threshold/>

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
=====dp=====			
0	0	0	0
0	1	2	3
0	2	4	6
0	3	6	9
0	4	8	12
0	4	8	12

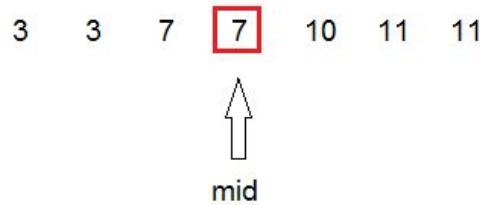


in array dp, [i,j] calculate the sum of area
0,0 to i,j
note that it record the position in original array[i-1,j-1]



540. Single Element in a Sorted Array

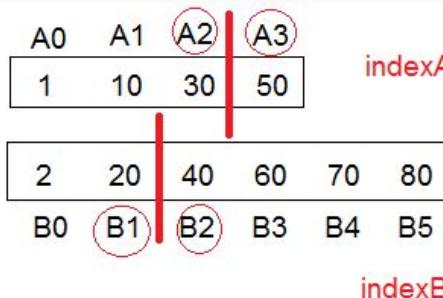
<https://leetcode.com/problems/single-element-in-a-sorted-array/>



1. single element
previous != single element != next
 2. single element in which part
not single element
- if single element is in the right part, left = mid
1. second not single element && index % 2 != 0
 2. first not single element && index % 2 == 0

4. Median of Two Sorted Arrays

<https://leetcode.com/problems/median-of-two-sorted-arrays/>

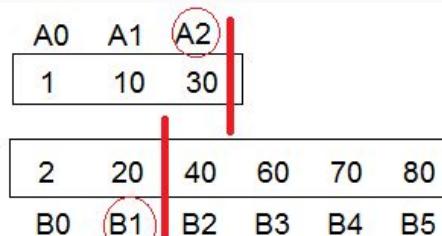


maxOf(A2, B1) + minOf(A3, B2)

2.0

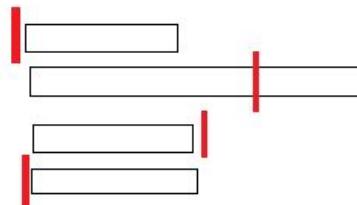
maxOf(left_part_AB) + minOf(right_part_AB)

2.0



maxOf(A2 , B1)

maxOf(left_part_AB)



left_part_AB = right_part_AB
max(left_part_AB) <= min(right_part_AB)

indexB = $(A+B+1) / 2 - \text{indexA}$

if midA < the correct indexA, left = mid

indexB >= 1 && B[indexB-1] > A[midA]

if midA > the correct indexA, right = mid
midA >= 1 && A[indexA-1] > B[indexB]

174. Dungeon Game

<https://leetcode.com/problems/dungeon-game/>

dynamic programming

-2 (K)	-3	3
-5	-10	1
10	30	-5(P)

start from (P), we need K is alive HP = 1
so K should have minimum $1 - (-5) = 6$
before enter this room(3,3)

Then calculate each room: the value before enter is $dp[i,j] = 1$ or ($\min(\min(dp[i+1,j], dp[i,j+1]) - dungeon[i][j])$)

the minimum HP for next step

-
current value

if the result is negative means current room can add HP
K just need to be alive HP = 1

7 (K)	5	2
6	11	5
1	1	6 (P)

max max 1

$S(m)$

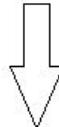
$O(m^n)$

how to binary search ?

315. Count of Smaller Numbers After Self

<https://leetcode.com/problems/count-of-smaller-numbers-after-self/>

5 2 6 1



5 2 1 6



5 1 2 6



1 2 5 6

1

nums[3] = 1, in new sorted array, index 0

1 6

nums[2] = 6, in new sorted array, index 1

1 2 6

nums[1] = 2, in new sorted array, index 1

1 2 5 6

nums[0] = 5, in new sorted array, index 2

S(n)
O(n * log(n))

binary search the insert position

[mid] <= target ----- left = mid

[mid] > target ----- right = mid

[mid] < target ----- left = mid

[mid] >= target ----- right = mid

target <= [left] ----- return left

target <= [right] ----- return right

else -> right + 1

2 3 3 3 9

327. Count of Range Sum

<https://leetcode.com/problems/count-of-range-sum/>

-2	5	-1
----	---	----

nums

0	-2	3	2
---	----	---	---

sums

sums[i+1] = sums[i] + nums[i]

0 1 2 3

sum of array from [i to j] = sums[j+1] - sum[i]

[0 to 0] = sum[1] - sum[0] = -2

[1 to 2] = sum[3] - sum[1] = 2 - (-2) = 4 <--> 5 + (-1)

for i in 0 .. nums.size

 for j in i+1 .. sums.size
 sums[j] - sum[i] ?

S (n)

O (n*n)

iterate all ranges

without loop calculate sum

!! LongArray !!

lower <= sums[j] - sums[i] <= upper sums[j] fix, search range sums[i]
when loop to calculate sums[j]

if sums could be sorted, then find min and max index for sums[i]

count = max - min

sums[i] >= sums[j] - upper <--> binary search first index

sums[i] <= sums[j] - lower <--> binary search last index

in order to keep sorted sums, binary search insert position

S (n)

O (n * log(n))

lower <= sums[j] - sums[i] <= upper sums[i] fix, search range sums[j]

given the sums[i], let's see sums[j]

sums[j] <= sums[i] + upper

sums[j] >= sums[i] + lower

Merge Sort, when merge, find the range of j

S (n)

O (n * log(n))

352. Data Stream as Disjoint Intervals

<https://leetcode.com/problems/data-stream-as-disjoint-intervals/>

1,3,7,2,6

[1,1]

[1,1],[3,3]

[1,1],[3,3],[7,7]

[1,1],[2,2],[3,3],[7,7] ==> [1,3],[7,7]

[1,3],[6,6],[7,7] ==> [1,3],[6,7]

0	1	key	value	0	1
		val	val		

leftInterval rightInterval

[1] >= val -> return

[1] +1 == val -> key = [0], remove leftInterval

rightInterval

[0] -1 == val, value = [1], remove rightInterval

treeMap.put(key,value)

```
size = 0 -> add(new)           MutableList<IntArray>
size = 1 -> {
    val +1 < first[0] -> add(0,new) / val + 1 == first[0] -> first[0] = val
    first[1] + 1 == val -> first[1] = val / first[1] + 1 < val -> add(1,new)
}
size > 1 -> {
    targetPosition == binarySearchLastPositionLessThanTarget
    targetPostion == 0 && val < [0] ->{
        val +1 < [0] -> add(0,new) / val + 1 == [0] -> [0] = val
    }
    targetPosition == size -1 ->{
        [1] + 1 = val -> [1] = val / [1] + 1 < val -> add(size,new)
    }
    1<=targetPosition<=size-2 ->{
        val -1 > left[1] && val +1 < right[0] -> add(targetP+1,new)
        val -1 == left[1] && val +1 < right[0] -> left[1] = val
        val -1 > left[1] && val +1 == right[0] -> right[0] = val
        val -1 == left[1] && val +1 == right[0] ->
            left[1] = right[1], removeAt(targetP+1)
    }
}
```

354. Russian Doll Envelopes

<https://leetcode.com/problems/russian-doll-envelopes/>

< 5 , 4 >	2	5	6	6	w
< 6 , 4 >	3	4	7	4	h
< 6 , 7 >					
< 2 , 3 >					

Step1: sort by w, then sort by descending h

Step2: search longest increasing subsequence

<https://leetcode.com/problems/longest-increasing-subsequence/>

Java :

https://github.com/MagicienDeCode/LeetCode_Solution/blob/master/src/main/java/binary_search/LongestIncreasingSubsequence300.java

Kotlin :

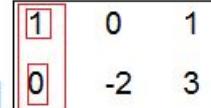
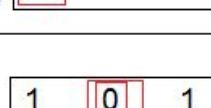
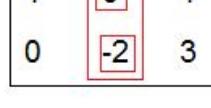
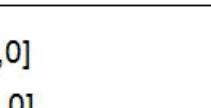
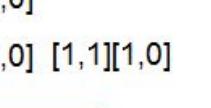
https://github.com/MagicienDeCode/LeetCode_Solution/blob/master/src/main/kotlin/binary_search/LongestIncreasingSubsequenceKotlin300.kt

363. Max Sum of Rectangle No Larger Than K

<https://leetcode.com/problems/max-sum-of-rectangle-no-larger-than-k/>

	0	1	2	col
0	1	0	1	for (col in 0 1 2)
				rowSum[0]
1	0	-2	3	rowSum[1]
row				for (c in col to 0)
				sortedSum.add 0
				currentSum = 0
				for (r in 0 1)
				rowSum[r] += matrix[r][c]
				currentSum += rowSum[r]
0				verify currentSum - k in sortedSum
1				result = maxOf(currentSum - ?)
rowSum				sortedSum.add (currentSum)

binary search the first element that greater equal than target

col =0	c =0	r =0	[0,0]	
		r =1	[1,0]	
			[1,0][0,0]	
col =1	c =1	r =0	[0,1]	
		r =1	[1,1]	
			[0,1][1,1]	
			[0,1][111]	
	c =0	r =0	[0,1][0,0]	
		r =1	[1,1][1,0]	
			[0,1][0,0] [1,1][1,0]	
			[0,1][0,0] [1,1][1,0]	

410. Split Array Largest Sum

<https://leetcode.com/problems/split-array-largest-sum/>

[7,2,5,10,8]

when m = 1, sum = array.sum()
when m = array.size, sum = max

ooooooooooooXXXXXX

count = 1
a valid sum is

```
foreach i in array
    sum += i
    if(sum>target)
        sum = i
    count ++
    verify count > aim?
```

						i = 2
	7	2	5	10	8	
1	7	9	14	24	32	7 2 5
2	7	7	7	14	18	7 7 9 5 minof = 7
						7 2 5 10
						7 17 9 15 minof = 14
						14 10
						7 2 5 10 8
						7 25 9 23 14 18 minof = 18
						24 8

483. Smallest Good Base

<https://leetcode.com/problems/smallest-good-base/>

$$13 = 3(0) * 1 + 3(1) * 1 + 3(2) * 1$$

check good base

current = 1

current = $1 * 3 + 1$

current = $4 * 3 + 1$

foreach base from 2 to n

if(good base)

return

$n \log(\text{base})(n) \Rightarrow$ Time Limit Exceeded

base = n-1
minK = 2

base = 2
maxK = $\log_2(n) + 1$

from maxK downTo minK
binarySearch valid base

baseLeft = 2
baseRight = ? n

$\log_2(n) * \log(\text{base})(n) * \log(\text{base})(n)$

K larger
base smaller

given K and base and n

if $\text{base}(0) + \dots + \text{base}(K) == n$ OK
< n -> left = mid
> n -> right = mid

attention Long overflow

493. Reverse Pairs

<https://leetcode.com/problems/reverse-pairs/>

[1, 3, 2, 3, 1]

i = 1 [i] = 3
j = 4 [j] = 1

i = 3 [i] = 3
j = 4 [j] = 1

for i in 0 .. size -2
 for j in i+1 .. size -1
 verify!!

Time Limit Exceeded

n * n

sortedList of j



for i in size-1 .. 0
 binarySearch
 then add

target > [j].toLong * 2

right ?
left ?
-1

2 * n * log(n)

Merge Sort

it will guaranteed the i < j, then just binary search [i] > 2*j

n * log(n)

668. Kth Smallest Number in Multiplication Table

<https://leetcode.com/problems/kth-smallest-number-in-multiplication-table/>

1	2	3	binary search from [1 to $m \times n$]
2	4	6	count of element less equal than target
3	6	9	< k ==> left = mid >= k ==> right = mid

foreach line

binary search lessEqual than target position

$\log(m \times n) * m * \log(n)$

foreach line

count +=
max <= target ? n
else target / line

$\log(n \times m) * m$

https://github.com/MagicienDeCode/LeetCode_Solution

Ctrl + F ==> 378. Kth Smallest Element in a Sorted Matrix

<https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix/>

719. Find K-th Smallest Pair Distance

<https://leetcode.com/problems/find-k-th-smallest-pair-distance/>

k

priority queue with desceding order



```
for i in 0..size-2  
    for j in i+1 .. size -1  
        distance = abs(nums[i]-nums[j])
```

Memory Limit Exceeded

1. sort

2. binary search from 0 to max-min

3. how to calculate how many distances <= target

1 2 4 10 target = 4

1 2 [0,1]

1 4

2 4 [0,2]

778. Swim in Rising Water

<https://leetcode.com/problems/swim-in-rising-water/>

0	1	2	3	4
24	23	22	21	5
12	13	14	15	16
11	17	18	19	20
10	9	8	7	6

BFS + PriorityQueue

1 (24)
2 (23 24)
3 (22 23 24)
4 (21 22 23 24)
5 (21 22 23 24)
16 (21 22 23 24)
...
...

binary search from min to max in matrix
for{for{}}

We need a matrix visited and we should clean this matrix each time before we calculate the mid.

how to verify if a given target is valid or not? DFS

```
if (inGrid(x, y, grid.size, grid[0].size)
    && visited[x][y] == 0 &&
    grid[x][y] <= target
) {
    visited[x][y] = 1
    dfs(grid, visited, x - 1, y, target)
    dfs(grid, visited, x, y - 1, target)
    dfs(grid, visited, x + 1, y, target)
    dfs(grid, visited, x, y + 1, target)
}
```

786. K-th Smallest Prime Fraction

<https://leetcode.com/problems/k-th-smallest-prime-fraction/>

priority_queue



for i
 for j
 i/j ==> priority_queue

if > K
priority_queue.poll()

Time Limit Exceeded

1 2 3 5 q = 1

binary search

while(true) 0.5 (0.75 , 0.25)

0.5

1/3 1/5

2/5

p = 0

i

j = i + 1

A[i] / A[j] < target

if p/q < A[i] / A[j]
update p = A[i], q = A[j]

p = 0

j = 1

i

A[i] / A[j] < target

if p/q < A[i] / A[j]
update p = A[i], q = A[j]

862. Shortest Subarray with Sum at Least K

<https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k/>

A	56	-21	56	35	-9	
sumsA	0	56	35	91	126	117
	0	1	2	3	4	5

sums[end] - sums[start] $\geq K$

(0,0)

sums[end] - K \geq sums[start]

(0,0) (56,1) $\rightarrow 56-61 = -5$

(0,0)(56,1) (35,2) $\rightarrow 35-61 = -26$

(sums,index)

(0,0)(35,2)(56,1) (91,3) $\rightarrow 91 - 61 = 30$

3

sortedList: List<Pair<Int,Int>>

(35,2)(56,1)(91,3) (126,4) $\rightarrow 126-61=65$

2

binary search the

last position that \leq (sums[end]-K)

(0,0) 56-0 $\geq K$

(0,0)(56,1) 35

foreach 0 .. position

 result = minOf(
 result,
 nowIndex - second
)

index_current > index_last

[current] < [last]

? - [current] $\geq K$

(0,0)(35,2)

remove 0 until position

why we can't binary search from 1 to A.size

how can we say a length is valid?

sums[i-length] - sums[i] $\geq K$

if valid => right = mid
else -> left = mid

imagine that length = 1 is valid ..
eumm

710. Random Pick with Blacklist

<https://leetcode.com/problems/random-pick-with-blacklist/>

0 1 2 3 4 5 6 7

hash[0] -> 5
hash[1] -> 2

2 5
0 1 3 4 6 7 [0,5]

first element make random from $[0, N - B.size] + index < B[index]$

0 1 2 3 4 5 6 7

hash[1] -> 2

0 -> 0 1 2 3 4 5 6 7
1 -> 1 2 3 4 5 6 7
B 0 1 3 4 6 7

-> 0 + 2 < 3
-> 1 + 4 < 6

random + mid < B[mid]

-> right = mid

else -> left = mid

0 1 0
2 1 2

0 2 1
1 0 2

left/right + random
else -> random + right + 1

793. Preimage Size of Factorial Zeroes Function

<https://leetcode.com/problems/preimage-size-of-factorial-zeroes-function/>

x = 1 ~ 4	K = 0		binary search from 0 to $(K+1) * 5$
x = 5 ~ 9	K = 1		
x = 10 ~ 14	K = 2		
x = 15 ~ 19	K = 3	15 / 5 = 3	if find a number that $f(\text{number}) = K$, return 5
x = 20 ~ 24	K = 4	16 / 5 = 3	else -> 0
x = 25 ~ 29	K = 6	26 / 5 = 5 .. 1	
		5 / 5 = 1 .. 0	

K = 5 result = 0

```
private fun numberOfZero(long: Long): Int {  
    var count = 0L  
    var current = long  
    while (current != 0L) {  
        count += current / 5  
        current /= 5  
    }  
    return count.toInt()  
}
```

878. Nth Magical Number

<https://leetcode.com/problems/nth-magical-number/>

$$\begin{array}{l} 12 / 2 = 6 \\ 12 / 3 = 4 \end{array} \quad \begin{array}{cccccc} 2 & 4 & 6 & 8 & 10 & 12 \\ 3 & \textcolor{red}{6} & 9 & \textcolor{red}{12} \end{array}$$

12 is N-th magical number for 2 and 3. 8

$$12 / 6 = 2 \quad 6 \quad 12$$

$$6 + 4 - 2 = 8$$

```
// LCM, Least Common Multiple          // GCD, Greatest Common Divisor
private long lcm(long a, long b) {        private long gcd(long a, long b) {
    return a * b / gcd(a, b);            if (a < b) {
}                                         a = a ^ b;
}                                         b = a ^ b;
                                         a = a ^ b;
}                                         }
                                         while (a % b != 0) {
                                         final long remainder = a % b;
                                         if (b < remainder) {
                                         a = remainder;
                                         } else {
                                         a = b;
                                         b = remainder;
                                         }
                                         }
                                         return b;
}
```

1044. Longest Duplicate Substring

<https://leetcode.com/problems/longest-duplicate-substring/>

a b a b a

b a b a

hashcode = b * 31(3) + a * 31(2) + b * 31(1) + a * 31(0) = long % (1000000007)

a b a = (b a b a - b * 31(3)) % (1000000007)

if(hashcode<0) ? + 1000000007

31*(target)
hash_map

currentCode

for index in string.indices
 if(index<target-1) continue
 if(index>=target) - if<0

double check

hashmap[currentCode] = index - target + 1

binar search from 1 to size -1

if(right)
 substring(right_result,right_result + right)

1044. Longest Duplicate Substring

<https://leetcode.com/problems/longest-duplicate-substring/>

2 3 1 4

1 2 3 4

1 1 2 6

K = 9

ck = 8

8/6 = 1

ck = ck % 6 = 2

2/2 = 1

ck = ck % 2 = 0

a,b,a,b,a

Trie (index = 0, depth = 0)

depth = 0

if(index+depth == S)

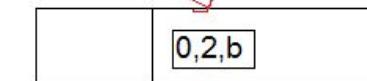


getIndex(0+0) a - 'a', Trie(0,1)

length = getIndex(1+0)

if([length] null)
= Trie(1,0+1)
return depth
return (1,1)

0,0



887. Super Egg Drop

<https://leetcode.com/problems/super-egg-drop/>

given floor ?

Time Limit Exceeded

```
DP[floor][egg] == minOf(  
    for f in floor  
        1 + maxOf(dp[f-1][egg-1], dp[floor-f][egg])  
    )  
)  
broken  
not broken
```

$K * N * \log(N)$

binary search, 2 .. floor

```
broken = dp[mid-1][egg-1]  
notBroken = dp[floor-mid][egg]
```

```
broken < notBroken -> left = mid  
else -> right = mid
```

$dp[time][egg]$, with eggs(K) how many times try can test the floor N

```
dp[time][0] = 0  
dp[1][egg] = egg
```

$dp[time][egg] = dp[time-1][egg-1] + dp[time-1][egg] + 1$

if($dp \geq N$) return time

$K * N$

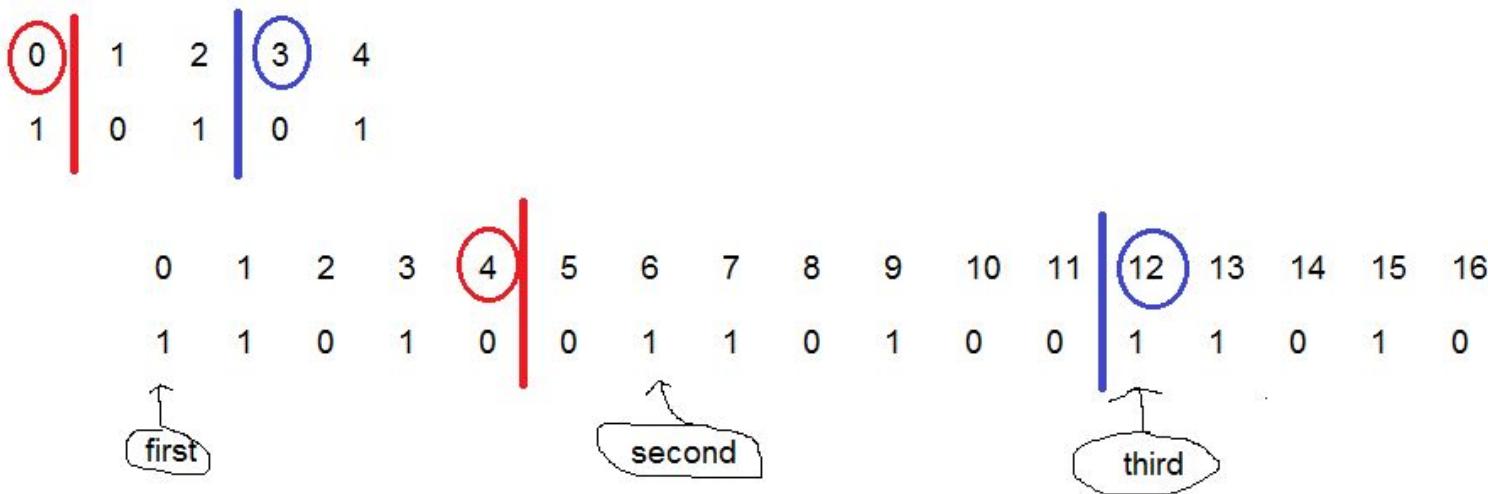
binary search 1..N

so hard, I recommend that we see the solution!

$K * \log(N)$

927. Three Equal Parts

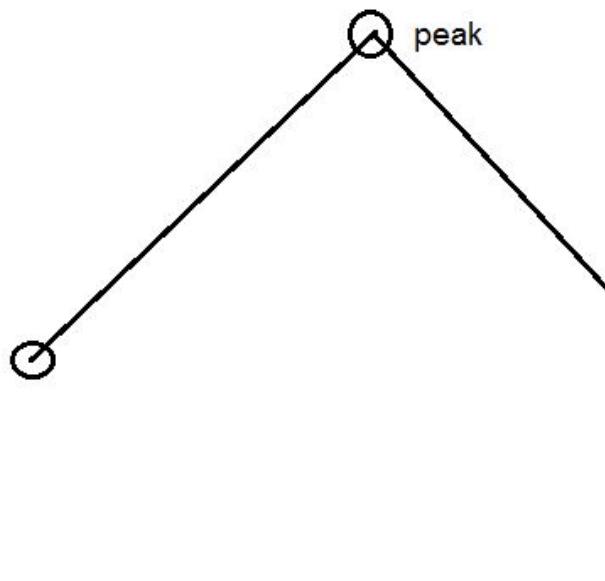
<https://leetcode.com/problems/three-equal-parts/>



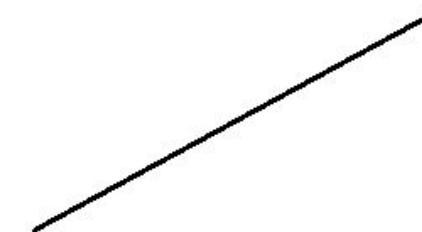
1095. Find in Mountain Array

<https://leetcode.com/problems/find-in-mountain-array/>

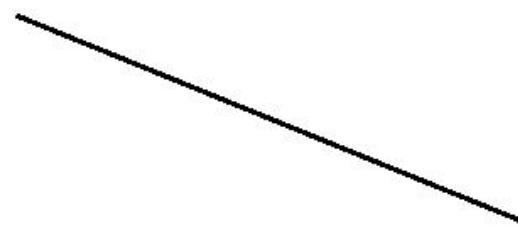
step1: binary search find the peak



step2: if target is in [0] to [peak], binary search



step3: if target is in [peak] to [length-1], binary search



1157. Online Majority Element In Subarray

<https://leetcode.com/problems/online-majority-element-in-subarray/>

摩尔投票法

Boyer-Moore Majority Vote Algorithm

```
var major = -1
var count = 0
for (index in left..right) {
    if (count == 0) {
        major = arr[index]
        ++count
    } else {
        count += if (arr[index] == major) 1 else -1
    }
}
count = 0
for (index in left..right) {
    if (arr[index] == major) {
        ++count
    }
}
return if (count >= threshold) major else -1
```

we need a list of index,
for example, [1,1,2,2,1,1]
the list of index for 1 is [0,1,4,5]

if we want to find majority element from index 1 to 5,

just find the index in list then the number of 1 is 3.

we can compare it with threshold.

so how many time ?

1482. Minimum Number of Days to Make m Bouquets

<https://leetcode.com/problems/minimum-number-of-days-to-make-m-bouquets/>

1 10 3 10 2

m = 3

k = 1

when target = 3

the array could be

O _ O _ O

k = 1

so total is 3 >= m = 3

binary search from min and max of array,

if(isvalid)

right = mid

else

left = mid

if left is valid -> left

if right is valid -> right

else -> -1