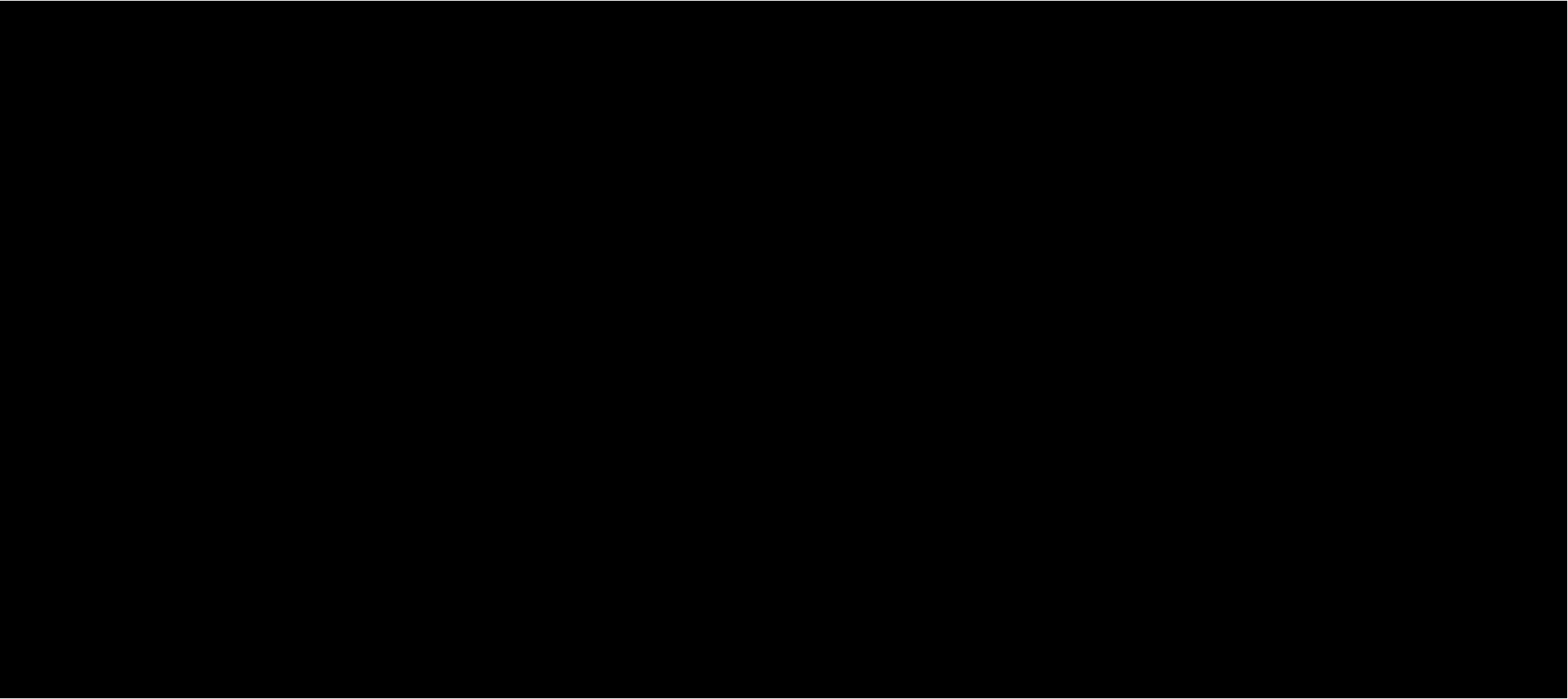


# 16| IP

IP IP IP



202.102.133.13 IP IP IP [202.102.133.0, 202.102.133.255] IP “ ”

[202.102.133.0, 202.102.133.255]  
[202.102.135.0, 202.102.136.255]  
[202.102.156.34, 202.102.157.255]  
[202.102.48.0, 202.102.48.255]  
[202.102.49.15, 202.102.51.251]  
[202.102.56.0, 202.102.56.255]

IP 12 IP IP

“

”

Bug

.

Donald E.Knuth

3

“

1946

1962

”

“

”

a[5] a[6] a[7]

8

8

5

8

a[4]

8 6

5 9

5 9

7 a[7]

8

a[7]

8

8

8

5

100

100

```
public int bsearch(int[] a, int n, int value) {
    int low = 0;
    int high = n - 1;
    while (low <= high) {
        int mid = low + ((high - low) >> 1);
        if (a[mid] >= value) {
            high = mid - 1;
        } else {
            low = mid + 1;
        }
    }

    if (low < n && a[low]==value) return low;
    else return -1;
}
```

90%

```
public int bsearch(int[] a, int n, int value) {
    int low = 0;
    int high = n - 1;
    while (low <= high) {
        int mid = low + ((high - low) >> 1);
        if (a[mid] > value) {
            high = mid - 1;
        } else if (a[mid] < value) {
            low = mid + 1;
        } else {
            if ((mid == 0) || (a[mid - 1] != value)) return mid;
            else high = mid - 1;
        }
    }
    return -1;
}
```

a[mid]=value      a[mid]      value

a[mid]>value

high= mid-1

a[mid]<value

low=mid+1

a[mid]

a[mid]

a[mid]

a[mid]

11

mid

0

mid

0

a[mid]

a[mid-1]

value

a[mid]

a[mid]

a[mid-1]

value

a[mid]

high=mid-1

[low, mid-1]

Bug

```
public int bsearch(int[] a, int n, int value) {
    int low = 0;
    int high = n - 1;
    while (low <= high) {
        int mid = low + ((high - low) >> 1);
        if (a[mid] > value) {
            high = mid - 1;
        } else if (a[mid] < value) {
            low = mid + 1;
        } else {
            if ((mid == n - 1) || (a[mid + 1] != value)) return mid;
            else low = mid + 1;
        }
    }
    return -1;
}
```

11

a[mid]

a[mid]

a[mid+1]

value

a[mid]

a[mid]

a[mid+1]

value

a[mid]

low=mid+1

[mid+1, high]

3 4 6 7 10

5

6

```
public int bsearch(int[] a, int n, int value) {
    int low = 0;
    int high = n - 1;
    while (low <= high) {
        int mid = low + ((high - low) >> 1);
        if (a[mid] >= value) {
            if ((mid == 0) || (a[mid - 1] < value)) return mid;
            else high = mid - 1;
        } else {
            low = mid + 1;
        }
    }
    return -1;
}
```

a[mid]

value

[mid+1, high]

low=mid+1

a[mid]

value

7

a[mid]

a[mid]

value

a[mid]

a[mid-1]

value

[low, mid-1]

high

mid-1

3 5 6 8 9 10

7

6

```
public int bsearch7(int[] a, int n, int value) {
    int low = 0;
```

```
int high = n - 1;
while (low <= high) {
    int mid = low + ((high - low) >> 1);
    if (a[mid] > value) {
        high = mid - 1;
    } else {
        if ((mid == n - 1) || (a[mid + 1] > value)) return mid;
        else low = mid + 1;
    }
}
return -1;
}
```

IP

IP

12

IP

IP

32

“

”

IP

IP

IP IP

IP

IP

“

”

“

”

Bug free

Bug

4 5 6 1 2 3

“

”

- Smallfly 2018-10-27 02:49:06

1.
2.

1. x;
2. +x ;
3.
4. -x

O(N

mid  
mid

O(logN) [62 ]

- zixuan 2018-10-31 08:50:52  
leetcode 33 [21 ]

- charon 2018-10-26 07:32:49  
JavaScript  
array value  
low,high value ~  
function search(array,value){  
let low = 0;  
let high = array.length - 1;  
  
while(low <= high){  
let mid = low + ((high - low) >> 1);  
if(value == array[low]) return low;  
if(value == array[high]) return high;  
if(value == array[mid]) return mid;  
  
if(value > array[mid] && value > array[high] && array[mid] < array[low]){  
high = mid - 1;  
}else if(value < array[mid] && value < array[low] && array[mid] < array[low]){  
high = mid - 1;  
}else if(value < array[mid] && value > array[low]){  
high = mid - 1;  
}else{  
low = mid + 1;  
}  
}  
  
return -1  
} [6 ]

- Victor 2018-10-27 14:38:14  
IP

12w IP  
ip\_table area\_name | start\_ip | end\_ip start\_ip end\_ip  
SQL  
select area\_name from ip\_table where input\_ip >= start\_ip and input\_ip <= end\_ip;

[5 ]

2018-10-28 15:27:38

- 2018-12-07 02:25:08

- 1.
2. ,

```

,
:
/**
 *
 *
 * @param null
 * @return
 * @author xiongfán
 * @date 2018/12/7 9:43:00
 */
public static int getFirstGreaterValue(int[] array,int value) {
int low = 0;
int high = array.length - 1;

while (low <= high) {
int mid = low + (high - low) >> 1;
if (array[mid] < value) {
low = mid + 1;
} else if (array[mid] > value) {
high = mid - 1;
} else {

if (mid == 0 || array[mid - 1] < array[mid]) {
return mid;
}
high = mid - 1;

}

return low>array.length-1?-1:low;
}
```



```
/**
 *      :
 *
 * @param null
 * @return
 * @author xiongfán
 * @date 2018/12/7 10:03:00
 */
public static int getLastLessValue(int[] array,int value) {
    int low = 0;
    int high = array.length - 1;

    while (low <= high) {
        int mid = low + (high - low) >> 1;
        if (array[mid] > value) {
            high = mid - 1;
        } else if (array[mid] < value) {
            low = mid + 1;
        } else {
            if (mid > array.length-1 || array[mid] < array[mid + 1]) {
                return mid;
            }
            low = mid + 1;
        }
    }

    return high<0?-1:high;
}
[4 ]
```

- 2018-10-31 05:15:22

```
@
/**
 *      4 5 6 1 2 3
 *
 *      O(n)
 */
public static int forEqualsThan(int[] arr, int num) {
    if (arr[0] == num) {
        return 0;
    }
    int length = arr.length;
    int low = 0;
    int high = length - 1;
```

```
//
for (int i = 0; i < length; i++) {
    if (i == length - 1) {
        if (arr[i] > arr[0]) {
            low = i;
            high = 0;
            break;
        }
    } else {
        if (arr[i] > arr[i + 1]) {
            low = i;
            high = low + 1;
            break;
        }
    }
}

//                                low    high

if (arr[0] < num) {
    high = low;
    low = 0;
}

if (arr[0] > num) {
    low = high;
    high = length - 1;
}

while (low <= high) {
    int index = low + ((high - low) >> 1);
    if (arr[index] > num) {
        high = index - 1;
    }
    if (arr[index] < num) {
        low = index + 1;
    }
    if (arr[index] == num) {
        return index;
    }
}

return -1;
} [3 ]
```

- komo0104 2018-10-25 18:10:25

[illegible]

O(n) [3 ]

- QFann 2018-12-18 08:12:10

```
public int search(int[] nums, int target) {
    if(nums.length ==0) return -1;
    if(nums.length ==1){
        if(nums[0] == target) return 0;
        else return -1;
    }
    int low = 0;
    int high = nums.length - 1;
    int index = subIndex(nums,low,high);
    if(index != -1){
        int val = binarySearch(nums,low,index,target);
        if (val != -1) return val;
        return binarySearch(nums,index+1,high,target);
    }
    return binarySearch(nums,low,high,target);
}
```

```
public static int subIndex(int [] nums,int low,int high){
    while (low <= high){
        int mid = low + ((high - low )>> 1);
        if(nums.length < 1) return -1;
        if(nums[mid] > nums[mid+1]) return mid;
        else if( nums[mid] < nums[low] ) high = mid ;
        else if (nums[mid] > nums[high]) low = mid ;
        else return -1;
    }
    return -1;
}
```

```
public static int binarySearch(int[] nums,int low,int high,int target){
    while (low <= high){
        int mid = low + ((high - low)>>1);
        if(nums[mid] == target) return mid;
        else if (nums[mid] < target) low = mid + 1;
        else high = mid -1;
    }
    return -1;
} [2 ]
```

- 2018-11-03 02:05:11

- 1.
- 2.
- 3.
- 4.

- 1.

- 2. “ ” “

- 1. IP  
[202.102.133.0, 202.102.133.255]  
[202.102.135.0, 202.102.136.255]  
[202.102.156.34, 202.102.157.255]  
[202.102.48.0, 202.102.48.255]  
[202.102.49.15, 202.102.51.251]  
[202.102.56.0, 202.102.56.255]

12

IP

IP

- 2. 4 5 6 1 2 3 “ ” [2 ]

- -libo 2018-10-27 02:30:56

- 1. index <pre, >=next;
- 2. [0,index-1], [index,length-1];
- 3.

2\*log(n).

[2 ]