Binary Search in an Array

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int target = scn.nextInt();
    System.out.println(BS(arr, n, target));
}
public static int BS(int[] arr, int n, int target) {
    int i = 0;
    int j = n - 1;
    while ( i <= j ) {
        int mid = (i + j) / 2;
        if ( target == arr[mid] ) {
            return mid;
        } else if ( target < arr[mid] ) {
            j = mid - 1;
        } else {
            i = mid + 1;
        }
    }
    return -1;
}
```

T.C = O(log N)

S.C = O(1)

Search Character

arr = ['a' , 'c' , 'f' , 'u' , 'z'] target = 'd'

0 1 2 3 4

i mid j

Code

```
int i=0, j = n-1;
while(i<=j){
    int mid = (i+j)/2;
    if( ch >= arr[mid]){
a       i = mid+1;
    }else{
b       j = mid-1;
    }
}
return arr[i];
```

0	1	2	3	4
'a'	'b'	'c'	'd'	'e'

j i mid

ch = 'c'

Imp

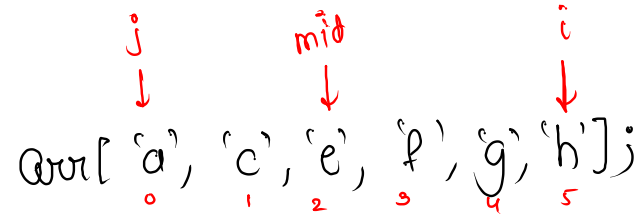
if required answer == target, return arr[mid]

if required answer > target, return arr[i]

if required answer < target, return arr[j]

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    char ch = scn.next().charAt(0);
    int n = scn.nextInt();
    char[] arr = new char[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.next().charAt(0);
    }
    BS(arr, n, ch);
}

public static void BS(char[] arr, int n, char ch) {
    int i = 0;
    int j = n - 1;
    while ( i <= j ) {
        int mid = (i + j) / 2;
        if ( ch >= arr[mid] ) {
            i = mid + 1;
        } else {
            j = mid - 1;
        }
    }
    if ( i == n ) {
        System.out.println("-1");
    } else {
        System.out.println(arr[i]);
    }
}
```



ch = 'b'

ans = 'c'

ch >= arr[mid]

'b' >= 'e'

'b' >= 'a'

'b' >= 'c'

→ Variation of BS

arr

0	1	2	3	4	5	6	7	8	9
1	2	2	2	2	3	3	3	3	4

tar = 3

mid

i

j

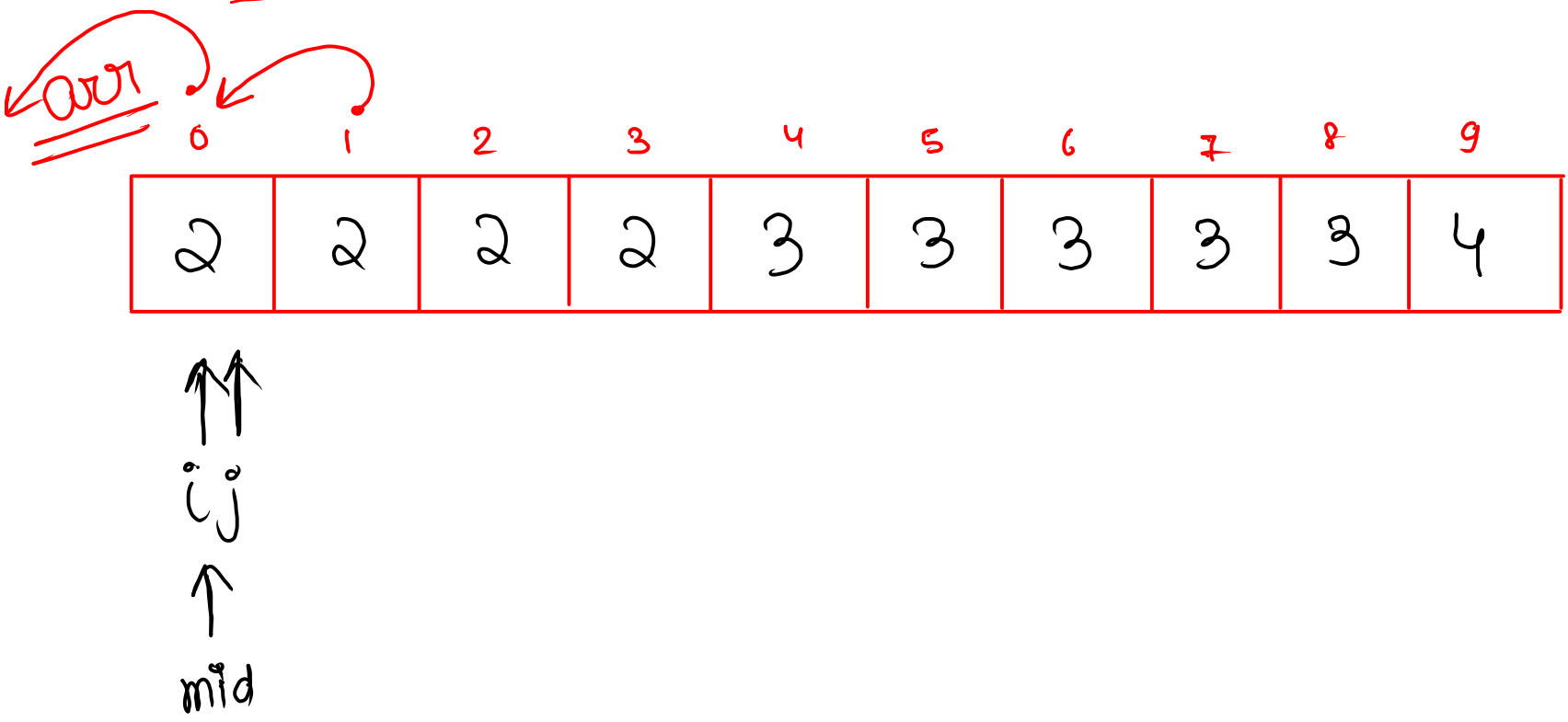
```
int i=0, j=n-1;
while( i<=j) {
    mid = (i+j)/2;
    if ( tar == arr[mid] ) {
        if (arr[mid] == arr[mid+1]) {
            i = mid+1;
        } else {
            return mid;
        }
    } else if ( tar < arr[mid]) {
        j = mid-1;
    } else {
        i = mid+1
    }
}
```

mid+1 < n &&

Binary Search
Upper bound
(BSUB)

T.C O(log N)

→ Binary Search Lower Bound



ans = 2

code

```
int i=0, j=n-1;
while( i<=j) {
    mid = (i+j)/2;
    if ( tar == arr[mid]) {
        if( arr[mid] == arr[mid-1]) {
            j = mid-1;
        } else {
            return mid;
        }
    } else if ( tar < arr[mid]) {
        j = mid-1;
    } else {
        i = mid+1
    }
}
```

mid-1>=0 &&

Qwel first and last occurence

arr

0	1	2	3	4	5	6	7	8	9
1	2	2	2	2	3	3	3	3	4

tar = 2

left = BSLB

right = BSUB

ans = right - left