18 December 2024 20:26

Find Last Occurence

n = 6

```
        0  1  2  3  4  5
arr = [ 1, 2, 3, 3, 3, 4 ]
```

K = 3

Output

↳ 4

Solution:-

```
       0  1  2  3  4  5
arr = [1, 2, 3, 3, 3, 4]
       ↑     ↑        ↑
     left    |      right
            mid
```

int lastindex = -1;

```
if( arr[mid] == target )
    lastindex = mid;   // 2
    left = mid+1;
}
```

```
0  1  2  3  4  5
1  2  3  3  3  4
         ↑  ↑  ↑
      left  |  right
           mid     ↑
```

left ↑ right

mid. ↑ left → mid.

lastindex = 4

```
int left=0, right=n-1;
int lastindex=-1;
while(left<=right) {
 int mid=(left+right)/2;
  if (arr[mid]==target) {
     lastindex=mid;
     left= mid+1;
  }else if (arr[mid]<target) {
      left=mid+1;
  }else {
     right=mid-1;
  }
  }
s.o.pln(lastindex);
```

Search Index Position

① n = 4

$$\begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ arr = [1, 3, 5, 6] \end{array}$$

K = 5

Output

2

② n = 4

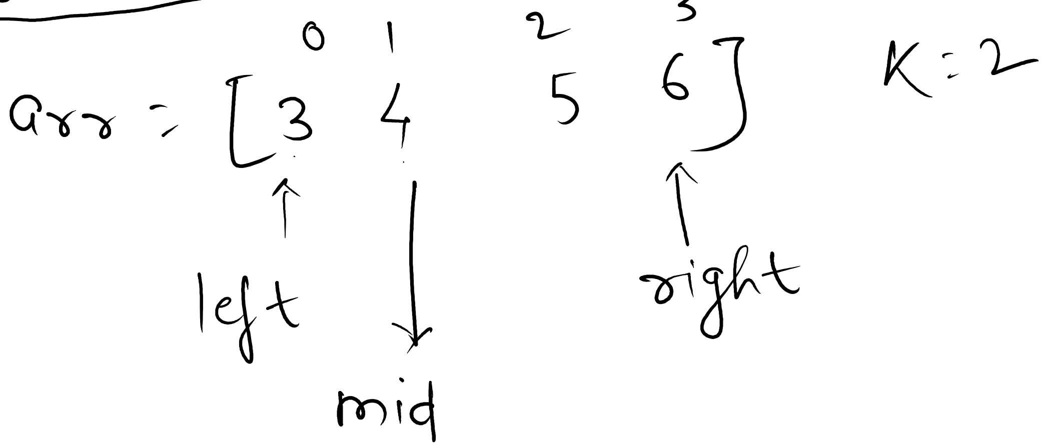
$$\begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ arr = [1, 3, 5, 6] \end{array}$$

K = 2

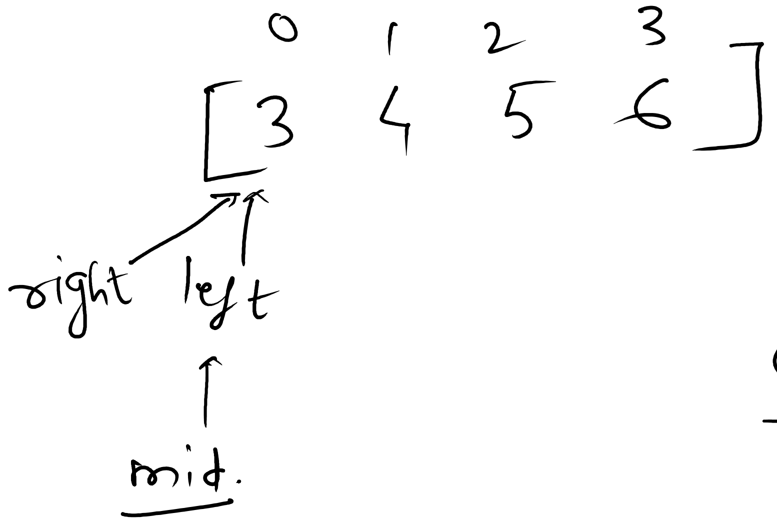
Output

1

Efficient Approach :-



Arr[mid] > k → ②



arr[mid] > k.

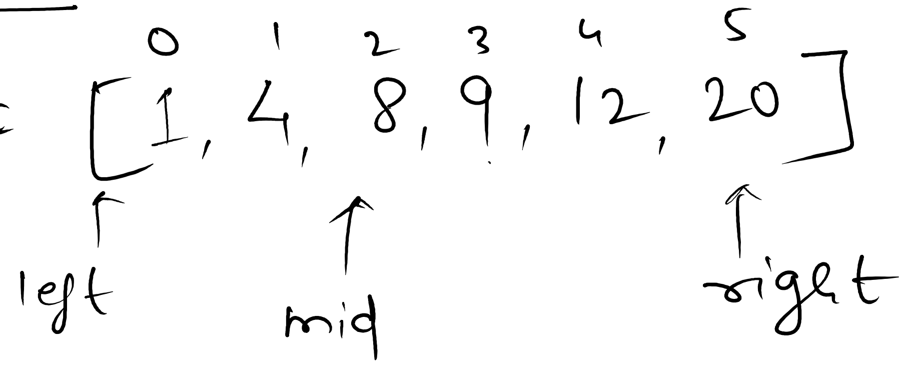
right = -1.

(left <= right) X

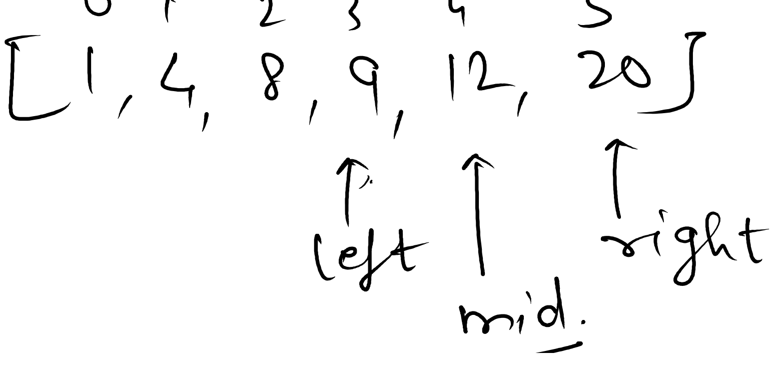
(left, right, mid) → ans

Example 2:- K = 10

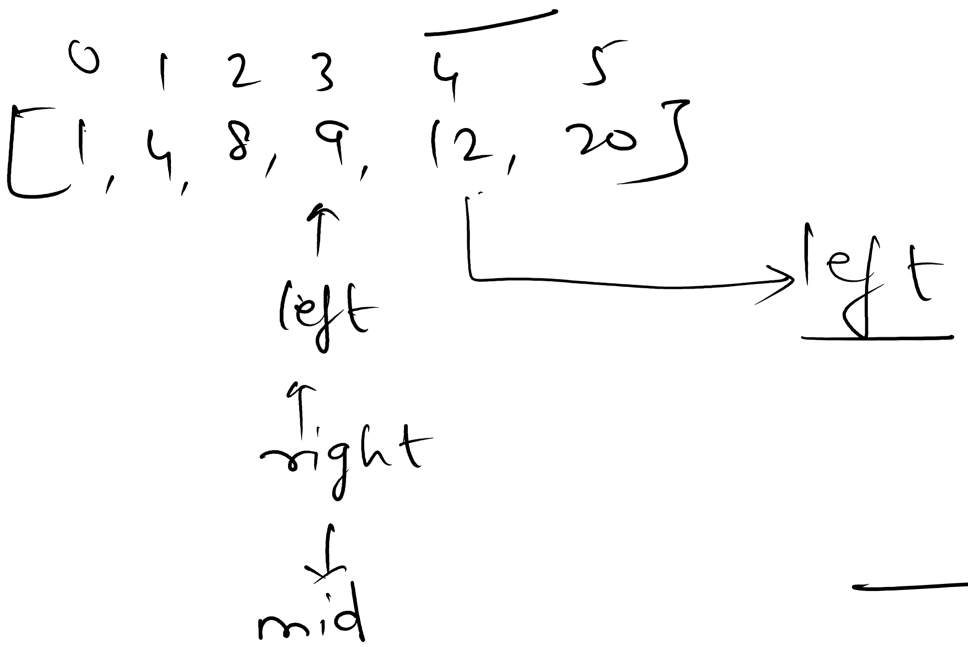
arr = [1, 4, 8, 9, 12, 20] → ④



arr[mid] < K



arr[mid] > K.



if(arr[mid] < k)

left = mid+1;

left = 4

right = 3.

- While will not run because
(left<=right)→false

```
int left=0, right=n-1;
 While (left<=right) {
 int mid= (left+right)/2;
  if (arr[mid]==k) {
    S.o.Pln(mid);
if (arr[mid] == k) {
    s.o.pln(mid);
    return;
 } else if (arr[mid] < k) {
        left = mid+1;
   } else {
        right = mid-1;
     }
 }
s.o.pln(left);
```

18 December 2024 21:45

Find Square Root

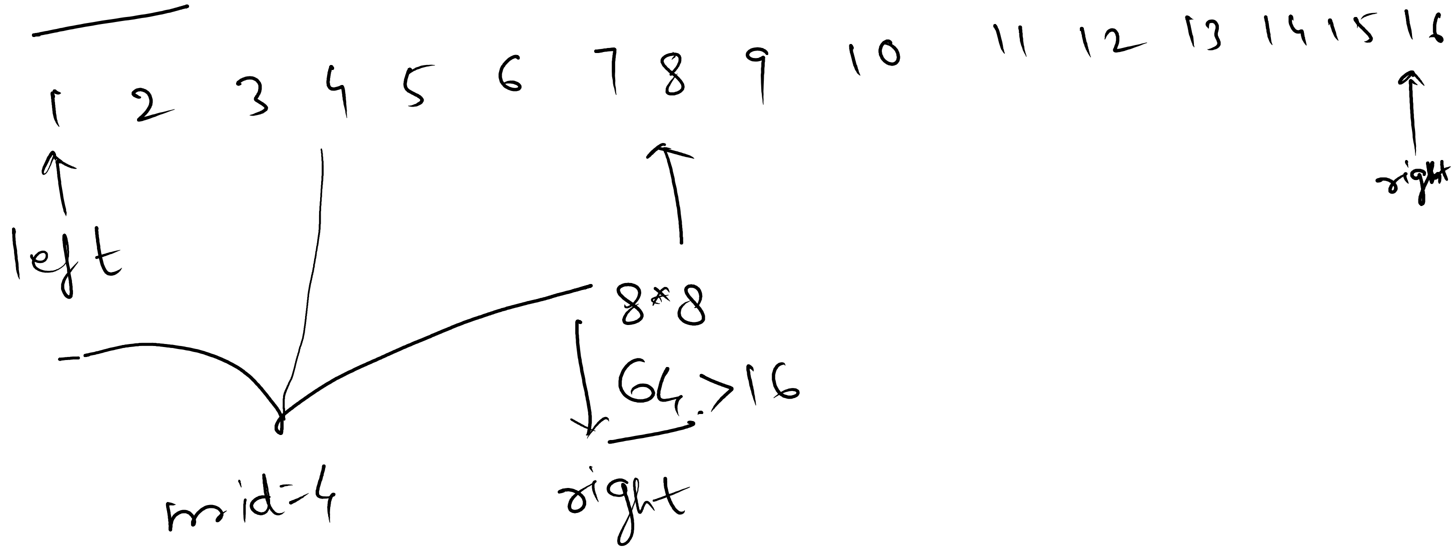
n = 16

Output -> 4

n = 79

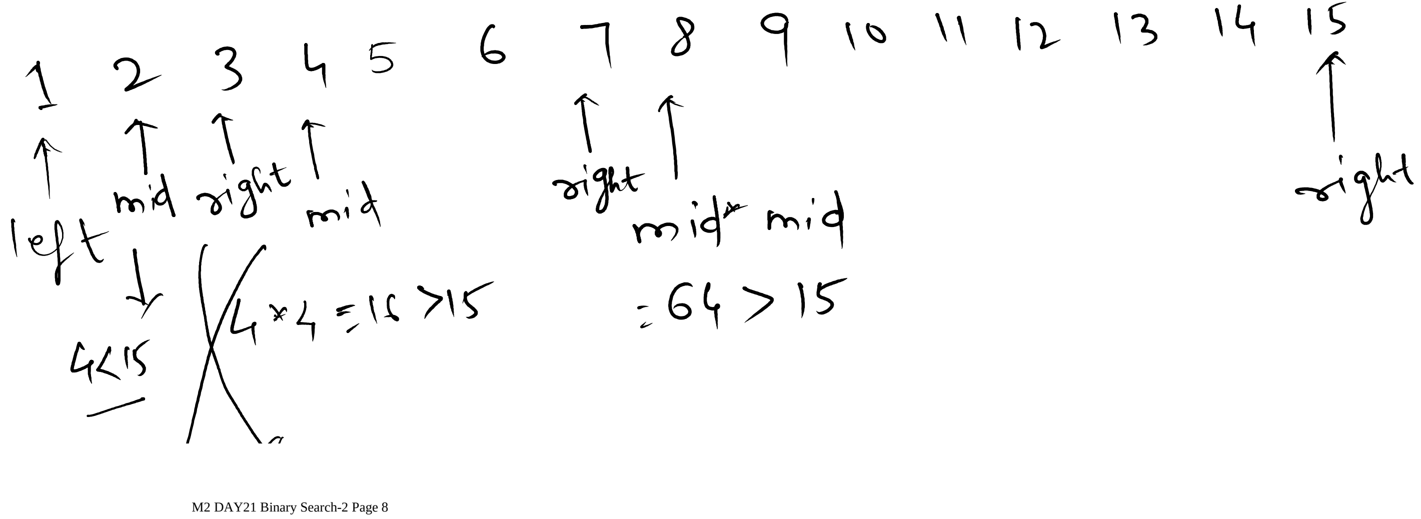
Output -> 8

n = 16



4*4=16

n=15 , Output -> 3

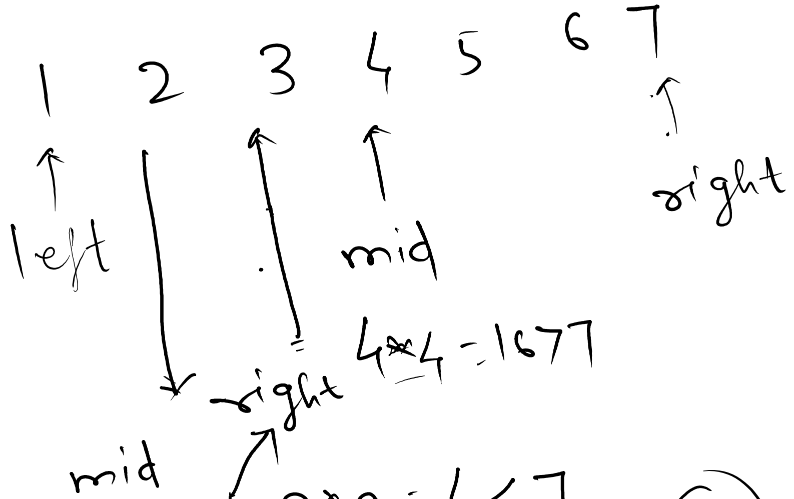


4<15

left → mid = 9<15, left=mid+1

→ left=4
right=3

n=7, output → 2



2*2=4<7
ans=mid;

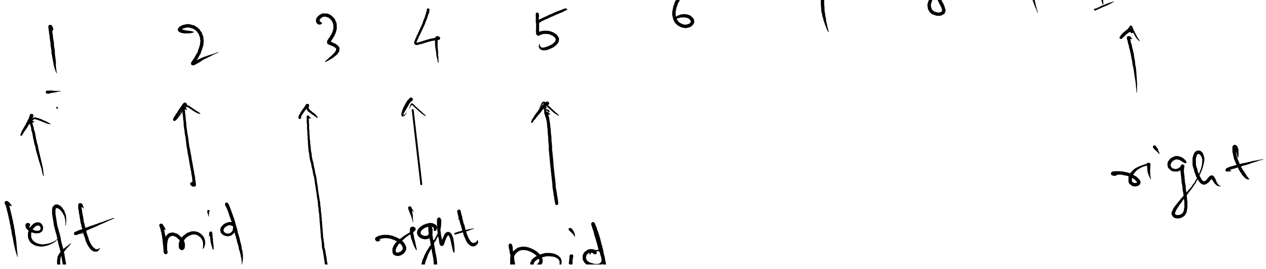
$\boxed{2}$

left → mid=3
9>7.

2 | 3 → 9>7.
4<7

left=3.
right=2. → left<=right.

n=10



righ

left mid | right mid

↓ ↓

4<10 left 25>10

↓

mid→ 3*3=9<10

ans=3.

left=4 right=mid

16>10.

right=3.

if(left<=right) ×

```
int left=1, right=n;
int ans=0;
while( left<=right){
 int mid=(left+right)/2;
  if(mid*mid==n){
    S.O.pln(mid);
    return;
  }else if(mid* mid <n){
     ans=mid;
     left=mid+1;
  }else{
  }else{
    right=mid-1;
   }
 }
S.o.pln(ans);
```