Last Occurrence

arr = { 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 6, 6 } tar = 3

```
if (arr[mid] < tar) go right
else if (arr[mid] > tar) go left
else {
    idx=mid
    go right
}
```

Search in descending sorted array

arr = { 100, 91, 87, 76, 66, 52, 43, 35, 29, 13, 5 }

lo hi mid

tar = 43

```
if( mid == tar) return
else if (mid > tar ) lo = mid+1;
else if (mid < tar) hi = mid-1;
```

H.W. : Find Peak in a mountain array.

{ -1, 0, 1, 2, 5, 6, 7, 8, 3, 2, 1, 0, -1 }

lo m hi

if (increasing) go right

if (decreasing) go left

else → peak

Problem in $\boxed{mid = \frac{lo + hi}{2}}$ → integer overflow error

$lo = 10$, $hi = 2^{31} - 1$

$$mid = \frac{lo + hi + lo - lo}{2} = \frac{2lo + hi - lo}{2} = lo + (hi - lo)/2$$

Although,

$lo = -2^{31}$, $hi = 2^{31} - 1$

Ques: Floor in a sorted array

ans <= x

arr = { 1, 2, 4, 10, 10, 12, 19 } x = 5 idx = -1 1 2

if(mid > x) go left

if(mid <= x) mark & go right

HW: Maximum count of positive integer and negative integer

Leetcode 2529

Ques: Square Root

Brute Force

$1 \le \sqrt{n} \le n$

$n = 15$

$T.C. = O(\sqrt{n})$

```
for(i=1 ; i<=n; i++){
    if(i*i>n) break;
```

3

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

root = 1 1 2 3

Ques: Square Root

Binary Search

$n=20$

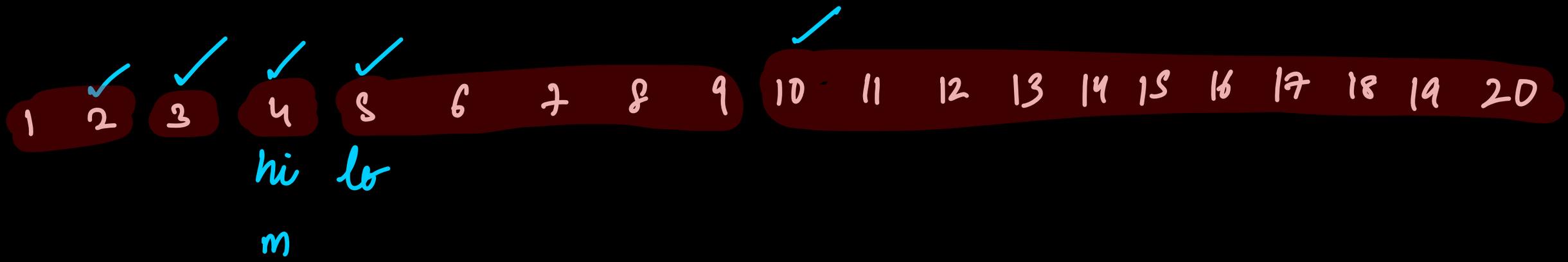
$1 \le \sqrt{n} \le n$

lo hi

Math.sqrt(n)

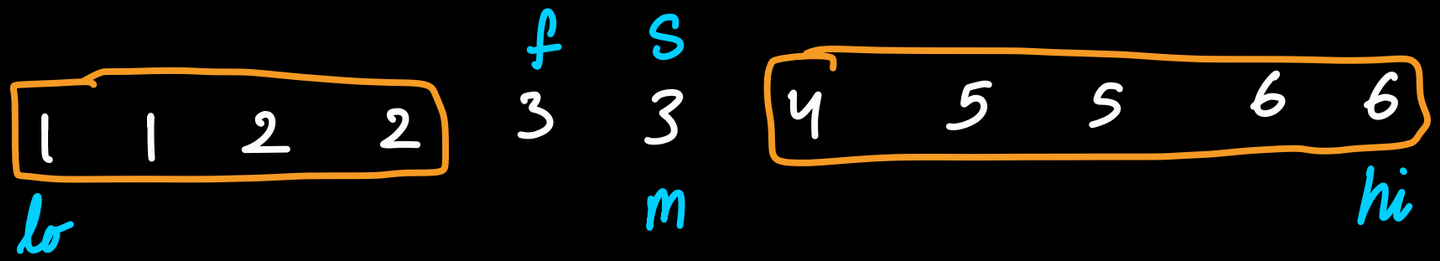
↓

≈ O(1)



return hi

Ques: Single among Doubles in a sorted array



Ques: Arranging Coins

$$n = 1 + 2 + 3 + \cdots + K$$

$$n = \frac{K*(K+1)}{2}$$

$$2n = K^2 + K$$

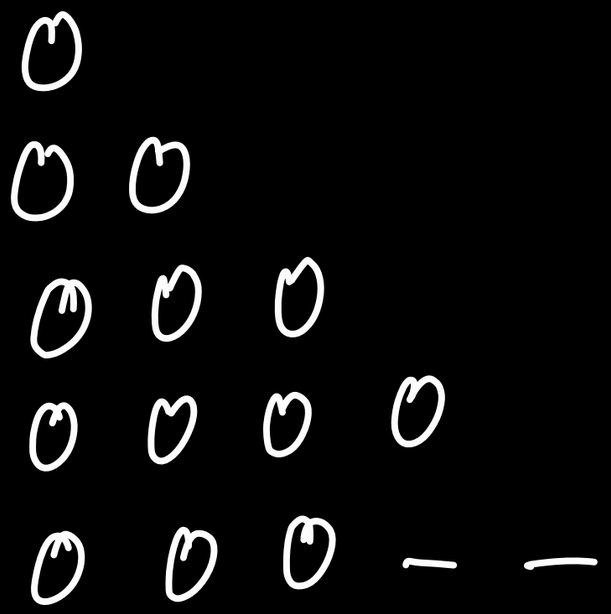
$$\Rightarrow K^2 + 1 \cdot K - 2n = 0$$

$$\Rightarrow K = \frac{-1 + \sqrt{1+8n}}{2}$$

Leetcode 441

Ques: Arranging Coins

$n = 13$



$$k = \frac{\sqrt{8n+1} - 1}{2}$$

$$\frac{\sqrt{8 \cdot 13 + 1} - 1}{2} = \frac{\sqrt{105} - 1}{2} = \frac{10 - 1}{2} = 4$$

Leetcode 441