Session 4

29 July 2021 10:30

* Height Checker

Input : 1 1 4 2 1 3 — n — height

expected: 1 1 1 2 3 4 — —

Output : 3

Approach 1:

```
int count=0;
int expected[n];
for(i=0; i<n; i++)
{   expected[i] = height[i];
}
sort(expected, expected+n);
for(i=0; i<n; i++)
{ if(expected[i] != height[i])
  {   count++;
  }
}
print/return count;
```

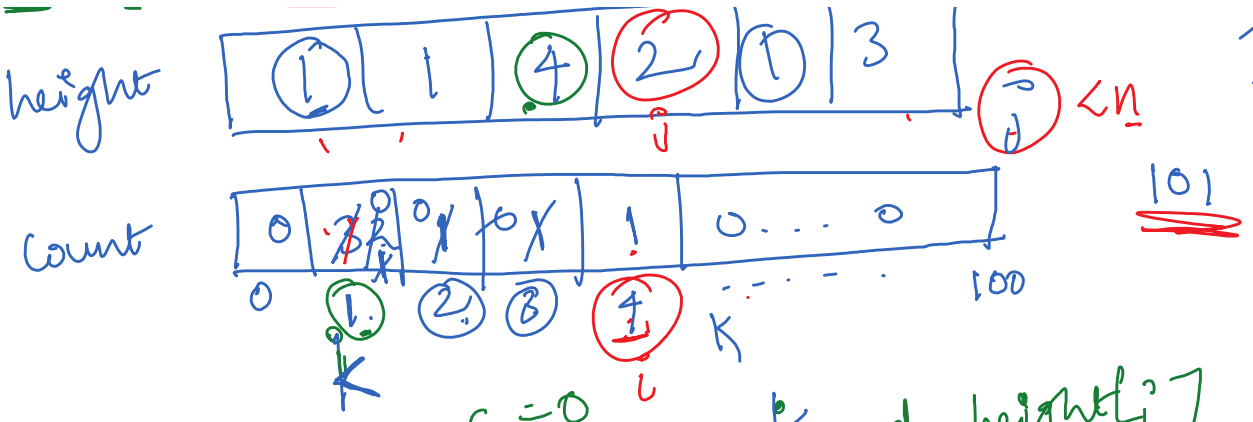
Approach 2 : range — 1 to 100

Constraints:

- `1 <= heights.length <= 100`
- `1 <= heights[i] <= 100`

Height 1 1 4 2 1 3 — n

count = ~~2~~ 3



int i=1, j=0;

c=0

c++

O(n

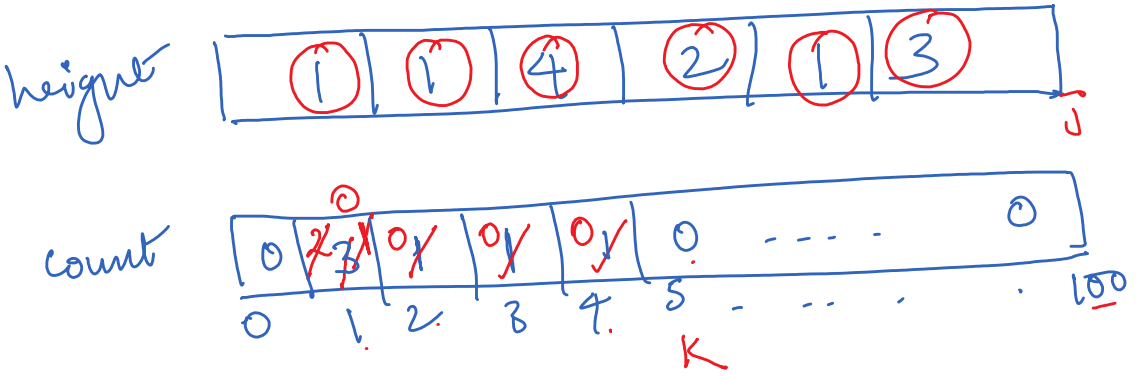
O(101) Const

Extra — O(1)

```
int unequal=0;
int count[101] = {0},
for( i=0; i<n; i++)
{   count[height[i]]++;
}
int K=1, j=0;
while(i<101 && j<n)
{   if( count[i]==0)
        i++;
    else{ if(i != height[j])
            unequal++
        j++;
        count[i]--;
```

3 3

K=1, j=0



K and weights