PW SKILLS

Quick Sort

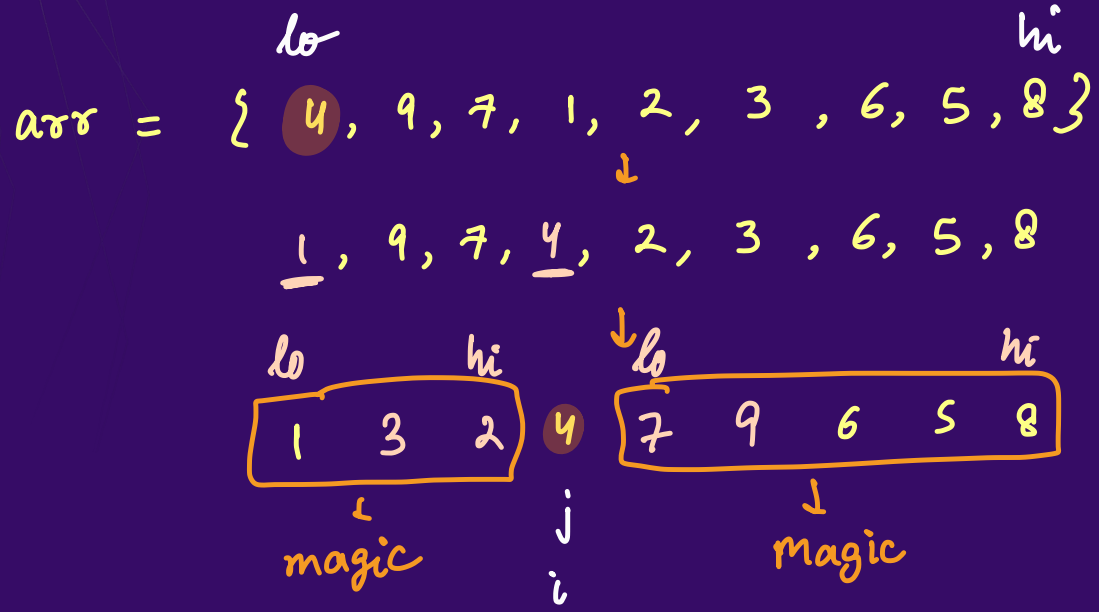
& Quick Select

Today’s checklist

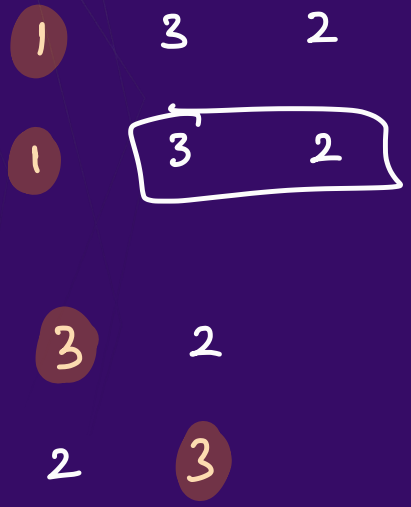
PW SKILLS

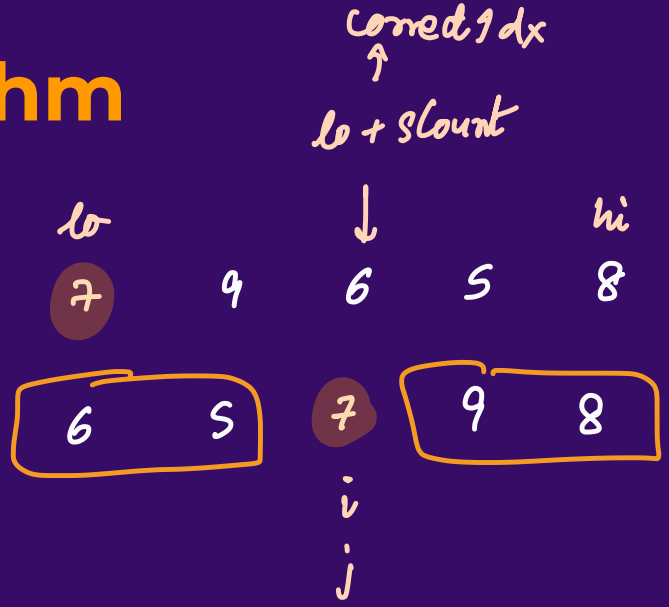
1. Quick Sort Algorithm
2. Quick Sort Time and Space Complexity
3. Randomised Pivot point
4. Stability
5. Quick Select Algorithm and kth Smallest/Largest

QuickSort Algorithm



QuickSort Algorithm





Partition Algorithm

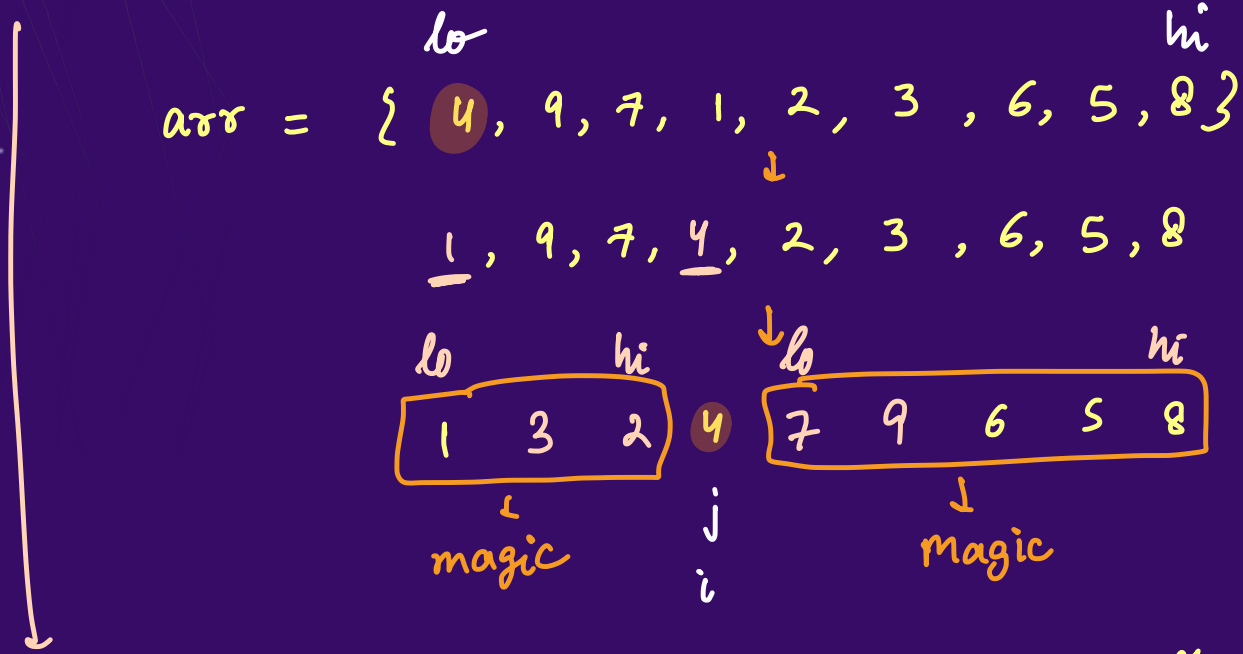
```java
public static int partition(int[] arr, int lo, int hi){
    int pivot = arr[lo], pivotIdx = lo;
    int smallerCount = 0;
    for(int i=lo+1;i<=hi;i++){
        if(arr[i]<=pivot) smallerCount++;
    }
    int correctIdx = pivotIdx + smallerCount;
    swap(arr,pivotIdx,correctIdx);
    // partition
    int i = lo, j = hi;
    while(i<correctIdx && j>correctIdx){
        if(arr[i]<=pivot) i++;
        else if(arr[j]>pivot) j--;
        else if(arr[i]>pivot && arr[j]<=pivot){
            swap(arr,i,j);
        }
    }
    return correctIdx;
}
```

Code

```java
public static void quickSort(int[] arr, int lo, int hi){
    if(lo>=hi) return;
    // pivot (arr[lo]) ko sahi jagah rakho
    // & left part me <=pivot
    int idx = partition(arr,lo,hi);
    quickSort(arr,lo,idx-1);
    quickSort(arr,idx+1,hi);
}
```

For each level, ~ n operations are performed

Time and Space complexity

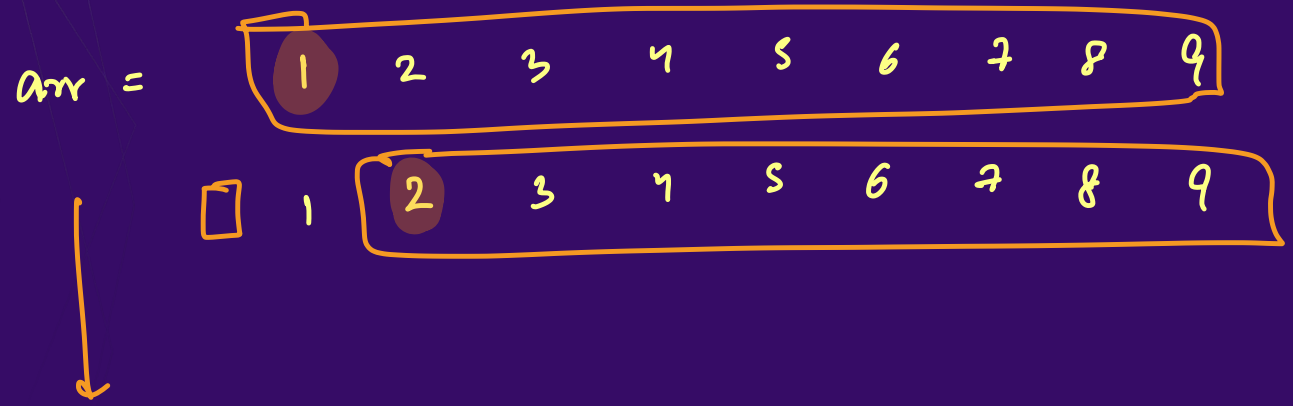


Avg. Case = $O(n * \log n)$

At Max itni hi calls lagti

Space Complexity : Recursive call stack space → $O(\log n)$

Worst Case TC



the no. of levels here are not 'logn' they are n

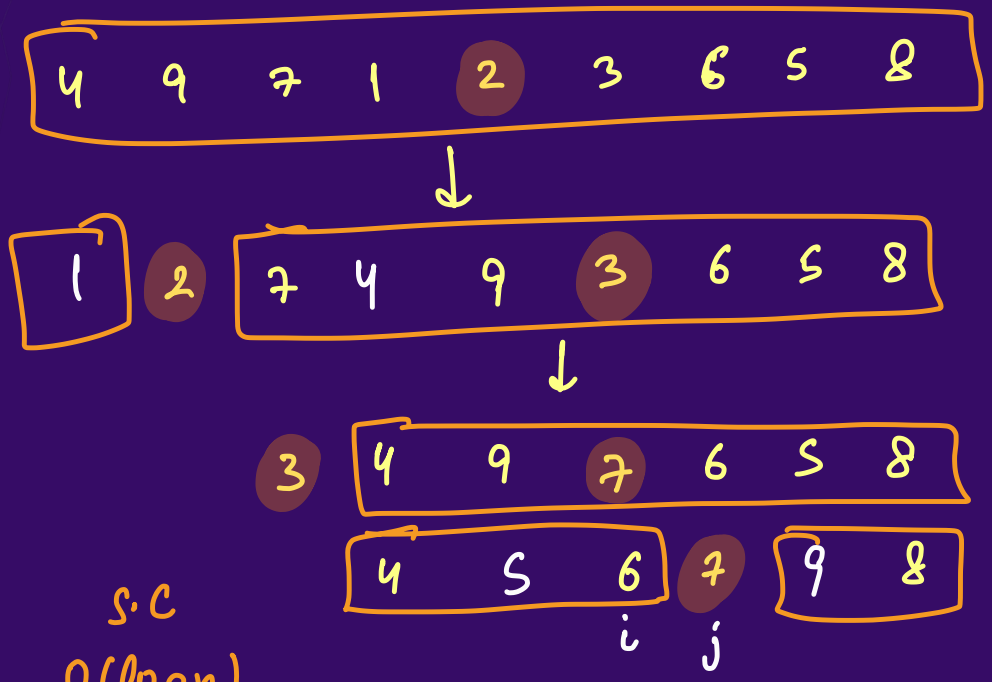
No. of ops ~ $n + n-1 + n-2 \cdots 1$

$= \frac{n(n+1)}{2} \sim O(n^2)$ T.C.

$O(n)$ S.C.

Randomized Pivot point

Instead of choosing arr[lo] as pivot, we can choose arr[$\frac{lo+hi}{2}$] as pivot



Worst Case

T.C

$O(n \log n)$

S.C

$O(\log n)$

Stability of Quick Sort

Unstable

Merge Sort vs Quick Sort

PW SKILLS

↓ Worst Case is better

Stable

↓ More space optimised

Ques:

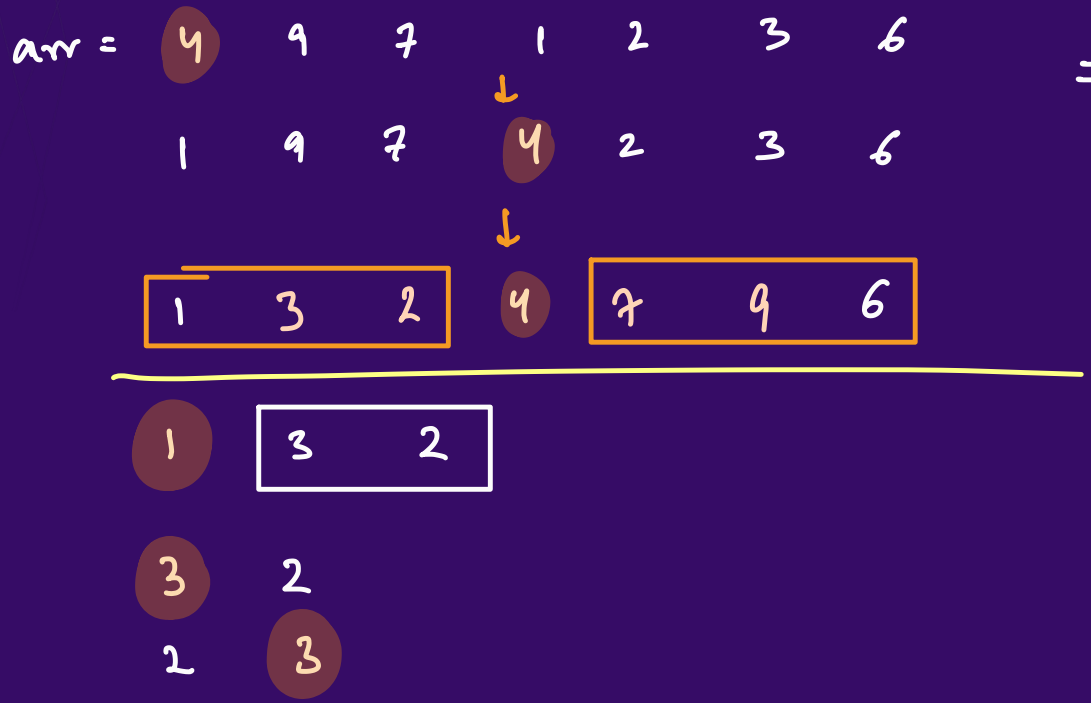
n = 7

Smallest

Q : Kth Largest Element in an Array.

K^{th} Largest

$= (n-K+1)$ Smallest



K = 2

[Leetcode 215]

Ques:

Q : Kth Largest Element in an Array.

$2\left[n + \frac{n}{2} + \frac{n}{4} + \cdots\right]$

$= 2n\left[1 + \frac{1}{2} + \frac{1}{4} \cdots\right]$

$< 4n$

```
public void quickSelect(int[] arr, int lo, int hi, int k){
    if(lo>hi) return;
    int idx = partition(arr,lo,hi);
    if(idx==k-1){
        ans = arr[idx];
        return;
    }
    if(k-1 < idx) quickSelect(arr,lo,idx-1,k);
    else quickSelect(arr,idx+1,hi,k);
}
```

T.C. = O(n) [Best & Avg. Case]

→ $O(n^2)$ Worst Case ∝∝

O(nlogn)

[Leetcode 215]

THANK YOU