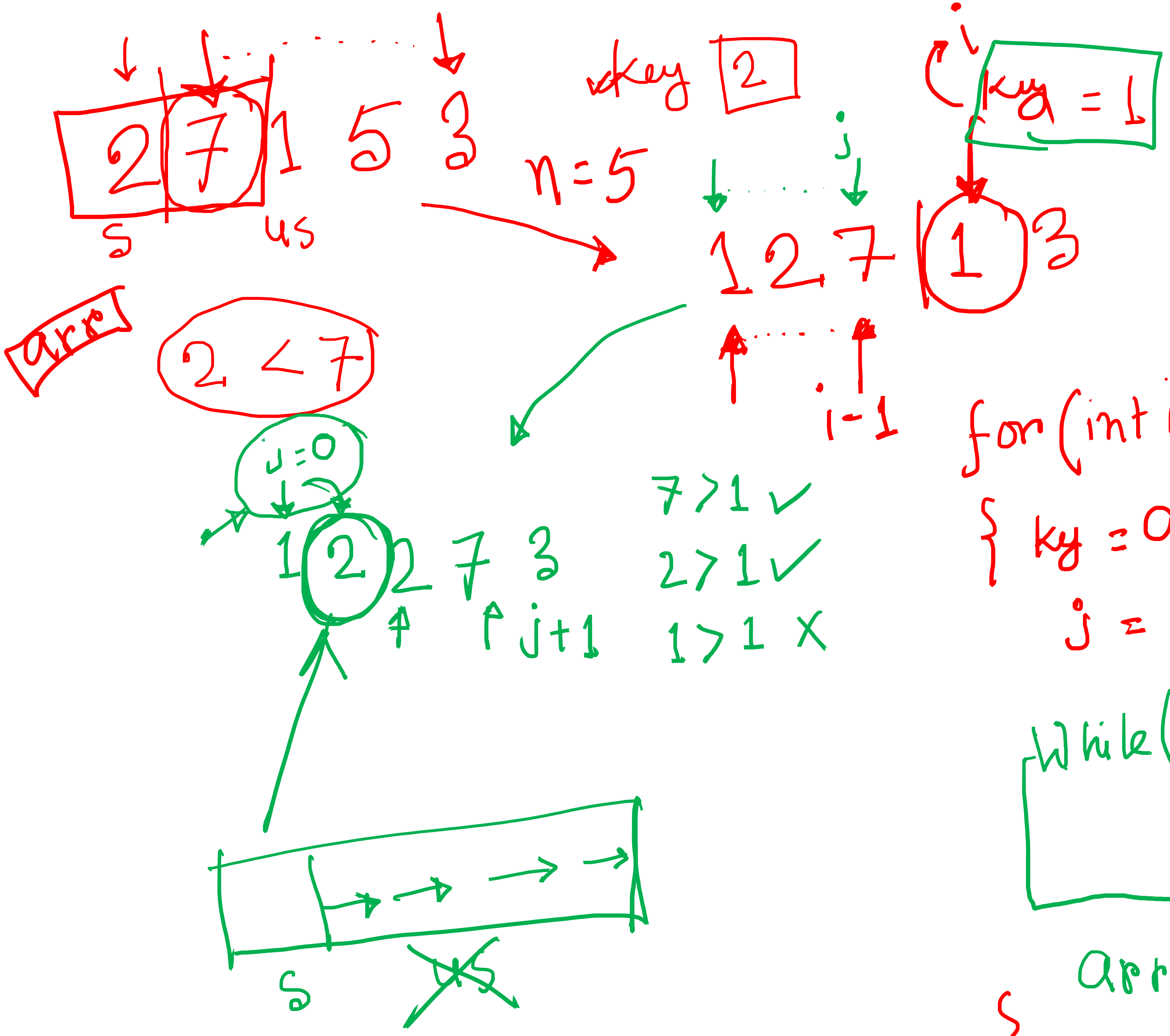
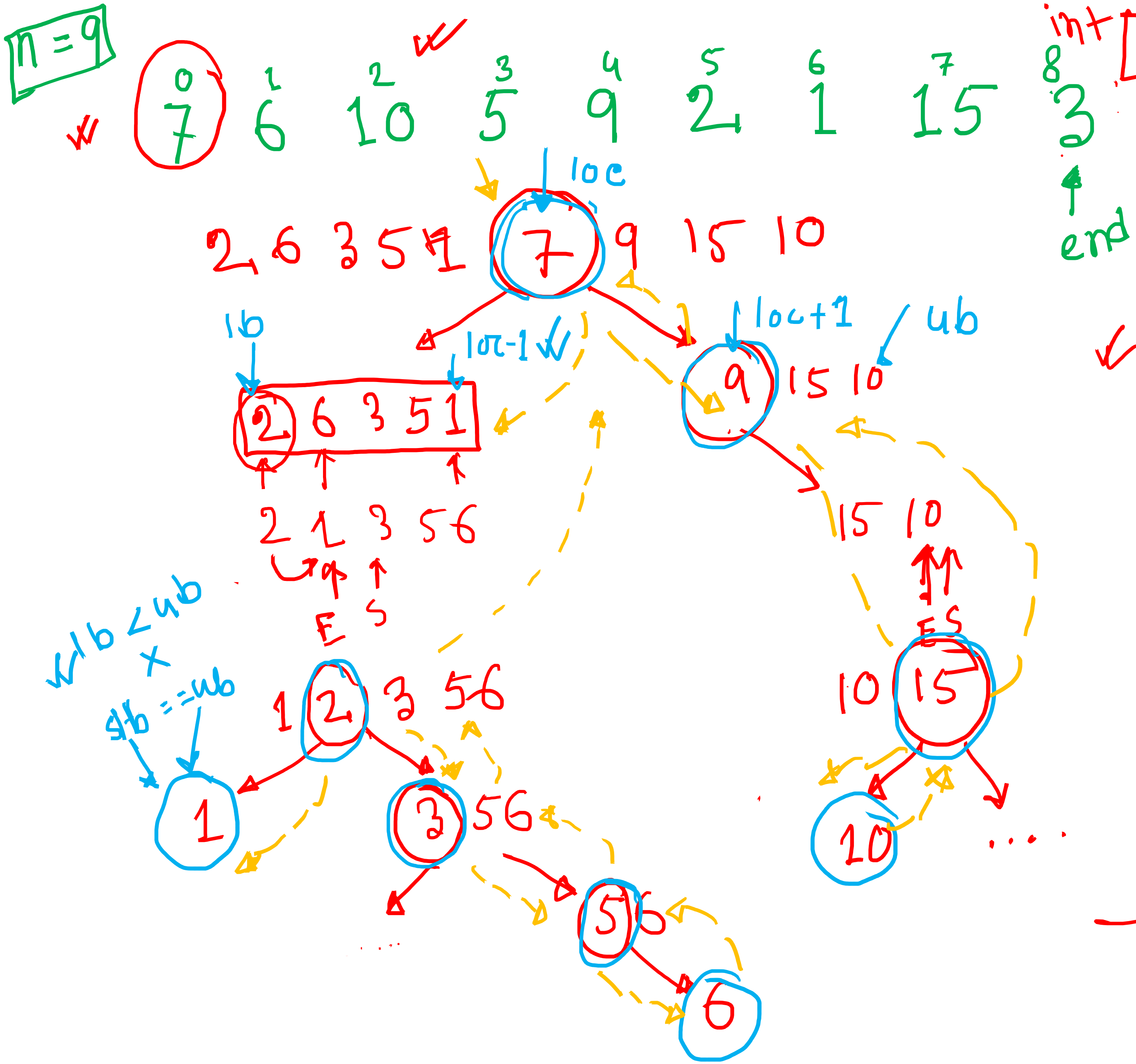
QUICK SORT

Lecturer Akib Zaman, Dept. of CSE, UIU



arr[j] > key

```
for(int i=1 ; i< n; i++)
{ ky = arr[i]
    j = i-1
  While(arr[j]>key && j>=0)
        arr[j+1] = arr[j]
         j--;
    arr[j+1] = key
}
```



```
int Partition(arr, lb, ub)
{
    int pivot = lb;
    int start = lb;
    int end = ub;
    While (start < end)
    [
        While (arr[start] <= arr[pivot]
        [   start++;
        while (arr[end] > arr(pivot)
        [   end--;
        if (start < end)
        [   swap(arr[start], arr[end])
    ]
    swap(arr[end, arr(plvot)
    return end;
}
```

```
                  ↓0   ↓8
QuickSort (arr, lb , ub)
{ if (lb<ub)
   { int loc = partition(arr, lb, ub)
        QuickSort(arr, lb, loc-1)
        QuickSort(arr, loc+1, ub)

}
```

```
len n = 9
```

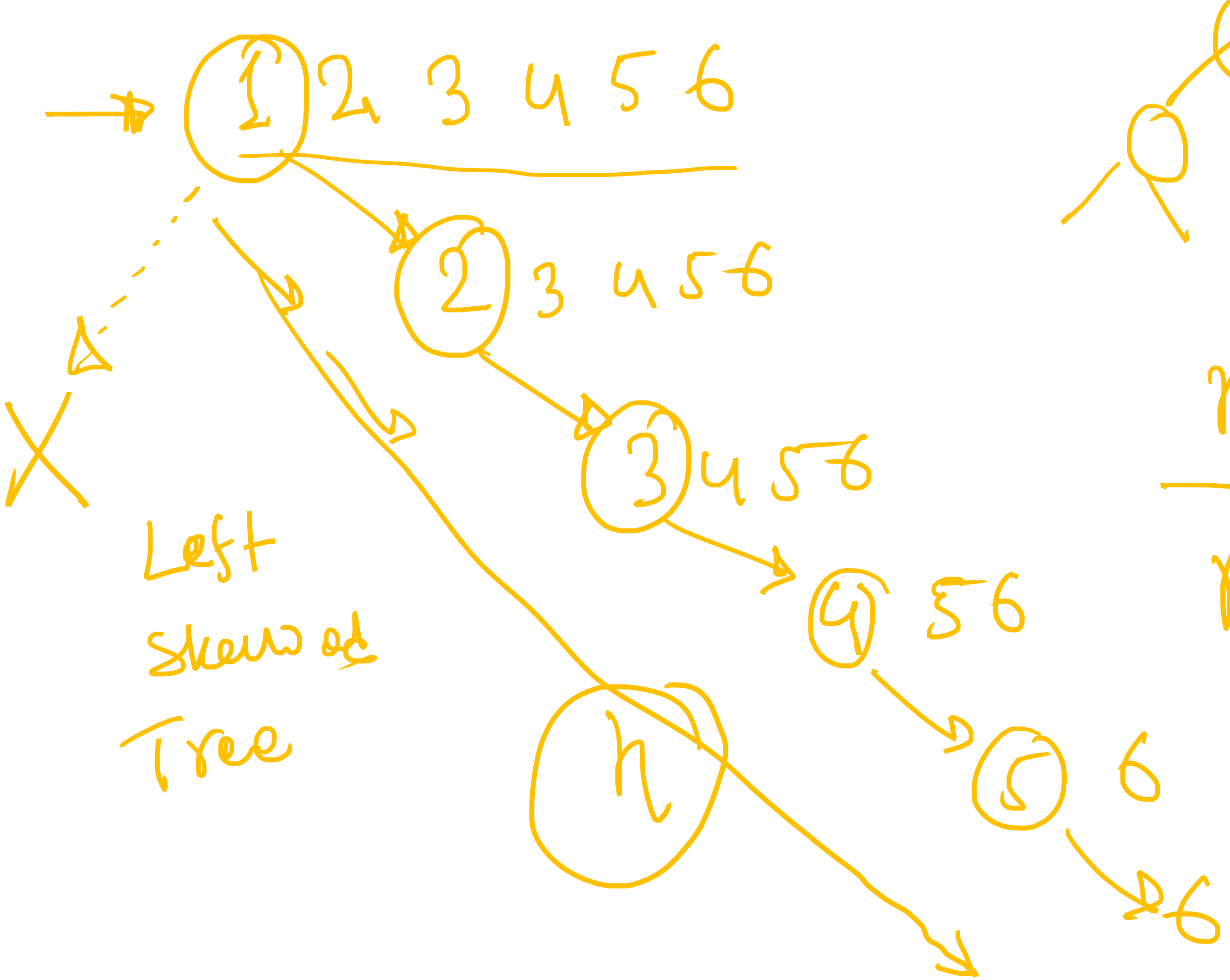
```
int Main( ) {   arr[ ] = {
                len (  )                 }
                Quicksort(arr, 0, 8)
}
```

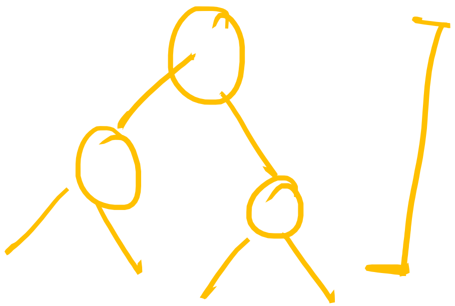
MergeSort : Extra array | Space complexity ↑ X

Worst case ⟶ $(n \log n)$

Quick sort : No extra array | space complexity ↓ ✓

worst case ⟶ n^2





Depth ↓ n/2 ✓

$$\frac{n\,[\log_2 n] \checkmark}{n\,[n]}$$

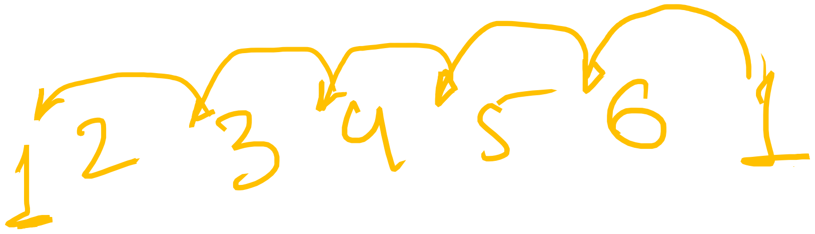
$\boxed{n^2}$ ✓ → Worst Case

Asending → There is no violations

1 2 3 4 5 6 → Violation

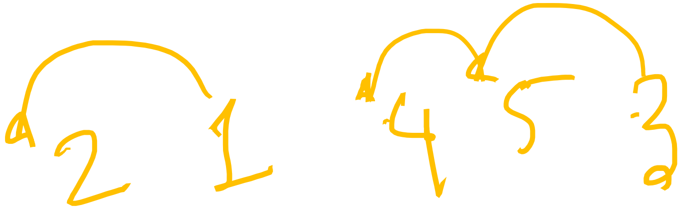
1	2	3	4	5	6
0	0	0	0	0	0

$\Sigma = 0$ ✓



$\Sigma = 5$

1	2	3	4	5	6
5	0	0	0	0	0



2	1	4	5	3
0	1	0	0	2

$\Sigma = 2$

✓

2

5

5 4 3 2 1

5

1 3 2 5 2

10 3

$$\sum = \begin{matrix}4 & 3 & 2 & 1\\ 1 & 2 & 3 & 4\end{matrix} = 10$$

$$\sum = \begin{matrix}1 & 3 & 2 & 5 & 2\\ 0 & 0 & 1 & 0 & 2\end{matrix} = 3$$

Recursion

+ Div -Conquer } (Lec 3 & 4)

+ Basic Topic (Lec 1 & 2)