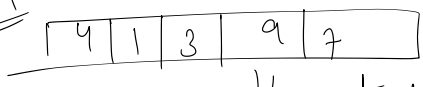
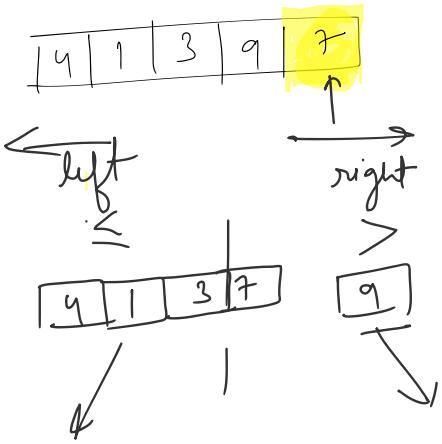
Day 20

11 December 2022 22:04

TC: 1



sort using Quick sort

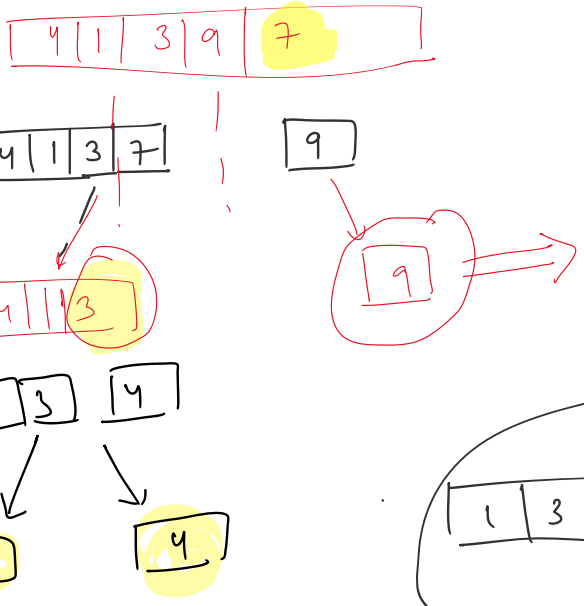


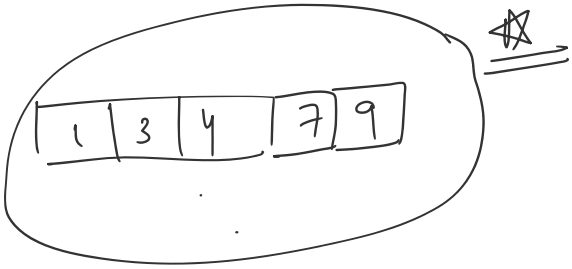
Quick Sort

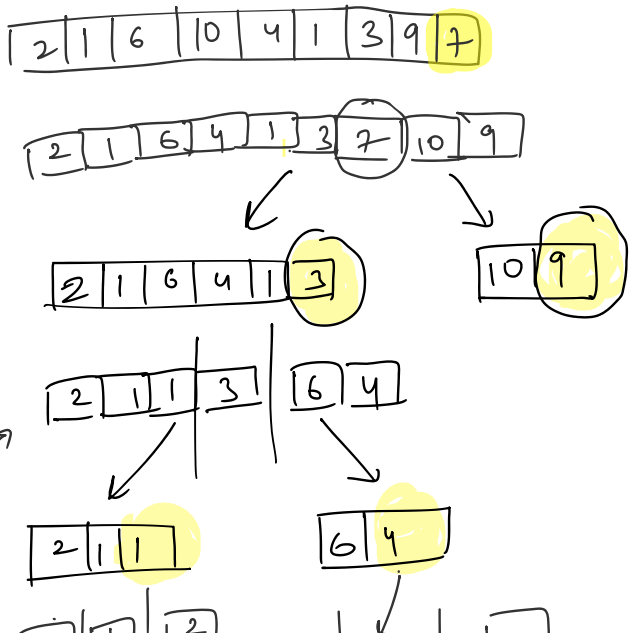
① Divide and Conquer Algorithm

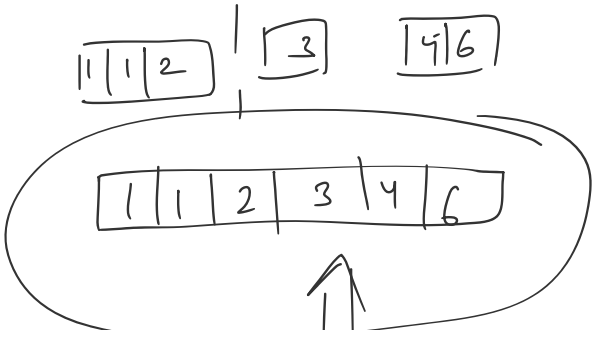
②

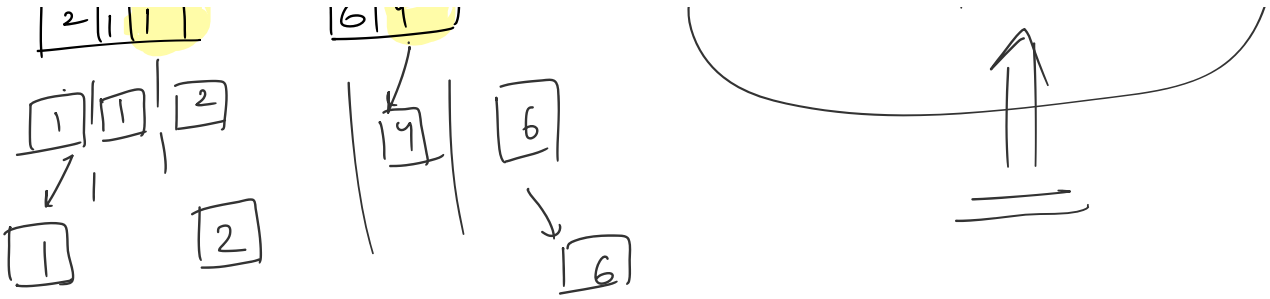
① pivot





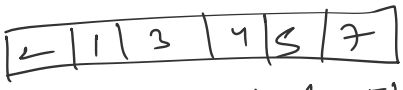




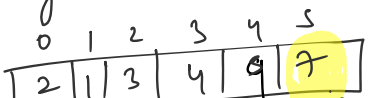


① last value. As pivot assume कर रहा है ||

② less than or equal to pivot को left side of array में और greater than को right.



① Divide Single Element नहीं हो जाता

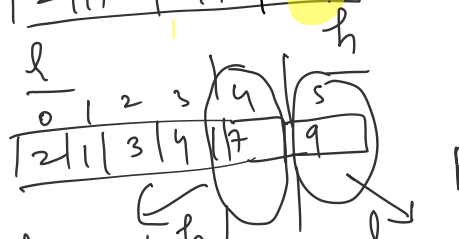


★ Auxiliary space
★ Inplace Algorithm
Quick sort

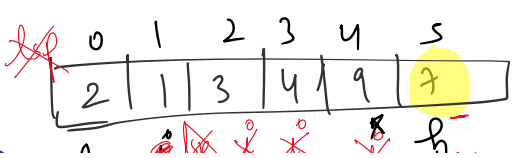
if (l < h) {

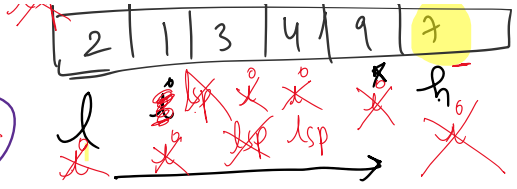
partition_function → pivot_index

partitioning function



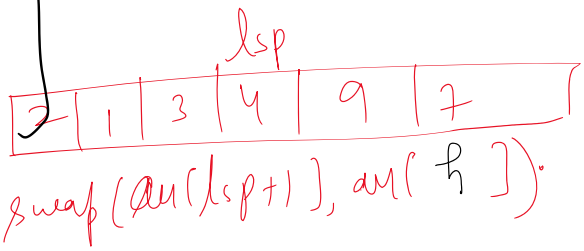
}



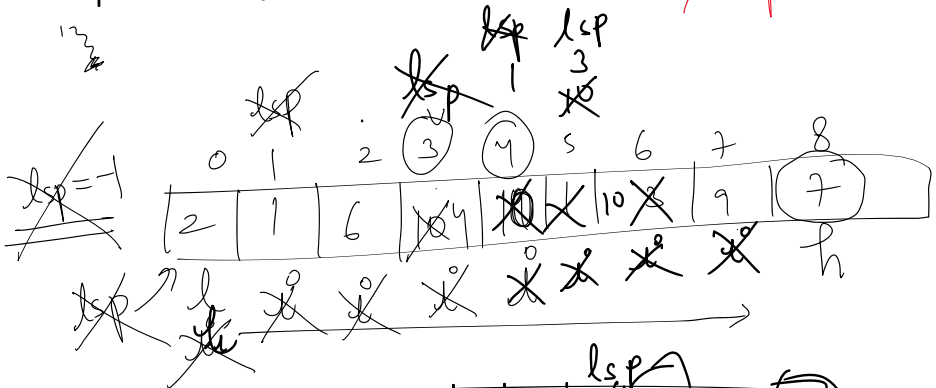


0 — lsp
↓
less than or equal pivot element

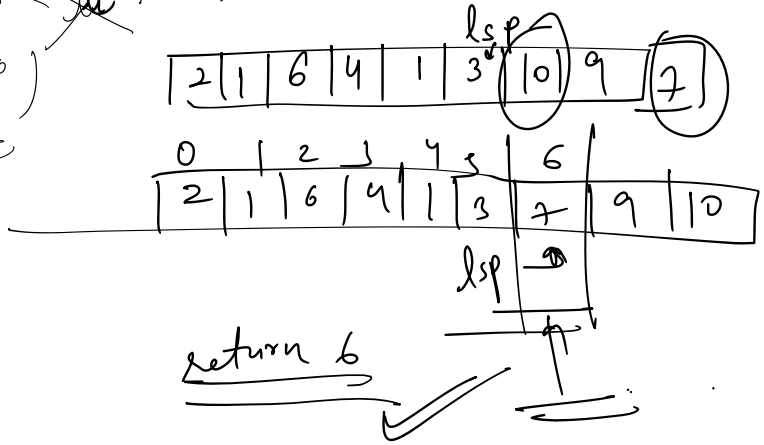
lsp = low - 1
for(int i = l; i<h; i++){ loop
if (arr[i] ≤ pivot_element)
{ swap (arr[lsp+1], arr[i]);
lsp = lsp+1;
}



swap (arr[lsp+1], arr[h]);



swap(lsp+1, i)



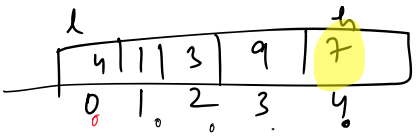
swap (lsp+1, h)

lsp+1

return 6

```
void quickSort(int arr[], int low, int high)
{
    // dividing condition
    if(low < high){
        int pivot_index = partition(arr,low,high);
        quickSort(arr,low,pivot_index-1);
        quickSort(arr,pivot_index+1,high);
    }
}
```

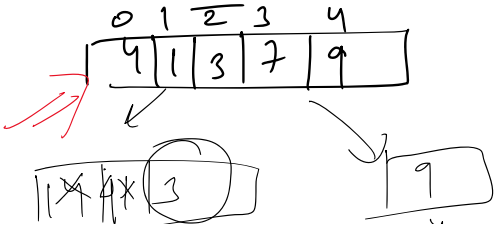
pi = 3



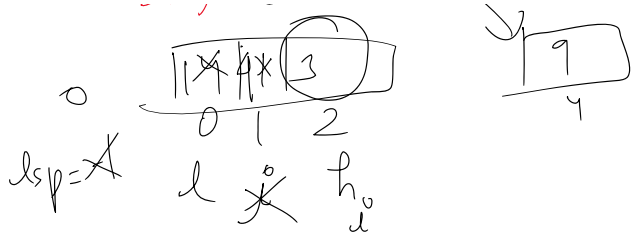
lsp = 2

```
public:
int partition (int arr[], int low, int high)
{
   int lsp = low - 1;
   int pivot_value = arr[high];
   for(int i = low;i < high;i++){
       if(arr[i] <= pivot_value){
           swap(arr[lsp+1],arr[i]);
           lsp = lsp + 1;
       }
   }
   swap(arr[lsp+1],arr[high]);
   return lsp+1;
}
```

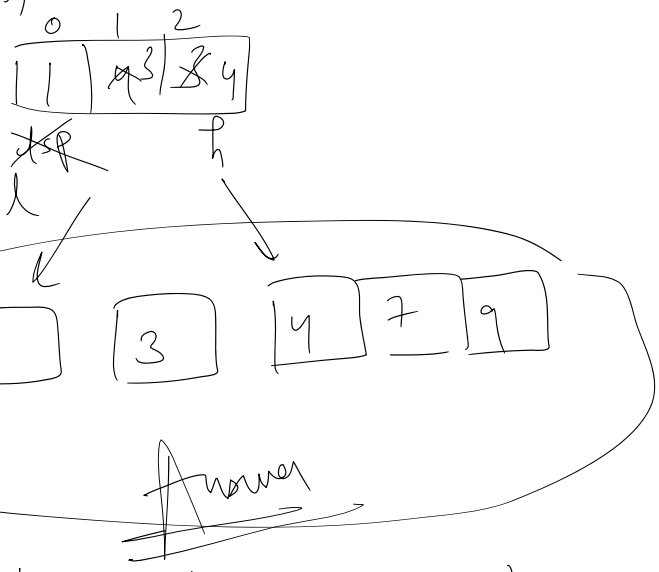
qs(0,4)
pi = 3
q(0,2)
qs(4,4)



pi = 3
q(0,2) qs(4,4)



swap (a[lsp+1], a[i])
a[0], a[1]



① Balanced Case (left || right) — O(nlogn)

② UnBalanced Case (left || —) — O(n²)

T.C — O(n²)

O(nlogn)

Random select pivot