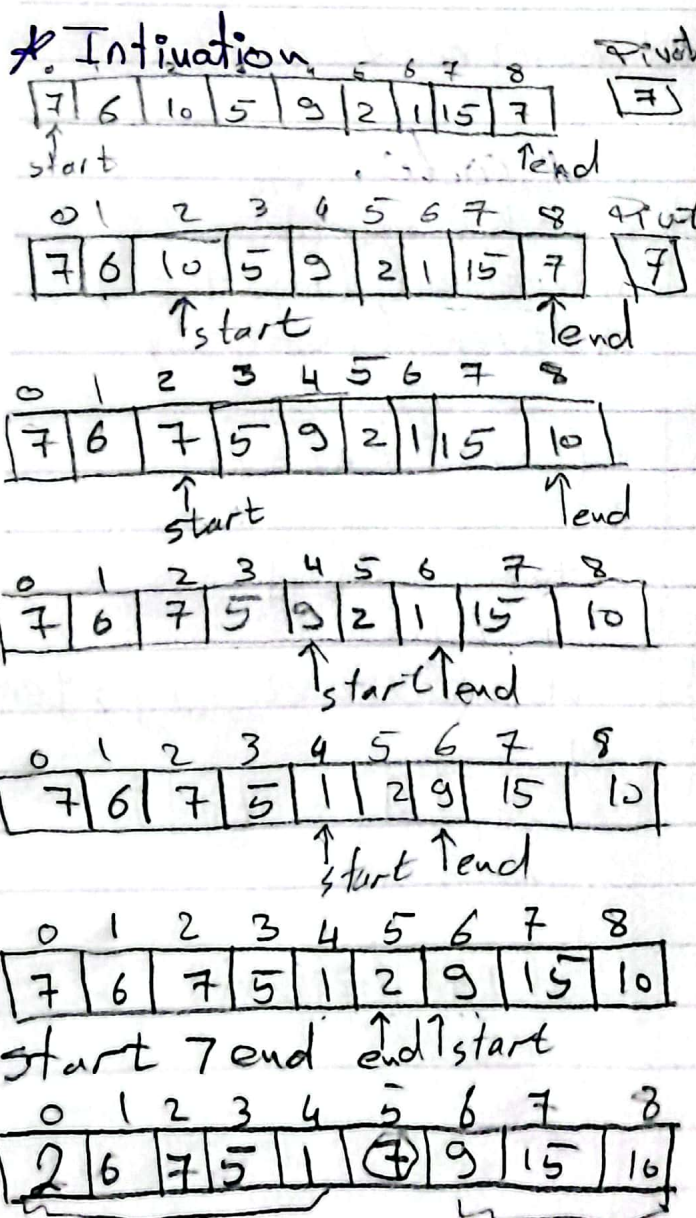# 2 Quick _ Sort. (from Divide & conquer Problems)

1) Is destructive :- Yes.

2) Is in Place :- Yes.

3) Time Complexity :- worst case = $\Theta(n^2)$
   Average case = $\Theta(n \log n)$

* Initiation

7	6	10	5	9	2	1	15	7

Pivot = 7

start ↑ ... Tend

	0	1	2	3	4	5	6	7	8	Pivot
	7	6	10	5	9	2	1	15	7	7

↑start ↑end

0	1	2	3	4	5	6	7	8
7	6	7	5	9	2	1	15	10

↑start ↑end

0	1	2	3	4	5	6	7	8
7	6	7	5	9	2	1	15	10

↑start↑end

0	1	2	3	4	5	6	7	8
7	6	7	5	1	2	9	15	10

↑start ↑end

0	1	2	3	4	5	6	7	8
7	6	7	5	1	2	9	15	10

start ↑end end↑start

0	1	2	3	4	5	6	7	8
2	6	7	5	1	⊕	9	15	10

* Code

```
Partition (Arr, Lb, Ub)
{
    Pivot = Arr[Lb];
    start = Lb;
    end = Ub;
    while(start < end)
    {
        while(Arr[start] <= Pivot)
        { start ++; }
        while(Arr[end] > Pivot)
        { end--; }
        if(start < end)
        { swap(Arr[start], Arr[end]);
        }
    }

    swap(Pivot, Arr[end]);
    return end;
}
```

code Cont' :-

```
Quick_Sort (Arr, Lb, Ub)
I  if ( Lb < Ub )
   {
        Loc = Partition ( A, Ub, Ub);
        Quick_Sort ( A, Lb, Loc-1);
        Quick_Sort ( A, Loc+1 , Ub);
   }
}


swap ( X, Y)
{ temp = X;
   X = Y;
     Y = temp;
}
```