Quick Sort

	0	1	2	3	4	5	6
arr	10	15	1	2	9	16	11

Choose any index value as 'pivot' element, here we will select value at index 0 i.e. $arr[0] = 10$

$$\therefore pivot = 10$$

Partition 1	Pivot	Partition 2
1, 2, 9	10	15, 16, 11
val ≤ pivot		val > pivot

1	2, 9

pivot   val > pivot

2	9

piv   val > pivot

11	15	16

val ≤ piv   pivot   val > piv
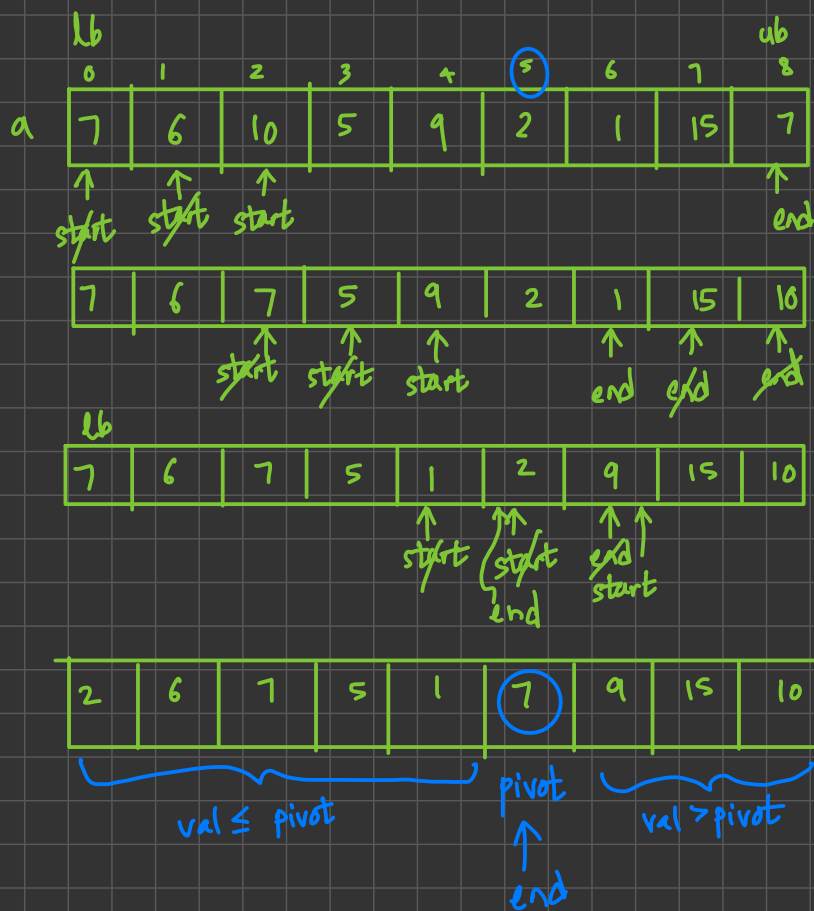
lb=0, ub=8
pivot = a[0] = 7
start = lb
end = ub
while (start < end)
  ① while (a[start] ≤ pivot)
        start++
  ② while (a[end] > pivot)
        end--
  ③ if (start < end)
        swap(a[start], a[end])

swap(a[lb], a[end])

return end

```
int  partition ( a , lb, ub)      [0   8]
{  pivot = a[lb];
   start = lb;
   end = ub;
   while (start < end )
   {
      ① while (a[start] <= pivot)
             start++;
      ② while (a[end] > pivot)
             end--;
      ③ if (start < end)
             swap( a[start], a[end]);  }

   }

   swap (a[lb], a[end])
   return end;
}
```

```
quick sort ( a, lb, ub)
{
   if (lb < ub)
   {  loc = partition(a, lb, ub)
      quicksort (a, lb, loc-1)
      quicksort (a, loc+1, ub)
   }
}
```