Challenge: Implement Quicksort

The quickSort function should recursively sort the subarray array[p..r].

- If the subarray has size 0 or 1, then it's already sorted, and so nothing needs to be done.

- Otherwise, quickSort uses divide-and-conquer to sort the subarray.

The divide step should partition the array, the conquer step should recursively quicksort the partitioned subarrays, and the combine step should do nothing.

Java | Python | C++ | JS

```
# This function partitions given array and returns
# the index of the pivot.
def partition(array, p, r):
  # Dont worry about this function. It's intentionally written with bad variable names
  # as you will implement it yourself in next challenge
  e=array
  t=p
  n=r
  def swap(e,t,n):
        r=e[t]
        e[t]=e[n]
        e[n]=r
  i=t
  s=t
  while s<n:
        if e[s]<=e[n]:
          swap(e,s,i)
          i = i + 1
        s = s + 1

  swap(e,n,i)
  return i

def quickSort(array, p, r):
  # Write method here

  return
```