Section 7.1 – Description of quicksort

7.1-1 Using Figure 7.1 as a model, illustrate the operation of PARTITION on the array $A = \langle 13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11 \rangle$.

13	19	9	5	12	8	7	4	21	2	6	11
13	19	9	5	12	8	7	4	21	2	6	11
13	19	9	5	12	8	7	4	21	2	6	11
9	19	13	5	12	8	7	4	21	2	6	11
9	5	13	19	12	8	7	4	21	2	6	11
9	5	13	19	12	8	7	4	21	2	6	11
9	5	8	19	12	13	7	4	21	2	6	11
9	5	8	7	12	13	19	4	21	2	6	11
9	5	8	7	4	13	19	12	21	2	6	11
9	5	8	7	4	13	19	12	21	2	6	11
9	5	8	7	4	2	19	12	21	13	6	11
9	5	8	7	4	2	6	12	21	13	19	11
9	5	8	7	4	2	6	11	21	13	19	12

7.1-2 What value of q does PARTITION return when all elements in the array $A = [p, \ldots, r]$ have the same value? Modify PARTITION so that $q = \lfloor (p + r)/2 \rfloor$ when all elements in the array $A[p, \ldots, r]$ have the same value.

It will return $q = r$. We can update PARTITION to split elements that are equal to the pivot on both sides as follows:

(a) Count the number of elements y such that $y = x$ and set this value to c;

(b) Subtract the final pivot index by $\lfloor c/2 \rfloor$.

The updated pseucode is stated below.

```
   Partition-Improved(A, p, r)
1  | x = A[r]
2  | i = p − 1
3  | c = 0
4  | for j = p to r − 1 do
5  | |  if A[j] ≤ x then
6  | |  |  if A[j] == x then
7  | |  |  |  c = c + 1
8  | |  |  i = i + 1
9  | |  |  exchange A[i] with A[j]
10 | exchange A[i + 1] with A[r]
11 | return (i + 1) − ⌊c/2⌋
```

7.1-3 Give a brief argument that the running time of PARTITION on a subarray of size n if $\Theta(n)$.

The **for** loop of lines 3–6 iterates $n - 1$ times and each iteration does a constant amount of work. Thus, it is $O(n)$.

7.1-4 How would you modify QUICKSORT to sort into nonincreasing order?

We just need to update the condition

$$A[j] \leq x,$$

to

$$A[j] \geq x.$$