CSE 2320 **Name: HOUDA BELEFQIH**

Homework 5 (80 points)

P1. (7 points) CLRS, 8.2-1, page 196: Using Figure 8.2 as a model, illustrate the operation of Count_Sort on the array A = <6,0,2,0,1,3,4,6,1,3,2>. (As in CLRS, make new copies for C and B to show how the first 3 elements are placed in B. After that, show the final version of B).

A : Original array :

1	2	3	4	5	6	7	8	9	10	11
6	**0**	**2**	**0**	**1**	**3**	**4**	**6**	**1**	**3**	**2**

C : Count array

0	1	2	3	4	5	6
2	**2**	**2**	**2**	**1**	**0**	**2**

→

0	1	2	3	4	5	6
2	**4**	**6**	**8**	**9**	**9**	**11**

LEFT ARRAY: B ***RIGHT ARRAY: C***

1	2	3	4	5	6	7	8	9	10	11
					2					

0	1	2	3	4	5	6
2	**4**	**5**	**8**	**9**	**9**	**11**

1	2	3	4	5	6	7	8	9	10	11
					2		**3**			

0	1	2	3	4	5	6
2	**4**	**5**	**7**	**9**	**9**	**11**

1	2	3	4	5	6	7	8	9	10	11
			1		**2**		**3**			

0	1	2	3	4	5	6
2	**3**	**5**	**7**	**9**	**9**	**11**

1	2	3	4	5	6	7	8	9	10	11
			1		**2**		**3**			**6**

0	1	2	3	4	5	6
2	**3**	**5**	**7**	**9**	**9**	**10**

1	2	3	4	5	6	7	8	9	10	11
			1		**2**		**3**	**4**		**6**

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

1	2	3	4	5	6	7	8	9	10	11
			1		2	3	3	4		6

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

1	2	3	4	5	6	7	8	9	10	11
		1	1		2	3	3	4		6

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

1	2	3	4	5	6	7	8	9	10	11
	0	1	1		2	3	3	4		6

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

1	2	3	4	5	6	7	8	9	10	11
	0	1	1	2	2	3	3	4		6

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

1	2	3	4	5	6	7	8	9	10	11
0	0	1	1	2	2	3	3	4		6

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

Final result :

1	2	3	4	5	6	7	8	9	10	11
0	0	1	1	2	2	3	3	4	6	6

0	1	2	3	4	5	6
0	2	4	6	8	9	9

P2. (7 points) Is Quick_Sort (as given in CLRS, page 171) stable?
If yes, prove it.
If no, give an example array, A, (however small or big), sort it with Quick_Sort, and show what the algorithm does that makes it not stable. Use the original array and the final, sorted array to base your proof (do not base your proof on a partially sorted array).

No it is not stable when picking each time last element of array/subarray as a pivot.

Example :

	0	1	2	3	4
Original array	**6**	**5a**	**5b**	**2**	**4**
After 1st execution	2	4	5b	6	5a
After 2nd execution	2	4	5b	5a	6
Final array	2	4	**5b**	**5a**	6

We started with 5a then 5b and ended up at the end with the 5's in reverse order. The swapping of the pivot at the end of the partition function makes it unstable.

P3. (14 points) Given the array A = < 8, 6, 9, 2, 7, 1, 5, 10, 6 >

a) (7 points) Using Figure 7.1, CLRS page 172, as a model, show **the execution** of the Partition function. Show the <= partition by **circling the last element of it** and show the > partition by putting **a square around the last element of it**.

	0	1	2	3	4	5	6	7	8
Original array	**8**	**6**	**9**	**2**	**7**	**1**	**5**	**10**	**6**
	[8]	6	9	2	7	1	5	10	6
	(6)	[8]	9	2	7	1	5	10	6
	(6)	8	[9]	2	7	1	5	10	6
	6	(2)	9	[8]	7	1	5	10	6
	6	(2)	9	8	[7]	1	5	10	6
	6	2	(1)	8	7	[9]	5	10	6
	6	2	1	(5)	7	9	[8]	10	6
	6	2	1	(5)	7	9	8	[10]	6
	6	2	1	(5)	**6**	9	8	10	[7]

a) (7 points) Show the **array** A, **circle the pivot** and write what is **returned** by the Partition function **after** each call to the Partition function during the execution of the Quick-Sort algorithm from CLRS page 171.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Returned by partition**
Original array	**8**	**6**	**9**	**2**	**7**	**1**	**5**	**10**	**6**	**-**
After 1st execution	6	2	1	5	(6)	9	8	10	7	4
After 2nd execution	2	1	(5)	6	6	9	8	10	7	2
After 3rd execution	(1)	2	5	6	6	9	8	10	7	0
After 4th execution	1	2	5	6	6	(7)	8	10	9	5
After 5th execution	1	2	5	6	6	7	8	(9)	10	7

SubArray being processed: cells in blue.

P4. Test cases in *student_tests.txt* :

1- Random array : N=6 ; A= {2, 9, 1, 3, 6, 7 }

2- Array with one element only (N=1) : N=1 ; A= {5}

3- Array with one value for all elements : N=5 ; A= {-7, -7, -7, -7, -7}

4- Already sorted array: N= 4 ; A= {2, 3, 7, 80}

5- Empty array : N=0

3 and 4 show that QuickSort is not adaptive.