CSCI 532 – Algorithm Design
Assignment 3

Name: Badarika Namineni CWID: 50233279

Question 1: Using Figure 7.1 as a model, illustrate the operation of PARTITION on the array A (13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11).

Solution:

1) A (13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11)

a)	i P,j 13	19	9	5	12	8	7	4	21	2	6	r 11
b)	i, P 13	j 19	9	5	12	8	7	4	21	2	6	r 11
c)	i, P 13	19	j 9	5	12	8	7	4	21	2	6	r 11

Swap i+1, j

	i P 9	19	j 13	5	12	8	7	4	21	2	6	r 11
d)	P 9	i 5	13	j 19	12	8	7	4	21	2	6	r 11
e)	P 9	i 5	13	19	j 12	8	7	4	21	2	6	r 11
f)	P 9	5	i 8	19	12	j 13	7	4	21	2	6	r 11
g)	P 9	5	8	i 7	12	13	j 19	4	21	2	6	r 11
h)	P 9	5	8	7	i 4	13	19	j 12	21	2	6	r 11
i)	P 9	5	8	7	i 4	13	19	12	j 21	2	6	r 11
j)	P 9	5	8	7	4	i 2	19	12	21	j 13	6	r 11
k)	9	5	8	7	4	2	i 6	12	21	13	j 19	r 11

Exchange i+1 to r

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

Question 2: Give a brief argument that the running time of PARTITION on a subarray of size n is $\Theta(n)$.

Solution:

Let us consider the algorithm for the partition

```
PARTITION (A, p, r)
x = A[r]
I = p-1
for (j = p to r-1)
{
if(A[j]<=x)
{
i = i+1
swap A[i] with A[j]
}
}
Swap A[i+1] with r
Return i+1
```

The for loop makes exactly $r - p$ iterations, each of which takes at most constant time. Since $r - p$ is the size of the subarray, PARTITION takes at most time proportional to the size of the subarray it is called on. Hence it takes $\Theta(n)$.

Question 3: Use the substitution method to prove that the recurrence $T(n) = T(n-1) + \Theta(n)$ has the solution $T(n) = \Theta(n2)$

Solution:

This is used in worst-case scenario where one side of the partition may contain 0 elements. In this case, we get $T(n) = T(n-1) + \Theta(n)$

By using substitution method,

We can write $T(n) = T(n-1) + cn$

$T(n) = T(n-2) + c(n-1) + cn$ [Where $T(n-1) = T(n-2) + c(n-1)$]

$T(n) = T(n-3) + c(n-2) + c(n-1) + cn$ [Where $T(n-2) = T(n-3) + c(n-2)$]

We get, $T(n) = c+c2+c3\ldots\ldots\ldots.. +cn =$ $\mathbf{\Theta(n^2)}$

Question 4: Show that in the recurrence $\max (T(q) + T(n-q-1)) + \Theta(n)$,

$0 \le q \le n-1$

$T(n) = \Omega(n2)$

Solution:

$$T(n) = \max_{0 \le q \le n-1} (T(q)+T(n-q-1)) + \Theta(n)$$

$= T(n-1) + \Theta (n)$ [Where $q = 0$]

$= \Theta (n^2)$

Since it is $\Theta (n^2)$, it can also be $\mathbf{T(n) = \Omega(n^2)}$

Question 5: Show that quicksort's best-case running time is Ω(nlgn)

Solution:
In the best-case, partition produces 2 sub-problems each of size not more than n/2, since one is of size [n/2] and one of size ([n/2]-1). In this case, quicksort runs much faster. The recurrence for the running time is T(n) = 2T(n/2) + Θ(n)

By using master's theorem, the solution for this recurrence is Θ(nlogn). Because it is Θ(nlogn), the best-case running time is Ω(nlogn).

Question 6: Show that RANDOMIZED-QUICKSORT's expected running time is Ω(nlgn)

Solution:
Considering the lower bound of the expected running time from Lemma theorem, we get

E[X] = $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1}$

$\sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1}$ Where k>=1

$\sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{2k}$

$\sum_{i=1}^{n-1} \Omega(logn)$

We get, **E[X] = Ω(nlogn)**