Design and Analysis of Algorithms : Assignment 1

1) Problem 3-2 from CLRS

A	B	O	o	Ω	ω	Θ
$\log^k n$	n^ϵ	Y	Y	n	n	n
n^k	c^n	Y	Y	n	n	n
$\sqrt{n}$	$n^{\sin n}$	-	-	-	-	-
2^n	$2^{n/2}$	n	n	Y	Y	n
$n^{\log c}$	$c^{\log n}$	Y	n	Y	n	Y
$\log n!$	$\log n^n$	Y	n	Y	n	Y

2) Problem 4.2-5 from CLRS

The answer is $\Theta(n \log n)$.

3) Problem 4-4 (a, c, f, j) from CLRS

a. $\Theta(n^{\log 3})$
c. $\Theta(n^{2.5})$
f. $\Theta(n)$
j. $\Theta(n \log \log n)$

4) Suppose we are given an array A[1 ... n] with the special property that A[1] $\geq$ A[2] and A[n - 1] $\leq$ A[n]. We say that an element A[i] is a local minimum if A[i - 1] $\geq$ A[i] and A[i + 1] $\geq$ A[i]. For example, there are six local minima (underlined) in the following array:

$$9, \underline{7}, 7, 2, \underline{1}, 3, 7, 5, \underline{4}, 7, \underline{3}, \underline{3}, 4, 8, \underline{6}, 9$$

We can obviously find a local minimum in $O(n)$ time by scanning through the array. Given and analyze an $O(\log n)$ time algorithm for the same.

5) Problem 7-3 from CLRS

The correctness can be proved by induction. Assume it is true for 1. Then for a call Stooge(A, i, j), assume it sorts any array of size less than $|j - i|$ and use that to show that it works for (A, i, j).

Algorithm 1 LocalMin

```
procedure LocalMin(A, i, j)                         ▷ Finds the local min in A[i ... j]
    if (j − i) ≤ 1 then
        return i
    end if
    mid ← (i+j)/2
    if A[mid − 1] ≥ A[mid] and A[mid] ≤ A[mid + 1] then   ▷ If mid is the local min
        return mid                                  ▷ Return it
    end if
    if A[mid − 1] < A[mid] then
        return LocalMin(A, i, mid)                  ▷ Search for min in the lower half
    else                                            ▷ Definitely A[mid] > A[mid + 1]
        return LocalMin(A, mid, j)                  ▷ Search for min in the upper half
    end if
end procedure
```

The recurrence is

$$
\begin{aligned}
T(n) &= 3T\left(\frac{2}{3}n\right) + \Theta(1) \\
T(n) &= 3T\left(\frac{2}{3}n\right) + 1 \\
&\vdots \\
&= 3^k T\left(\left(\frac{2}{3}\right)^k n\right) + \sum_{i=0}^{k-1} 3^i \\
&= \sum_{i=0}^{k} 3^i \qquad\qquad n \approx \left(\frac{2}{3}\right)^k, k = \log_{(3/2)} n \\
&= \frac{3^{\log_{(3/2)} n+1} - 1}{3 - 1} \\
&= \Theta(3^{\log_{(3/2)} n}) = \Theta(n^{\log_{(3/2)} 3}) \\
&= \Theta(n^{2.7095})
\end{aligned}
$$

6) Problem 8.3-2 from CLRS

Use Radix sort. We can use Lemma 8.4 with $b = \log(n^2) = 2\log n$, $r = \log n$. Then by using radix sort, we can sort it in time $\Theta((b/r)(n + 2^r)) = \Theta(n)$.