National University of Computer and Emerging Sciences, Lahore Campus

Course:	Design & Analysis of Algorithms	Course Code:	CS-2009
Program:	BS (Computer Science)	Semester:	Fall 2022
Duration:	60 Minutes	Total Marks:	21
Paper Date:	21-Mar-22	Weight	N/A
Section:	N/A	Page(s):	8
Exam:	Midterm	~~Roll No~~.	*Solution*

Instruction/Notes:

Weightage of the exam is Section Specific (i.e. for each section the weightage would be as per the announcement done in that regard).

Do NOT un-staple your exam, otherwise it might be cancelled.

Ample space is provided for rough work; NO EXTRA sheets will be provided.

Question 1: 3 Marks

Arrange the function $(1.5)^n, n^{100}, (log\ n)^3, \sqrt{n}\ log\ n, 10^n, (n!)^2$, and $n^{99} + n^{98}$ in a list so that each function is big-*O* of the next function.

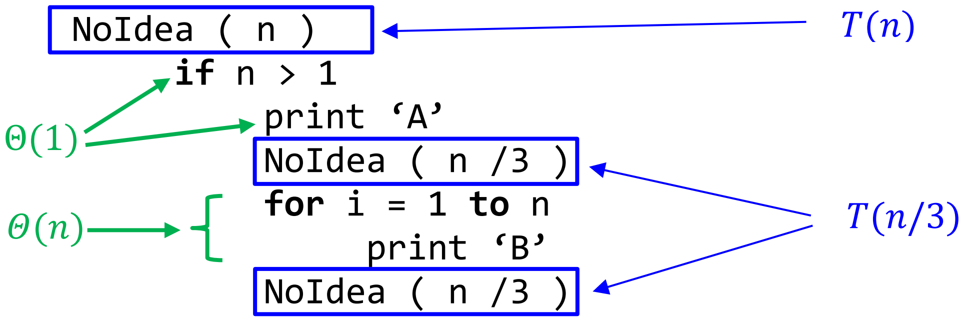
Answer:

$$(log\ n)^3 = O\left(\sqrt{n}\ log\ n\right) = O(n^{99} + n^{98}) = O(n^{100}) = O((1.5)^n) = O(10^n) = O((n!)^2)$$

Solution

Question 2: **4 Marks**

Consider the following algorithm



```
NoIdea ( n )
        if n > 1
           print 'A'
           NoIdea ( n /3 )
           for i = 1 to n
                print 'B'
           NoIdea ( n /3 )
```

Write down the time-complexity of the above algorithm in recursive form (*only write the recursive equation, you are not required to solve the recurrence*).

Answer:

$$T(n) = \begin{cases} 2T\left(\frac{n}{3}\right) + \Theta(n) & \text{if } n > 1 \\ \Theta(1) & \text{otherwise} \end{cases}$$

Question 3: **5 Marks**

What would be the output of MyAlgo(A,1,10), where A is given as below:

$$A = \langle 3,7,2,6,1,8,5,4,0,9 \rangle$$

Note: Show complete working to claim for any **partial credit**.

```
MyAlgo(A,p,r)

if $p < r$
    q=DoSomething$(A, p, r)$
    MyAlgo$(A, p, q - 1)$
    MyAlgo$(A, q, r - 1)$


//---------------------------------------------------//

DoSomething$(A, p, r)$

$x$=A[r]
i=p-1
for j=p to r-1
    if $A[j] \leq x$
        i=i+1
        exchange A[i] with A[j]
return i+1
```

Answer: $A = \langle 3,2,1,6,7,8,5,4,0,9 \rangle$

Working on the Next page

Solution

```
MyAlgo(A, 1, 10)               //Call 1
```

$A = \langle 3,7,2,6,1,8,5,4,0,9 \rangle$

```
    MyAlgo(A, 1, 9)          //Call 1.1
```

$A = \langle 3,7,2,6,1,8,5,4,0,9 \rangle$

```
        MyAlgo(A, 1, 0)    //Call 1.1.1
        MyAlgo(A, 1, 8)    //Call 1.1.2
```

$A = \langle 3,2,1,6,7,8,5,4,0,9 \rangle$

```
            MyAlgo(A, 1, 3) //Call 1.1.2.1
```

$A = \langle 3,2,1,6,7,8,5,4,0,9 \rangle$

```
                MyAlgo(A, 1, 0) //Call 1.1.2.1.1
                MyAlgo(A, 1, 2) //Call 1.1.2.1.2
```

$A = \langle 3,2,1,6,7,8,5,4,0,9 \rangle$

```
                    MyAlgo(A, 1, 0) //Call 1.1.2.1.2.1
                    MyAlgo(A, 1, 1) //Call 1.1.2.1.2.2
            MyAlgo(A, 4, 7) //Call 1.1.2.2
```

$A = \langle 3,2,1,6,7,8,5,4,0,9 \rangle$

```
                MyAlgo(A, 4, 3) //Call 1.1.2.2.1
                MyAlgo(A, 4, 6) //Call 1.1.2.2.2
```

$A = \langle 3,2,1,6,7,8,5,4,0,9 \rangle$

```
                    MyAlgo(A, 4, 5) //Call 1.1.2.2.2.1
```

$A = \langle 3,2,1,6,7,8,5,4,0,9 \rangle$

```
                      MyAlgo(A, 4, 4) //Call 1.1.2.2.2.1.1
                      MyAlgo(A, 5, 4) //Call 1.1.2.2.2.1.2
                    MyAlgo(A, 6, 5) //Call 1.1.2.2.2.2
    MyAlgo(A, 10, 9)         //Call 1.2
```

Question 4: **[1+6] Marks**

Let $A[1 \dots n]$ be an array of n numbers such that

$$A[i] \leq A[i+2] \qquad \forall 1 \leq i \leq n-2$$

i) Write an instance of A for $n = 15$ (your array must not be already sorted).

$$A = \langle 2,1,4,3,6,5,8,7,10,9,12,11,14,13,15 \rangle$$

ii) Give an algorithm that sorts A in non-decreasing order i.e. the output of your algorithm should be the array A such that

$$A[i] \leq A[i+1] \qquad \forall 1 \leq i \leq n-1$$

Your algorithm must run in $\boldsymbol{O(n)}$ worst-case time.

Hint: Some modification to the Merge procedure could be helpful in this scenario.

Solution on the Next page

Solution

```
$n_1=\left\lceil \frac{n}{2} \right\rceil$
$n_2=\left\lfloor \frac{n}{2} \right\rfloor$
Let L[1…$n_1$+1] and R[1…$n_2$+1] be new arrays
i=1
j=1
for i=1 to $n_1$
    L[i]=A[j]
    j=j+2
A[$n_1$+1]=∞
i=1
j=2
for i=1 to $n_2$
    R[i]=A[j]
    j=j+2
A[$n_2$+1]=∞
i=1
j=1
for k=1 to n
    if L[i]≤R[j]
        A[k]=L[i]
        i=i+1
    else A[k]=R[j]
        j=j+1
```