Roll No. 21L-5421 Section 3F

National University of Computer and Emerging Sciences, Lahore Campus

Course:	Data Structures	Course Code:	CS 201
Program:	BS(CS)	Semester:	Fall 2022
Duration:	10 Minutes	Total Marks:	10
Paper Date:	5 Sep 2021	Exam	Quiz 1
Section:	F		

Instruction/Notes: Solve the exam on this question paper.

(10) Good!

Question: Consider the following program

i) Give an estimate of T(N). (Show your work and give a T(N) estimate for each line of code.)

```
void Func(){
    int count = 0;                                   1
    for(int i = 1; i < n; i=i+2 ){                   i=1 - 1 ; i<n - log n/2 + 1 ; i=i+2 - n/2
        for( int k = i; k > 0; k=k/2 )               int k=i - n/2 ; k>0 - (log2n+1) n/2 ; k=k/2 - (log2n) n/2
            count=count++;                           log2n (n/2)
    }

    int mul3=0;                                      1
    for(int j=0; j< n; j++)                          n+1
        if(j % 3 == 0)                               n
            for( int m = 1; m < j*j; m++ )           Σi^2
                mul3++;                              Σi^2
    }
}
```

i=1 n=6
i=1,3,5
i=1

ii) Find the tight big Oh for the Best-case and Worst-case scenario. Explain in one line how you drive it.

ii) Best Case O(n)

Worst Case $O\,\Sigma i^2 \Rightarrow \frac{n(n+1)(2n+1)}{6} = n^3$

Best Case would still require a loop to check all values till n if they are divisible by 3 however worst case would be that every number is divisible of 3.

i) Best Case $O(\log_2 n/2)$

Rough

i	k	# of times
1	1	1
3	3	1
5	5, 2.5, 1.25	2

i	k	# of t
1	1, 0.5, 0.25	

m	m<j*j	j	m++
1	m<0		
j 1	m 1		m++
j2	m 1,2,3,4		
j3	1-9		

$\log_2 n\,(n/2)$

Course:	Data Structures	Course Code:	CS 201
Program:	BS(CS)	Semester:	Fall 2022
Duration:	10 Minutes	Total Marks:	10
Paper Date:	5 Sep 2021	Exam	Quiz 1
Section:	F		

Instruction/Notes: Solve the exam on this question paper.

Question: Consider the following program

i) **Give an estimate of T(N).** *(Show your work and give a T(N) estimate for each line of code.)*

```
int val = 0;
for(int i = 1; i < n; i=i+2 ){
        for( int j = i; j > 0; j=j/3 )
                val++;
}

int no=0, k=0;
while (k< n/2){
        if(k % 2 == 0)
            for( int j = 1; j < k*k; j++ )
                no=no+j;
        k=k+2;
}
```

T(N) for each line

1

$i=1 \rightarrow 1$, $i<n \rightarrow n/2$, $i=i+2 \rightarrow n/2$

$j=i \rightarrow n/2$, $j>0 \rightarrow \frac{n}{2}\log_3 i$

Basically $\sum_{1}^{n} \log_3 i$ and i = odd

$\log_3 1 + \log_3 3 + \ldots \log_3 (n)$

$= n \log_3 n$

1

$n/4$

$n/4$

$\frac{n}{4} k^2$

$\therefore \sum_{0}^{n/2} i^2$ when i = even

Sum of squares of first $\frac{n}{2}$ even numbers

$= n^3$

ii) **Find the tight big Oh for the Best-case and Worst-case scenario.** Explain in one line how you drive it.

Best Case : $\frac{n}{4} = O(n)$ Inner loop doesnot run $k \% 2 != 0$

Worst Case : n^3 Inner loop does run