21L-5654
BDS-5B

Date: Day:

Design and Analysis of Algorithms

Roll number : 21L-5654
Name : Muhammad Hamza Khan

Question 1

Insertion sort runs $4n^2$ steps and merge sort runs $32 n \log n$ steps.
So

$\Rightarrow 4n^2 < 32 n \log n$

dividing by 4 on b/sides.

$\Rightarrow n^2 < 8 n \log n$

dividing by n on both sides

$\Rightarrow n < 8 \log n$

$\Rightarrow \frac{n}{8} < \log n$

$\Rightarrow 2^{n/8} < n$

For $n=8$, $2^{8/8} < 8 \Rightarrow 2 < 8$ (holds)

For $n=16$, $2^{16/8} < 16 \Rightarrow 4 < 16$ (holds)

For $n=32$, $2^{32/8} < 32 \Rightarrow 16 < 32$ (holds)

For $n=44$, $2^{44/8} < 44 \Rightarrow 44.8 \nless 44$ (doesnot hold)

For $n=43$, $2^{43/8} < 43 \Rightarrow 41.4 < 43$ (holds)

Therefore, at $n=44$, the ~~insertion~~ merge sort beats insertion sort.

Hence $2 \leq n \leq 43$

insertion sort beats merge sort.

Date: Day:

Question 2

$100n^2 \times 2^n$, we have to find the value of "n".

for this, we will put values of n till the inequality doesnt hold.

when $n=1$, $100(1)^2 > 2^1$ (holds)

when $n=2$ $100(4) > 2^2$ (holds).

when $n=3$ $100(9) > 2^3 \Rightarrow 900 > 8$ (holds)

when $n=10$ $100(10)^2 > 2^{10} \Rightarrow 10000 > 1024$ (holds)

when $n=20$ $100(20)^2 > 2^{20} \Rightarrow 40{,}000 > 1048576$ (not hold)

when $n=15$ $100(15)^2 > 2^{15} \Rightarrow 22500 > 32768$ (not hold)

when $n=14$ $100(14)^2 > 2^{14} \Rightarrow 19600 > 16384$ (holds)

Therefore, at $n=15$, the first algo (A) ~~runs~~ starts running faster than second (B).

Question 3

(a)

```
int s, i, n;          ——— 1
cin >> n;             ——— 1
s = 0;                ——— 1
for(i=n; i>=1; i--){  ——— n
    s++; }            ——— n-1
```

$T(n) = 1+1+1+n+(n-1)$

$= O(n)$

let $c = 1$,

$T(n) \leq 1 \times n$, it holds

$T(n) = O(n)$ proved.

Date: Day:

(b) int sum, i, j, n; —— 1
sum = 0; —— 1
cin >> n; —— 1
for (i = 1; i <= n; i = i × 2) —— log n
{ for (j = 1; j < n; j = j × 2) —— log n * log n
{ sum++; —— 1 × log n (log n)
}}

$T(n) = 1 + 1 + 1 + \log n + \log n(\log n)$
$= O(\log n^2)$ let c = 1, $T(n) \leq 1 \times \log n^2$

(c) int x = 0, j = n; —— 1
while (j > 0) { —— $\log_4 n$
x += j × 3; —— $\log_4 n$
j /= 4; —— $\log_4 n$
}

$T(n) = O(\log_4 n)$

c = 1, $T(n) \leq 1 \times \log_4 n$.

Question 4

Pseudocode on next page

Date: Day:

Question 4

Pseudocode:

```
SelectionSort(Array, n) {   //n is size
    int temp_index;
    for( i=0 to n-1){  temp_index = i;
        for ( j = i+1 ; j < n ;j++){
            if ( Array[j] < Array[temp_index])
                {
                    temp_index=j;
                }
        if ( Array[i] != temp_index)
        {   swap(Array[i], Array[temp_index]);
        }
    }
}
```

The first loop typically runs n-1 times because selection sort is comparisson based algo in which initially all elements are considered unsorted and after n-1 comparisons the last element is automatically at its correct place. we find minimum element in every iteration and swap it so it comes to its correct position.

> Worst time complexity $\Theta(n^2)$:
If array is sorted in reverse order, the algo will perform maximum steps, it will take n^2 steps.

> Best time complexity $\Theta(n^2)$:
If array is sorted in ascending order, algo will perform minimum steps and still check all elements, so best case is also $\Theta(n^2)$

Date: Day:

Question 5

$T(n) = \frac{1}{8}n^3 - 5n^2$ is $\Theta(n^3)$

$$c_1 n^3 \leq \frac{1}{8}n^3 - 5n^2 \leq c_2 n^3$$

$$c_1 \leq \frac{1}{8} - \frac{5}{n} \leq c_2$$

if $n_0 = 10$

$$c_1 \leq \frac{1}{8} - \frac{5}{10} \leq c_2$$

$$c_1 \leq \frac{5-4\cdot5}{80} \leq c_2$$

~~$c_1 \leq -0.025 \leq c_2$~~

$$c_1 \leq -0.0625 \leq c_2$$

if $n_0 = 1 \Rightarrow c_1 \leq \frac{1}{8} - 5 \leq c_2$; $c_1 = -5$, $c_2 = -1$.

if n is very large $c_1 \leq \frac{1}{8} \leq c_2$; $c_1 = \frac{1}{16}$, $c_2 = 1$

So $O(n^3) = \Theta(n^3) = \Omega(n^3)$.

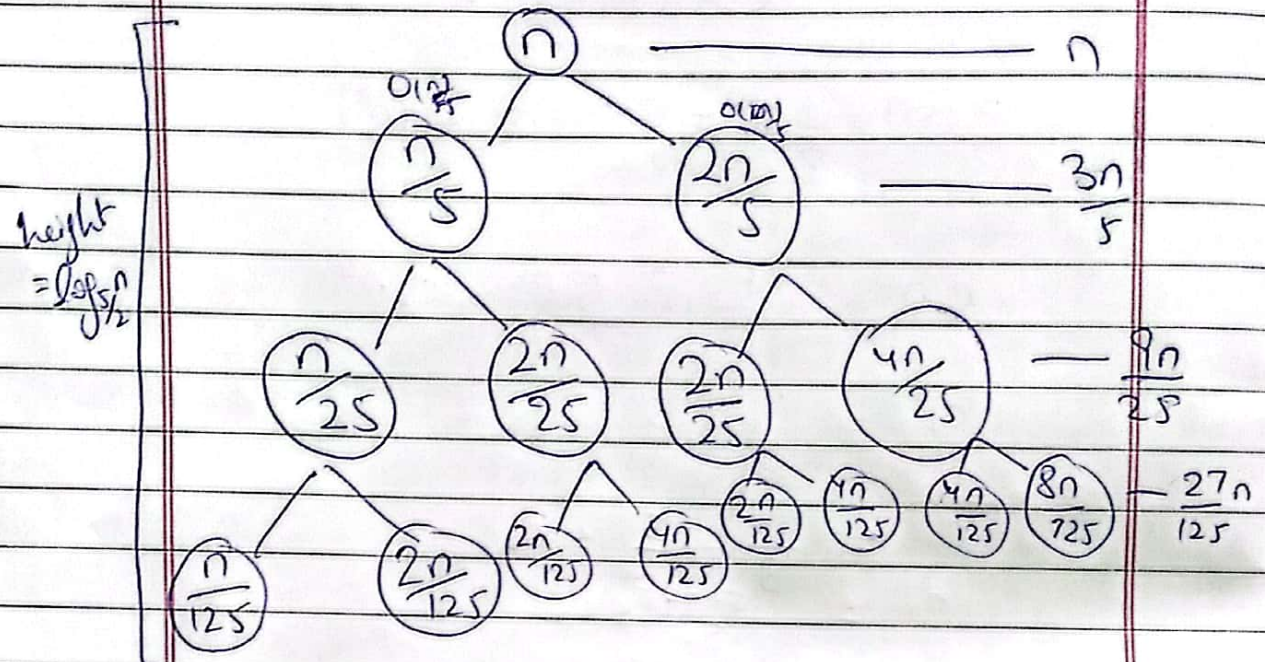
Question 6:

```
Function Mystery(n){
    if (n>1){
        Print "hello"
        Mystery(n/5)
        For(i — n)
            Print "World"
        Mystery(2n/5)
    }}
```

$$T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{2n}{5}\right) + O(n)$$

Solving by recursion tree method:

Date: _____ Day: _____



$$T(n) = \left[n + \frac{3n}{5} + \frac{9n}{25} + \frac{27n}{125} + \cdots \left(\frac{3n}{5}\right)^{\log_{5/2} n} \right]$$

$$= n\left[1 + \frac{3}{5} + \left(\frac{3}{5}\right)^2 + \left(\frac{3}{5}\right)^3 + \cdots \left(\frac{3}{5}\right)^{\log_{5/2} n \cdots \infty} \right]$$

Geometric Series: $\sum_0^n x^n = \frac{1}{1-x}$

$\therefore x = \frac{3}{5}$ & $x < 1$

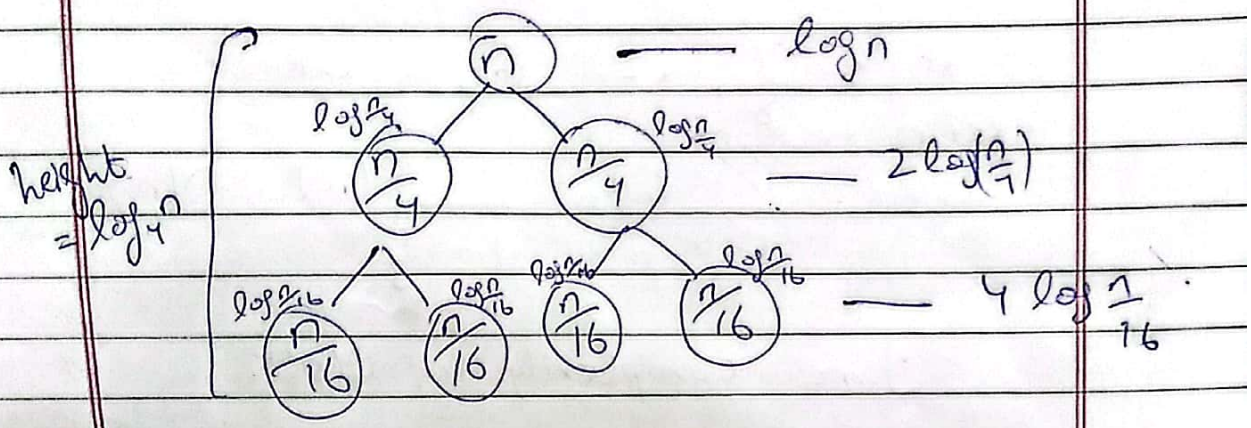
$$= O\left(n\left[\frac{1}{1-3/5}\right]\right) = O\left(n\left(\frac{5}{2}\right)\right) = O(n)$$

Hence, its runtime is $O(n)$.

Question 7

(a) $T(n) = 2T\left(\frac{n}{4}\right) + O(\lg n)$

Date: _____ Day: _____



$$= 2^0 \log\frac{n}{4^0} + 2^1 \log\frac{n}{16^1} + \cdots 2^{\log_4 n} \log\frac{n}{4^{\log_4 n}}$$

$$= \log n + 2(\log n - \log 4) + 4(\log n - \log 16) + \cdots$$

$$= \log n + 2(\log n - 2) + 4(\log n - 4) + \cdots$$

$$= \log n + (2\log n - 4) + (4\log n - 16) + \cdots$$

$$= \log n + 2\log n + 4\log n \cdots - (4 + 16 + \cdots)$$

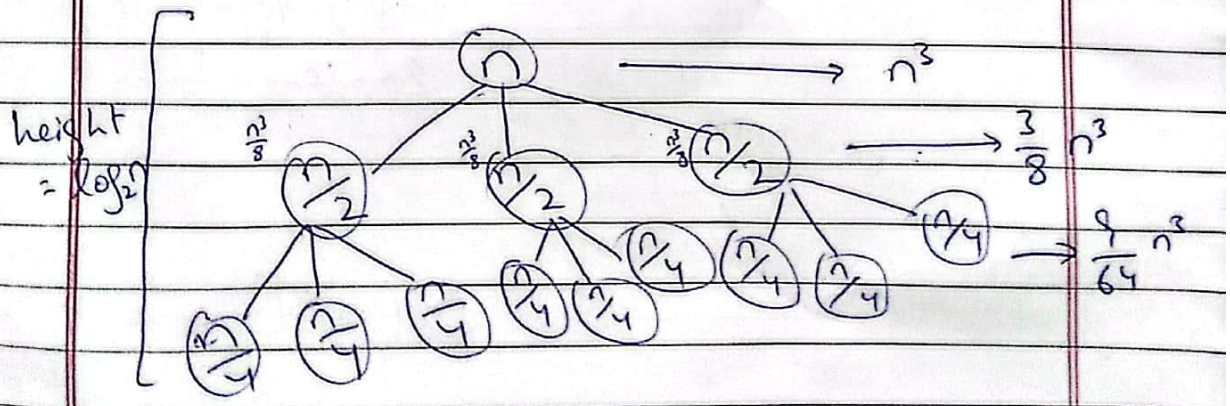
$$= \log n \left(1 + 2 + 4 + \cdots\right) - (4 + 16 + \cdots)$$

$$= \log n \sum_{x=0}^{\log_4 n} \frac{2^{\log_4 n + 1} - 1}{2 - 1} - \sum_{k=0}^{\log_4 n} 4^k$$

$$= \log n \left(2n^{\log_4 2} - 1\right) - \frac{4n - 1}{3}$$

Time complexity = $\Theta(\log n \times n^{0.5})$

(b) $T(n) = 3T(n/2) + O(n^3)$



So, series is

$$= n^3 + \frac{3}{8}n^3 + \frac{9}{64}n^3 + \cdots \left(\frac{3}{8}\right)^{\log_2 n}$$

$$= n^3 \left[1 + \frac{3}{8} + \left(\frac{3}{8}\right)^2 + \cdots \left(\frac{3}{8}\right)^{\log_2 n + \cdots \infty}\right]$$

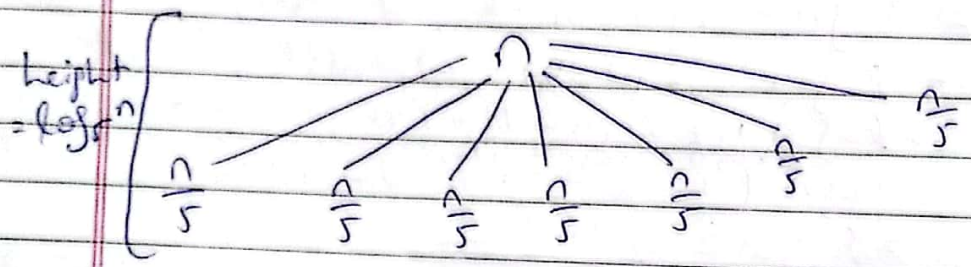
Date: Day:

if $x = \frac{3}{8}$, $x \geq 1$ so, we extend the series to inf.

$$= \frac{1}{1-x}(n^3) = \frac{1}{1-\frac{3}{8}}(n^3) = O(n^3)$$

Time complexity is $O(n^3)$.

(c) $T(n) = 7T(n/5) + O(1)$



$$= 1 + 7 + 14 + \cdots 7^{\log_5 n}$$

$$= 1 + 7 + (7)^2 + (7)^3 + \cdots (7)^{\log_5 n}$$

$$= \frac{7^{\log_5 n + 1} - 1}{7 - 1}$$ ∴ apply formula of $x > 1$

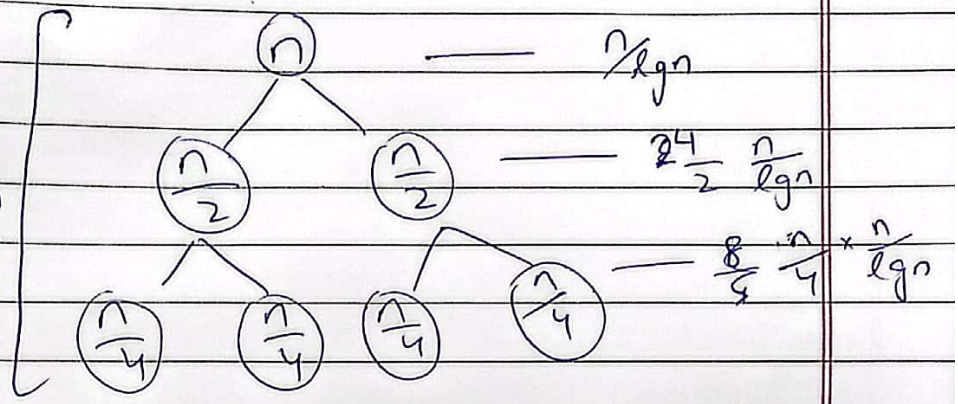
$$\sum_0^n x^n = \frac{x^{n+1} - 1}{x - 1}$$

$$= \frac{(7^{\log_5 n} \times 7) - 1}{6}$$

$$= \frac{(n^{\log_5 7} \times 7) - 1}{6} = O(n^{1.2}) \text{ - complexity}$$

Date: Day:

(d) $T(n) = 2T(\frac{n}{2}) + n/\lg n$



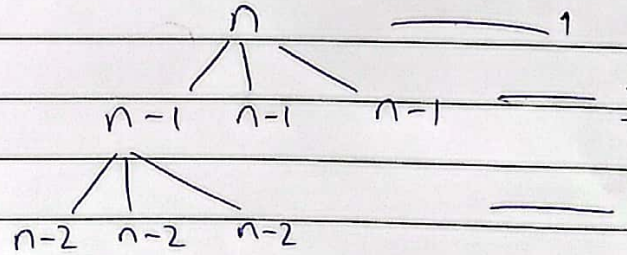
$$= \frac{n}{\lg n}\left(1 + 2 + 4 + \cdots + 2^{\log_2 n}\right)$$

$$= \frac{n}{\lg n}\left(\frac{2^{\log_2 n + 1} - 1}{2-1}\right)$$

$$= \frac{n}{\lg n}(\text{constant}) = O\left(\frac{n}{\lg n}\right)$$

The upper bound of ~~this~~ $T(n)$ is $O\left(\frac{n}{\lg n}\right)$

(e) $T(n) = 3T(n-1) + \Theta(1)$



$\Rightarrow 1 + 3 + 9 + 27 + \cdots$

∴ as $x = 3$ and $x > 1$ so

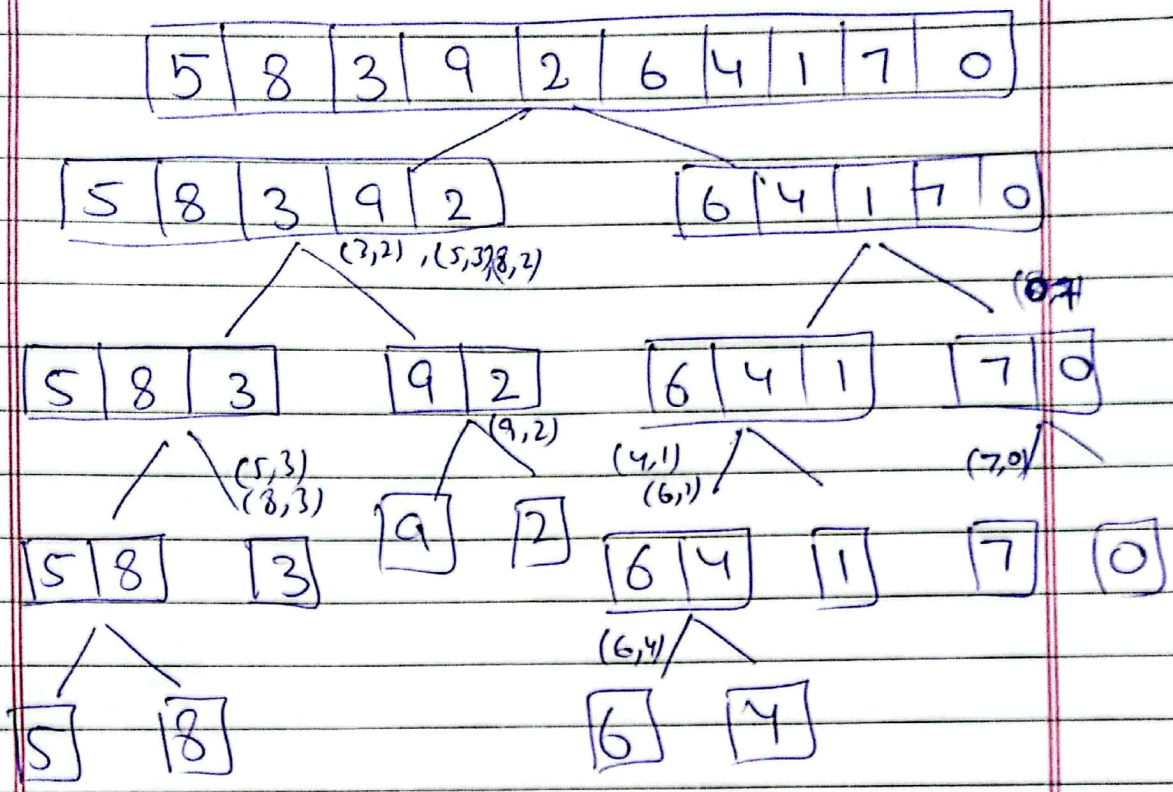
$$\sum_{0}^{n} x^n = \frac{x^{n+1} - 1}{x - 1}$$

$$\Rightarrow \frac{3^{n+1} - 1}{2} = \frac{3^n \times 3 - 1}{2}$$

$$= \Theta(3^n)$$

Date: Day:

Question 3



Split: (5,4) (5,1) (5,0) (8,6) (8,4) (8,1)
(8,7) (8,0) (2,1) (2,0) (9,6) (9,4) (9,7)
(9,1) (9,0) (3,1) (3,0)

left: (9,2) (5,3) (8,3) (3,2) (5,2) (8,2)

Right: (6,4) (7,0) (6,1) (4,1) (1,0) (4,0) (6,0)