RollNumber: 20L-5626
Name: Rohan Jawad

Design & Analysis of Algorithms

Assignment 1

Q1)

$$4n^2 < 32n\log n$$
$$n < 8\log n$$
$$n/8 < \log n$$
$$2^{n/8} < n$$

for $n = 8$

$$2 < 8$$

for $n = 32$

$$2^{32/8} < 32$$
$$16 < 32$$

for $n = 44$

$$2^{44/8} < 44$$
$$45.2 < 44$$

Hence at $n = 44$ insertion beat merge sort.

Q2)

$$2^n < 100n^2$$
$$\log_2 2^n < \log_2 100n^2$$
$$n\log 2 < \log 100 + \log n^2$$
$$n\log 2 < 6.64 + 2\log n$$

for $n = 1$

$$1 < 6.64$$

for $n = 2$ $\quad 2 < 8.64$

$n = 15$

$15 < 14.45$

Q3) a)

$1 + 1 + 1 + n + n - 1$

$2n + 2$

$2n + 2 \le C$ $\quad n_0 = 1$

$2 + \frac{2}{n} \le C$

$2 + 2 < C$

$4 < C$ $\quad n = 2$

$2 + 1 < C$

$3 < 4$

Time Complexity $= O(n)$

b)

$1 + 1 + 1 + \log_2 n * \log_2 n + 1$

$= 4 + 2\log_2 n^2$

$T.C = O(\log_2 n^2)$

c)

$1 + 3\log_4 n$

$1 + 3\log_4 n \ll \log_4 n$

$c = 4, n = 5$

$T.C = 1 + 3\log_4 n$

Scanned with CamScanner

Q4)

Pseudo Code

```
int arr, size
for i=0 to size -1            (n-1)
    int min = i               (n-1)
   for j=1 to size            (n-1)(n-1)
      if arr[j] < arr[min]    (n-1)(n-1)
         min = j
      i++                     (n-1)(n-1)
   j++                        (n-1)(n-1)
   int temp = arr[i]          (n-1)
    arr[i] = arr[min]
     arr[min] = temp
  i++
 end
```

$2(n-1) + 4(n-1)^2$

Time Complexity = $\Theta(n^2)$

Best Case: Is when the array is already sorted but still the algorithm does same number of steps i.e $\Theta(n^2)$

Worst Case: is when the array is in descending order but still the running time of the algorithm will be $O(n^2)$

Q3)

$T(n) = 1/8\, n^3 - 5n^2$

$c_2 f(n) \le 1/8\, n^3 - 5n^2 \le c_1 f(n)$

$1/8\, n^3 - 5n^2 \le c_1 (n^3)$

$1/8 - 5/n \le c_1$ $\quad n_0 = 50$

$1/8 - 1/10 \le c_1$

$\frac{5-4}{40} \le c_1$

$1/40 \le c_1$

$0.028 \le c_1$

At $n = 51$

$1/8 - 5/51 \le 0.028$

$0.125 - 0.098 \le 0.028$

$0.027 \le 0.028$

$c_2 (n^3) \le 1/8\, n^3 - 5n^2$

$c_2 \le 0.026$ $\quad n_0 = 50$

At $n = 51$ $\quad 1/8 - 5/n \ge 0.020$

$0.027 \ge 0.020$

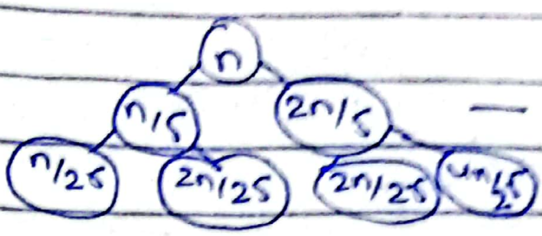
$T(n)$ is $O(n^3)$ and $\Omega(n^3)$

Hence $T(n)$ is $\Theta(n^3)$

Scanned with CamScanner

Q6)

$$T(n) = T(n/5) + T(2n/5) + O(n)$$



— $3n/5$

— $9n/25$

So series is

$$= n + 3n/5 + 9n/25 \cdots (3/5)^{\log_{5/4} n}$$

$$= n\,[1 + (3/5)^1 + (3/5)^2 \ldots (3/5)^{\log_{5/4}}$$

As $3/5 < 1$

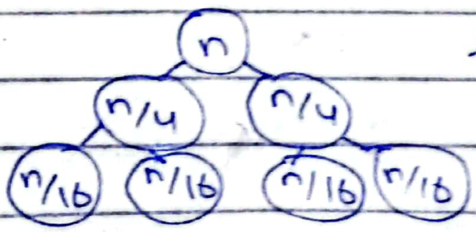
$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$$

$$= n\,[1/_{1-3/5}]$$

$$= n\,[5/2] = O(n)$$

Q7

a) $T(n) = 2T(n/4) + O(\lg n)$



$\log n$

$1/2 \log n$

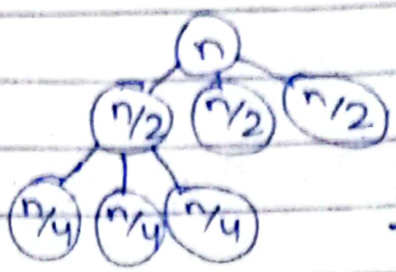
$1/4 \log n$

$$= \log n\,[1 + 1/2 + 1/4 \ldots (1/2)^{\log_4}]$$

$$= \log n\,\left[\frac{1}{1 - 1/2}\right]$$

$$= O(\log n)$$

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



n^3

$3n^3/2$

— $\frac{9}{64} n^3$

Series will be

$$\left(\frac{3}{8}\right)^0 + \left(\frac{3}{8}\right)^1 + \left(3/8\right)^2 \cdots \left(3/8\right)^{\log_2 n}$$

As $3/8 < 1$

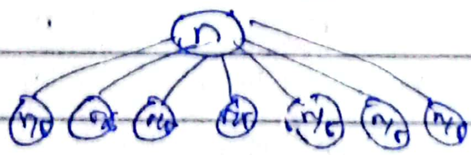
$$\frac{1}{1-x}$$

$$1/1-3/8 \quad n^3$$

$= 1.6\ n^3$

Time Complexity $= O(n^3)$

c) $T(n) = 7T(n/8) + O(1)$



As all nodes are taking constant time so

$= (7)^0 + (7)^1 + (7)^2 \ldots (7)^{\log_8 n}$

As $7 > 1$

So

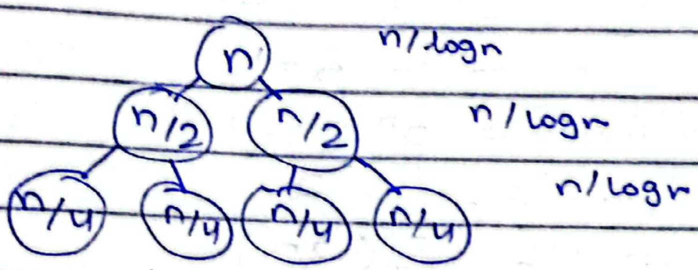
$$\frac{x^{\log_5 n+1}-1}{x-1}$$

$$= \frac{7^{1} + 7^{\log_5 n} - 1}{7-1}$$

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

$= \Theta(n^{1.2})$ – complexity

d) $T(n) = 2T(n/2) + n/\log n$



$n/\log n \, (1 + 2 + 4 \dots 2^{\log_2 n})$

Since $2 > 1$

So

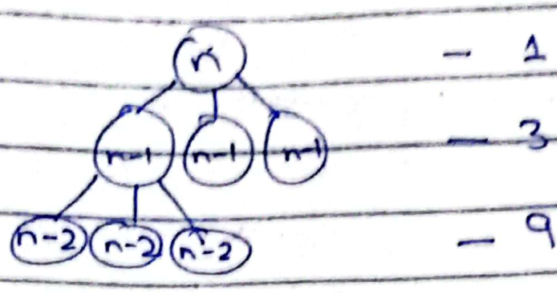
$$= n/\log n \, \frac{2^{\log_2 n+1}-1}{2-1}$$

T.C $= O(n/\log n)$

Scanned with CamScanner

e)

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



Series is

$$= 3^0 + 3^1 + 3^2 \dots 3^n$$

As $3 > 1$

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

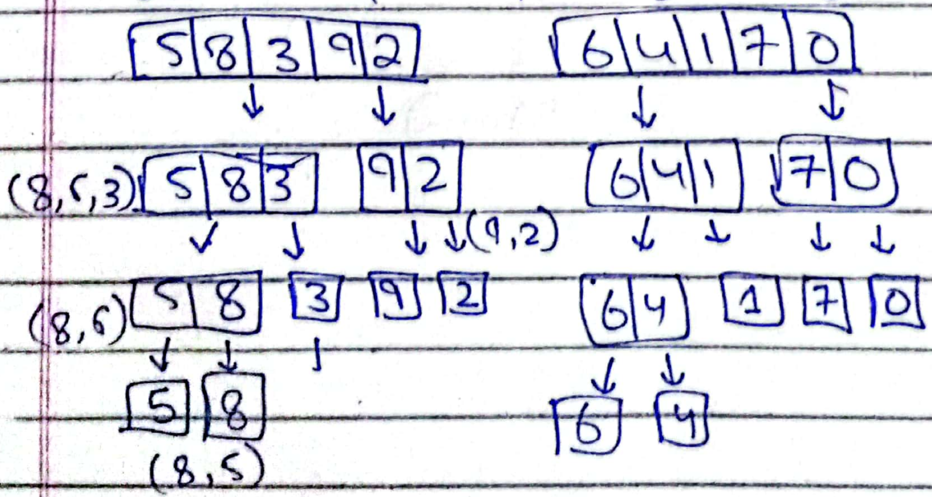
$$= \frac{3^n \cdot 3 - 1}{2}$$

$$T.C = \Theta(3^n)$$

Q8

5 8 3 9 2 6 4 1 7 0

divide & conquer following



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

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

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

Scanned with CamScanner