CSE221 ; Assignment 02

22101593

1

A. $T(n) = 2 \times T\left(\frac{n}{2}\right) + \frac{1}{n}$

$\therefore a = 2$ and $b = 2$.

Though, $a \geq 1$ and $b > 1$, we can use Master Theorem.

$f(n) = 1/n$.

$\therefore R(n) = \frac{f(n)}{n^{\log_b a}} = \frac{1/n}{n^{\log_2 2}} = \frac{1}{n \times n} = \frac{1}{n^2} = n^{-2}$

$\therefore R(n) = n^{-2}$; $r = -2$.

$\therefore r < 0$, $u(n) = O(1)$.

$\therefore$ Time Complexity $T(n) = n^{\log_b a} \left[U(n)\right]$

$= n^{\log_2 2} \times (O(1))$.

$= n \times O(1)$.

B. $T(n) = 2\,T(n/3) + n$.

$a = 2$; $b = 3$; $f(n) = n$.

using Master Theorem;

$$R(n) = \frac{f(n)}{n^{\log_b a}} = \frac{n}{n^{\log_3 2}} = n^{1-\log_3 2} = n^{0.369} = n^0 = 1 = (\log_2 n)^0$$

$r = 0.369 \simeq 0$.

$$u(n) = \frac{(\log_2 n)^{0+1}}{0+1} = \log_2 n.$$

$\therefore$ Time Complexity $= n^{\log_b a} \cdot [u(n)]$

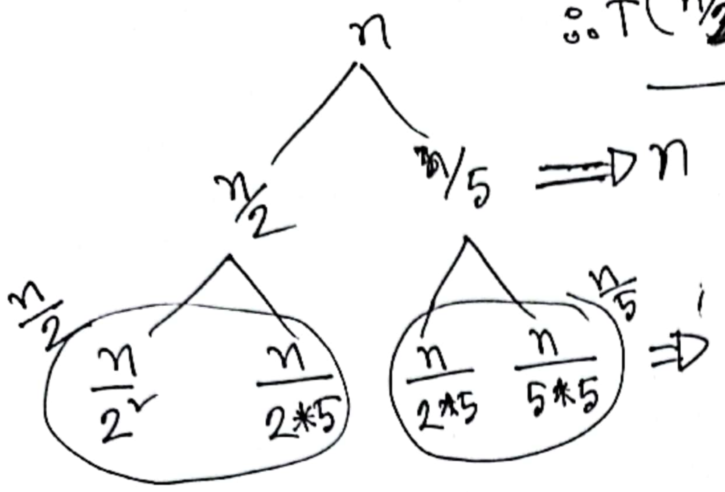
$$= n^{\log_3 2} \times \log_2 n.$$

(Ans)

C. $T(n) = T(n/2) + T(n/5) + n$

$\therefore T(n/2) = T(n/2^2) + T\left(\frac{n}{2*5}\right) + \frac{n}{2}$

$\therefore T(n/5) = T\left(\frac{n}{2*5}\right) + T\left(\frac{n}{5*5}\right) + \frac{n}{5}$



$$\frac{n}{5^K} = 1$$

$\therefore n = 5^K$

$\therefore K = \log_5 n$

$\therefore$ Time Complexity $= n \times \log_5 n$.

D. $T(n) = 2T(n/4) + n^2$

using Master Theorem;

$a = 2;\ b = 4;\ f(n) = n^2$.

$$R(n) = \frac{f(n)}{n^{\log_b a}} = \frac{n^2}{n^{\log_4 2}} = \frac{n^2}{n^{1/2}} = n^{2-1/2} = n^{3/2}.$$

$r = 3/2 > 0$

$\therefore u(n) = O(n^r) = O(n^{3/2})$.

$\therefore$ Time Complexity $= n^{\log_4 2}\,[O(n^{3/2})]$

$= n^{1/2} \times n^{3/2} = O(n^2)$.

2.

A. The teacher's comment is based on the fact that QuickSort's worst-case performance, which is $O(n^2)$, occurs when the input array is already sorted, and the pivot is chosen as the first or last element. In this case, the array is sorted in descending order and the pivot is the first element, which is the largest number.

B. We know, for Quick Sort, the recurrence relation is $T(n) = T(n-1) + n$

~~$\therefore a=1$; $b=1$; $f(n)=n$~~

~~Though $a \geq 1$ and $b \geq 1$~~

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

$$T(2) = T(2-1) + 2 = T(1) + 2 = 1 + 2$$

$$T(3) = T(3-1) + 3 = T(2) + 3 = 1 + 2 + 3$$

$$T(4) = T(4-1) + 4 = T(3) + 4 = 1 + 2 + 3 + 4.$$

$\vdots$

$$\therefore T(n) = \frac{n(n+1)}{2} = \frac{n^2 + n}{2} = \frac{n^2}{2} + \frac{n}{2}$$

$$\therefore T(n) = O(n^2).$$

(Ans)

3

$A = 1\ 2\ 3$ (A_1, B_1, A_2)
$D = 4\ 5\ 6$

A.

$A = A_1 \quad A_2 \quad A_3$

$B = B_1 \quad B_2 \quad B_3$

$\therefore A = A_1 * 10^{\left(2*\frac{n}{3}\right)} + A_2 * 10^{\left(\frac{n}{3}\right)} + A_3$

$B = B_1 * 10^{\left(2*\frac{n}{3}\right)} + B_2 * 10^{\left(\frac{n}{3}\right)} + B_3$

B.

$A * B = \left(A_1 * 10^{\left(2*\frac{n}{3}\right)} + A_2 * 10^{\left(\frac{n}{3}\right)} + A_3\right) \cdot \left(B_1 * 10^{2n/3} + B_2 * 10^{n/3} + B_3\right)$

$= A_1 * B_1 * 10^{\left(\frac{2n}{3} + \frac{2n}{3}\right)} + A_1 B_2 10^{\left(\frac{2n}{3} + \frac{n}{3}\right)} + A_1 B_3 10^{\frac{2n}{3}}$

$+ A_2 B_1 10^{\left(\frac{n}{3} + \frac{2n}{3}\right)} + A_2 B_2 10^{\left(\frac{n}{3} + \frac{n}{3}\right)} + A_2 B_3 10^{\frac{n}{3}}$

$+ A_3 B_1 * 10^{\frac{2n}{3}} + A_3 B_2 * 10^{\frac{n}{3}} + A_3 B_3 .$

$= A_1 B_1 10^{4n/3} + A_1 B_2 10^{n} + A_1 B_3 10^{\frac{2n}{3}} + A_2 B_1 10^{n} + A_2 B_2 10^{n} + A_2 B_3 10^{n/3}$

$+ A_3 B_1 10^{2n/3} + A_3 B_2 10^{n/3} + A_3 B_3 .$

$= A_1 B_1 10^{\frac{4n}{3}} + (A_1 B_3 + A_3 B_1) 10^{2n/3} + (A_2 B_3 + A_3 B_2) 10^{n/3}$

$(A_1 B_2 + A_2 B_1 + A_2 B_2) \times 10^{n} + A_3 B_3 .$

(c)

```
function BenjaminKaratsuba (A, B, n) :
    if n == 1 ;
        return A * B

    #Split A & B into 3 parts
    A1, A2, A3 = Split (A, n)
    B1, B2, B3 = Split (B, n) .

    #Recursive Calls.
    P1 = BenjaminKaratsuba (A1, B1, n/3)
    P2 = BenjaminKaratsuba (A2, B2, n/3)
    P3 = BenjaminKaratsuba (A3, A3, n/3)
    P4 = BenjaminKaratsuba (A1+A2, B1+B2, n/3)
    P5 = BenjaminKaratsuba (A2+A3, B2+B3, n/3)
    P6 = BenjaminKaratsuba (A3+A1, B3+A1, n/3)

    #Combine the results.
    result = P1 * 10^(2n/3)
             + ((P4-P1-P2) + (P5-P2-P3)) 10^(n/3)
             + P3 .

    return result.
```

(D) $T(n) = 3T(n/3) + n$

$a = 3;\ b = 3;\ f(n) = n.$

Though $a \geq 1$ & $b > 1$; I will use Master Theorem.

$R(n) = \dfrac{n}{n^{\log_3 3}} = \dfrac{n}{n} = 1 = n^0 \quad \therefore r = 0 \quad \therefore u(n) = \dfrac{(\log_2 n)^{0+1}}{0+1} = \log_2 n.$

$\therefore$ Time Complexity, $T(n) = n^{\log_b a} \times [U(n)]$

$= n^{\log_3 3} \times \log_2 n = O(n \log n).$

On the other hand,

Karatsuba algorithm's time complexity $= O(n^{1.59})$.

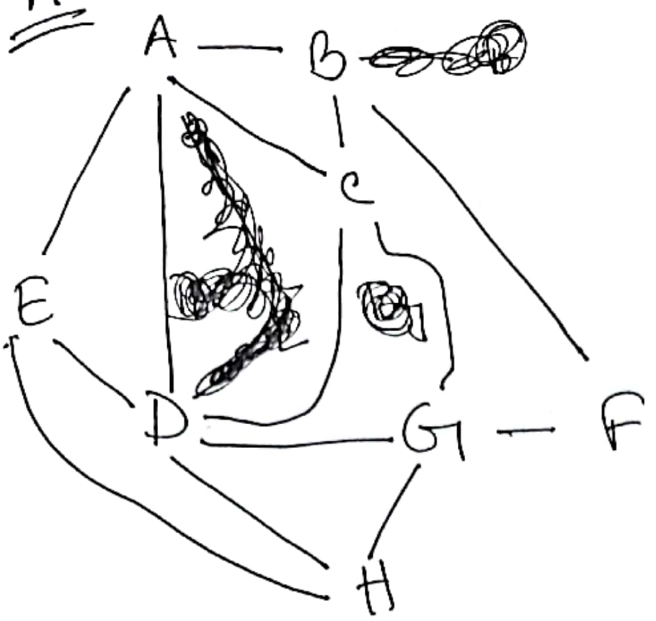
$< O(n \log n)$.

$\therefore$ That's why, Benjamin's claim of getting a faster algorithm is not true.

(Ans)

(4)

A.



B.

Adjacency list :-

A → B → C → D → E | Null

B → A → C → F | Null

C → A → B → D → G | Null

D → A → C → E → G → H | Null

E → A → D → H | Null

F → B → G | Null

G → C → D → F → H | Null

H → D → E → G | Null

Adjacency Matrix :-

	A	B	C	D	E	F	G	H
A	0	1	1	1	1	0	0	0
B	1	0	1	0	0	1	0	0
C	1	1	0	1	0	0	1	0
D	1	0	1	0	1	0	1	1
E	1	0	0	1	0	0	0	1
F	0	1	0	0	0	0	1	0
G	0	0	1	1	0	1	0	1
H	0	0	0	1	1	0	1	0

C.

A & B = 1 (ACB)

A & C = 2 (ABC, ADC)

A & D = 2 (AED, ACD)

A & E = 1 (ADE)

A & F = 0

A & G = 2 (ACG, ADG)

A & H = 2 (AEH, ADH)

B & C = 1 (BAC)

B & D = 2 (~~A~~BCD, BAD)

B & E = 1 (BAE)

B & F = 0

B & G = 2 (BCG, BFG)

B & H = 0

C & D = 2 (CAD, CGD)

C & E = 2 (CAE, CDE)

C & F = 2 (CBF, CGF)

C & G = 1 (CDG)

C & H = 2 (CGH, CDH)

D & E = 2 (EAD, EHD)

D & F = 1 (DGF)

D & G = 2 (DCG, DHG)

D & H = 2 (DEH, DGH)

E & F = 0

E & G = 2 (E~~D~~G, EHG)

E & H = 1 (EDH)

F & G = 0

F & H = ~~0~~ 1 (FGH)

G & H = 1 (GDH)

D. Breadth First Search Algo

queue = $\cancel{F}$ | $\cancel{B}$ | $\cancel{G}$ | $\cancel{A}$ | $\cancel{C}$ | $\cancel{D}$ | $\cancel{H}$

visited = F, B, G, A | C | D | H

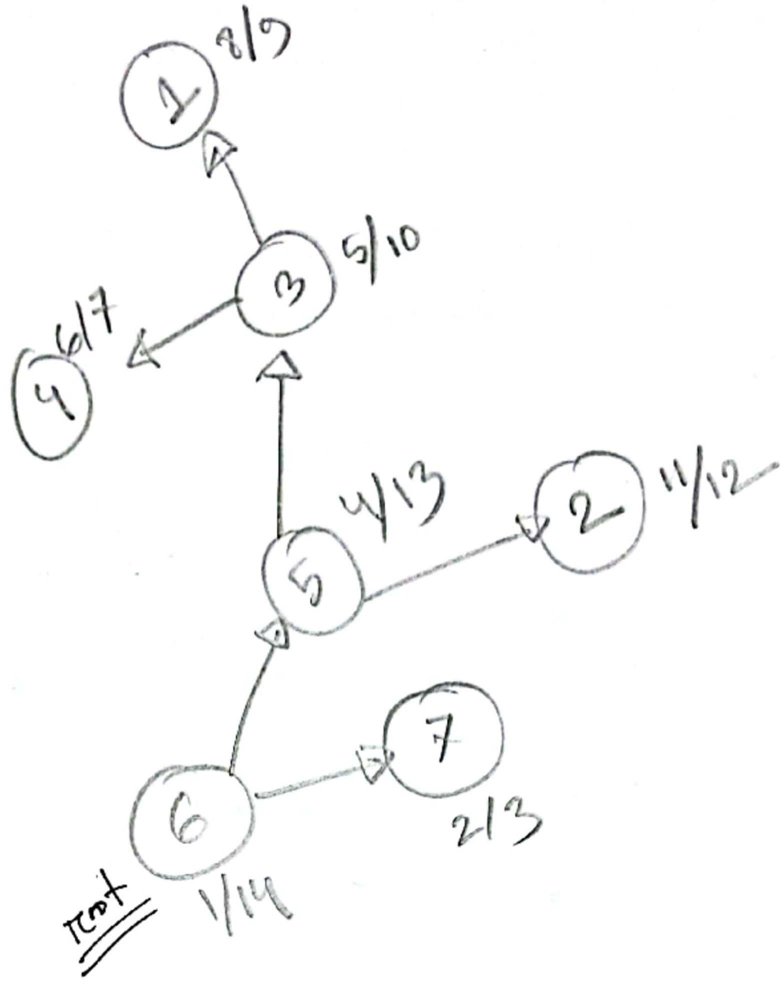
∴ From this, we can say that,

F can see all the post of B, G, A, C, D, H along with his post.

That's why we can say that, F can't see the posts of all other users in his feed.

5

A.



B.

Nodes	1	2	3	4	5	6	7
Parent	3	5	5	3	6	[illegible]	6
Starting Time	8	11	5	6	4	1	2
Finish Time	9	12	10	7	13	14	3
distance from root	3	3	2	3	1	0	1

6

a. $\sum_{v \in V} \deg(v) = 2m$

$\Rightarrow \deg(A) + \deg(B) + \deg(C) + \deg(D) + \deg(E) + \deg(F) + \deg(G) + \deg(H) + \deg(S) = 2m.$

$\Rightarrow 3 + 4 + 4 + 3 + 3 + 3 + 4 + 2 + 2 = 2m$

$\Rightarrow 28 = 2m$

$\therefore m = 14$

And we see that total 14 edges ~~are~~ has in the graph.

And from $\sum_{v \in V} \deg(v) = 2m$ formula, we see that the number of edges (m) are 14.

That's why for undirected graph $\sum_{v \in V} \deg(v) = 2m$ is true. And Given graph is also undirected. ~~That's~~ For this reason we say that this statement is true for ~~that~~ the undirected graph.

b.

For simple Graph,

We know that, maximum number of edges will be = $\frac{n(n-1)}{2}$; n = num of nodes.

from the graph,

we have 8 nodes & 12 edges.

∴ So, if given graph will be simple graph, then the maximum number of edges will be = $\frac{8\times(8-1)}{2} = 28$.

Therefore, the number of additional edges that can be added to this graph is = 28-12 = 16.

(Ans)