(i) If $f(n) = O(g(n))$

Using formal definition of Big-Oh,

If $f(n) = O(g(n))$, then there exists positive constants c and n_0 such that

$f(n) \leq c \cdot g(n)$ for all $n \geq n_0$.

To Prove: $f(n) + g(n) = O(g(n))$

LHS $\Rightarrow f(n) + g(n)$

using $f(n) = O(g(n))$

$\Rightarrow \leq c \cdot g(n) + g(n)$

$\Rightarrow \leq (c+1)\, g(n)$

$\Rightarrow \leq C \cdot g(n)$

$\Rightarrow O(g(n))$

thus, $f(n) + g(n) = O(g(n))$, where C exists positive constants and n_0 such that $f(n) + g(n) \leq C \cdot g(n)$ for all $n \geq n_0$.

(2)

I) $a(n) = \Theta(d(n) \cdot e(n))$

II) $a(n) \cdot d(n) \cdot e(n) = \Theta(n^3)$

III) $b(n)^3 = \Omega(a(n)^2)$

IV) $c(n) + d(n) = \Theta(b(n)^2)$

V) $d(n)^2 = \Theta(a(n) \cdot e(n))$

VI) $e(n)^2 = \Omega(b(n))$

} GIVEN 6 FACTS

fact I) $a(n) = \Theta(d(n) \cdot e(n))$

USING fact II IN → $a(n) \cdot d(n) \cdot e(n) = \Theta(n^3)$

→ $(d(n) \cdot e(n)) \cdot d(n) \cdot e(n) = \Theta(n^3)$ ⟹ using I

→ $d(n)^2 \cdot e(n)^2 = \Theta(n^3)$

Square root both sides.

→ $d(n) \cdot e(n) = \Theta(n^{3/2})$ —— (A)

IN FACT I USING (A)

∴ $\boxed{a(n) = \Theta(n^{3/2})}$ —— (X1)

NOW, IN FACT V → $d(n)^2 = \Theta(a(n) \cdot e(n))$

$d(n)^2 = \Theta(n^{3/2} \cdot e(n))$ ⟹ USING (X1)

USING (A), ONE POSSIBLE SOLN IS

$d(n) = \Theta(n^{3/4})$ and $e(n) = \Theta(n^{3/4})$ —— (X3)

USING FACT III $\Rightarrow b(n)^3 = \Omega(a(n)^2)$

USING $X_1 \Rightarrow a(n) = \Theta(n^{3/2})$

So, $a(n)^2 = \Theta(n^3)$

$\Rightarrow b(n)^3 = \Omega(n^3)$

$\Rightarrow \boxed{b(n) = \Omega(n)}$ ——— X_2

OR

asymptotically at least, $\boxed{b(n) = \Theta(n)}$

USE FACT IV: $c(n) + d(n) = \Theta(b(n)^2)$

Since, $b(n) = \Theta(n) \Longrightarrow b(n)^2 = \Theta(n^2) \Longrightarrow$ USING X_2

$\Rightarrow c(n) + d(n) = \Theta(n^2)$

USING X_3, WHERE $d(n) = \Theta(n^{3/4})$

$c(n)$ SHOULD BE THE DOMINANT TERM

So, $\boxed{c(n) = \Theta(n^2)}$ ——— X_4

Hence,

$$\boxed{\begin{aligned} a(n) &= \Theta(n^{3/2}) \\ b(n) &= \Theta(n) \\ c(n) &= \Theta(n^2) \\ d(n) &= \Theta(n^{3/4}) \\ e(n) &= \Theta(n^{3/4}) \end{aligned}}$$

(3)

LINE 3: OUTER LOOP RUNS FOR i = 1 to n

$$\therefore O(n)$$

LINE 5 : LOOPS TILL $j < n$, BUT j INCREMENTS BY $j = 2 * j$.

$$\therefore O(\log n)$$

LINE 6 : INNERMOST LOOPS TILL j to n

$$\therefore O(n)$$

Total complexity : $O(n) \times O(\log n) \times O(n)$

$$= O(n^2 \cdot \log n)$$

(4) All 3 Recursive calls are →

$(n - n/3)$

Line 11 : $n - n/3$

Line 12 : $n - n/3$

Line 13 : $n - n/3$

$\therefore T(n) = 3 \cdot T\left(n - \frac{n}{3}\right) + O(n)$

$= 3 \cdot T\left(\frac{2n}{3}\right) + O(n)$

OR

$$T(n) = 3 \cdot T\left(n/3/2\right) + O(n)$$

$\therefore$

(5) $T(n) = 3 \cdot T(n/3/2) + O(n)$

Here, $a = 3$

$b = 3/2$

$$\log_b a = \log_{3/2} 3 = \frac{\log 3}{\log 3/2}$$

$\log_2 3 \approx 1.584$

$$\log_2 \frac{3}{2} = \log 3 - \log 2 = 1.584 - 1$$

$$= 0.584$$

$$\therefore \log_{3/2} 3 = \frac{1.584}{0.584} = 2.712$$

Hence, $c = 2.712$ & $k = 1$

$\therefore c > k$, Case I WHERE $n^{\log_b a} > n^c$

$$\therefore \boxed{O(n^{2.712})}$$

6 (a) Mode retrieval in $O(1)$
Add elements in $O(\lg n)$
Remove " in $O(\log n)$

The Data Structure we can use →

1) A Balanced B.S.T (AVL tree) to STORE ELEMENTS WHICH WILL ALLOW $O(\lg n)$ INSERTION and DELETION

2) A HashMap to track the frequency of an element.
and a variable to track the mode in the data.

(b) MAINTAINING A MODE VARIABLE AND SIMPLY RETURNING IT.

THIS OPERATION WILL TAKE $O(1)$

(c) ADDING AN ELEMENT

- INSERT ELEMENT IN B.S.T (takes $O(\log n)$)
- UPDATE FREQUENCY OF ELEMENT IN HASH MAP
- IF FREQUENCY ELEMENT > FRQ. OF MODE THEN
- UPDATE MODE.

(d) REMOVING AN ELEMENT

— REMOVE ELEMENT FROM B.S.T (takes O(log n) time)

— UPDATE FREQUENCY IN HASH MAP

~~— IF FREQUENCY OF DECREASED OF ELEMENT~~

— IF MODE ELEMENTS FREQUENCY IS CHANGED,

— SEARCH HASHMAP FOR NEW MODE.