Date:.....

Sun Mon Tue Wed Thu Fri Sat

Q.1

9, 19, 14, 29, 13, 4, 23, 10, 37

split in 4 parts

9,19 | 14,29 | 13,4 | 23,10,37

Sort

9,19 | 14,29 | 4,13 | 10,23,37

Merge	i ↓	j ↓	k ↓	l ↓	4 parts
	L-1	L-2	L-3	L-4	i, j, k, l
Start →	2 (9)	(14) 5	(4) 1	(0) 3	i<j<, k<l
4 →	(9)	8 (29) →	(13) →	7 (23) →	
→	→	→	→	37 →	

Compare 9, 14, (4), 10

Compare (9), 14, 13, 10

Compare 19, 14, 13, (10)

Compare 19, 14, (13), 23

Compare 19, (14), 23

Compare (19), 29, 23

Compare 29, (23), 37

Compare (29), 37

Compare 37

Merge list 4, 9, 10, 13, 14, 19, 23, 29, 37

Array divide into 4 parts $\log_4 n$ levels of division.

$\log_4 n$ is same as $\frac{\log_2 n}{\log_2 4} = \frac{\log_2 n}{2}$

which means $\log_2 n$ divided by 2

Merging Arrays required $O(n)$ time
each array has a size $n/4$

Combining both

Divide takes: $\log_4 n$ level's and each level takes $O(n)$ time for merging.

Therefore overall time complexity becomes

$O(n) \times O(\log_4 n) = O(n \times \log_4 n)$.

Date:.....

Sun Mon Tue Wed Thu Fri Sat

Q.2

$3n^3 + 5n^2 + 25n = \Omega(n^3)$

$\Omega(g(n))$ = lower bound

If $f(n) = \Omega(g(n))$, then there exist a positive constant c and n_0 such that for all $n \geq n_0$ $f(n) \geq c \times g(n)$

$3n^3 + 5n^2 + 25n \geq c \times g(n)$

We have to guess, c, $g(n)$ and n_0

Since $3n^3$ is the dominating term b/c n^3 grows faster then n^2 and n as n becomes large.

$3n^3 + 5n^2 + 25n \geq 3n^3$

Here $c = 3$, $n_0 = 1$

OR

$c > 0$

As n increases $3n^3 + 5n^2 + 25n$ behaves like $3n^3$ for large n, $3n^3$ grows same rate as n^3. The entire equation gives at least fast as n^3

Therefore, $3n^3 + 5n^2 + 2n = \Omega(n^3)$

$\Omega(n^3)$ as n^3 is the term that defines growth rate

$5\log_2(\log_2 n) + 4\log^2 n = O(?)$

Compare both terms

$\log_2(\log_2 n)$ much slower than $\log^2 n$, $\log^2 n$ is the square of logarithmic, which grow faster than just $\log_2 n$ and much faster than $\log_2(\log_2 n)$.

$\log^2 n = (\log_2 n)^2$ grows faster than $\log_2(\log_2 n)$ as n increases.

So, $4\log^2 n$ is dominating term must faster than $5\log_2(\log_2 n)$ for large values of n

So big oh $f(n) \leq c \times g(n) \quad \forall n \geq n_0$, and we prove c

$5\log_2\log_2 n + 4\log^2 n \leq c * g(n)$

So

$5\log_2\log_2 n + 4\log^2 n \leq 5 * (\log^2 n)$

Big oh is dominant term

$c = 6 \quad , n_0 = 2^{10} = 1024$

$n \geq 1024$.

Date: .....

Sun Mon Tue Wed Thu Fri Sat

1)

First loop

for (i = n/2 ; i > 1 ; i /= 6)

i = n/2 and this divided by 6 k: itr

Solve this and get $((n/2)/6^k) \approx 1$ (no. of itr in which (n/2)/6 becomes 1.

No. of itr is $O(\log_6 n)$

$= O(n \log n)$

2- Inner loop

for (j = 2 ; j <= n , j *= 4)

j = 2 and multiplied by 4 in each iteration.

No. of itr, 4 times, $2 * 4^k$ → Solve this and get

$O(\log_4 n)$ $O(n \log n)$

3- Last loop

for (k = 0; k <= j , k = * 3)

k = 0 and multiplied by 3 in each iteration

Let k=1 the no of itr = how many times I * 3 before j exceeds.

$k = 3^t$

$3^t \leq j$

Apply log on b/s

$\log_3(3^t) \leq \log_3 j$

$t \log_3 3 = \log_3 j$

$t = \log_3 j$

$t = O(\log_3 j) = O(\log j)$

1st loop:

$(n/2)/(6k) = 1$

$(n/2) = 6k$

Apply log on b/s

$\log_2(n/2) \quad k \log_2 6$

$\frac{\log_2 n - \log_2 2}{\log_2 6} = k$

$\frac{\log_2 n - \log_2 2}{\log_{10} 6} = k$

constant

$\frac{\log n}{\log 6} = O(\log n)$

Date:.....

Sun Mon Tue Wed Thu Fri Sat

2nd loop

$2 * 4^k > n$

$\log 2 * \log 4^k > \log n$

$\log 2 + k \log 4 > \log 2n$

$k \log 4 > \log 2n - \log 2$

$k > \frac{\log 2n - \log 2}{\log 4}$

$k > \frac{\log 2n}{\log 4} - \frac{\log 2}{\log 4}$ constant

$\frac{\log 2n}{\log 4}$ $O(\log n)$

$O(\log n) * O(\log n) * O(\log n)$

T.C :> $O(\log n)^3$

(iii)

1st loop

$i = n$, $i = i/2$	$(n/2)^{k \text{ times}} = 1$

$i = n/2$

$n = 2^k$

$\log n = k$

2nd loop

$j = 2 \; ; \; j <= n \; , \; j += 1$

'j' starts with 2 and increments n times

$(j+2) * n$ times

$nj + 2n =$

$O(n)$

$n \times \log n = = O(n \log n)$

Date:.....

Sun Mon Tue Wed Thu Fri Sat

h-0

$\therefore \log \cdot \log_2 n^2 = \log(2 \cdot \log n) = \log 2 + \log \log(n)$

$\log(\log(n^2)) : \log(\log n)$

2 $\log^2 n$

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

Faster than $n^{1/2}$ $\log(\log n)$ slower than

n^{-2} small compare to polynomial and log

$\log_2\left(2^{\log_2(n^2)}\right)$

Apply log: $\log_2(n^2) = 2 \log n$

$n^{1/2}$ Faster than Log but slower than n^2

quadratic n^2 is faster than all

n^{-2}, $\log \cdot \log n^2$, $2 \log n$, $\log^2 n$, $n^{1/2}$, n^2.

i) $f(n) \cdot \omega(g(n))$

$f(n) \cdot \omega(g(n))$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$$

$f(n) = n$ and $g(n) = \sqrt{n}$

$\lim_{n \to \infty} n/\sqrt{n} = \infty$

$$\frac{n}{\sqrt{n}} = \frac{n^1}{n^{1/2}} = n^{1-1/2} = n^{1/2} \sqrt{n}$$

$\lim_{n \to \infty}$ as $n \to \infty$, then $n \sqrt{n} = \infty$

$n \cdot \omega(\sqrt{n})$ proved.

Date:.....

Sun Mon Tue Wed Thu Fri Sat

14)

$f(n) = O(g(n))$, then $g(n) = \Omega(f(n))$

Let $f(n) = 3n$

$g(n) = 6n$

Check if $f(n) = O(g(n))$

$f(n) \leq c * g(n)$

$3n \leq c*(6n)$

$3n \leq 2*6n$ | OR $3n \leq c*n$

$3n \leq 12n$ | $\frac{3n}{6n} \leq c$

$\frac{1}{2} \leq c$

Proof: $3n$ is $O(n)$ ✓

$3n$ is $O(g(n))$

$g(n) = \Omega(f(n))$

$g(n) \geq c * f(n)$

$6n \geq c*3n$ | $6n \geq c*3n$

$\frac{6n}{3n} \geq c$ | $6n \geq 1*3n$

$c = 2$. ✓ | $6n \geq 3n$ $c = 1$ ✓

$g(n) = \Omega(f(n))$

$g(n)$ is $\Omega(f(n))$

$g(n)$ is $n = \Omega(n)$

$f(n) = 2^n$

$f(n) = O(f(n)^2)$

$f(n) = 2^n$ so $f((n))^2 = (2^n)^2 = 2^{2n}$

$f(n) \leq c \times f(n)^2$

$2^n \leq c \cdot 2^{2n}$ R.H.S cannot be big than L.H.S

$\frac{2^n}{2^{2n}} = 2^n \cdot 2^{2n} = 2^{-n} \leq c$

$2^{-n} \leq c$ is always less than c

So this disapproves $f(n) = \Phi f(n^2)$

Date:

Sun Mon Tue Wed Thu Fri Sat

iv)

$T(n) = \Theta f(n)$, then $T(n) = O f(n)$ and $T(n) = \Omega f(n)$

Let $T(n)$ Worstcase of the Algo is

Worst case $T(n) = 5n^2 + 4n + 2$

Best case $T(n) = 2n^2 + 3$

Let check $T(n) = \Theta f(n)$ $\Theta(n^2)$

$T(n) = O(n^2)$

Worst case

$T(n) = 5n^2 + 4n + 2 \leq c * 9n^2$

Here $c = 9$. Proved

Now $T(n) = \Omega(n^2)$

Best case

$2n^2 + 3 \geq c * (n^2)$

$2n^2 + 3 \geq 1 \cdot n^2$ ← even $2n^2$ is always more than cn^2.

Proved $T(n) = \Omega(n^2)$.

~~Big Oh of Algorithm is~~ worst

$T(n) = n^2$.