Q1. We know,

$$o(g(n)) = \left\{ f(n) : \lim_{n\to\infty} \frac{f(n)}{g(n)} \right\} = 0.$$

Given order, $n!$, 2^n, $n^{\log n}$, $n.\log n$, $n\sqrt{n}$.

Consider $n!$, 2^n :

$o(2^n) = n!$ or $\omega(2^n) = n!$ let's check,

$f(n) = 2^n$ $g(n) = n!$ say,

$$\lim_{n\to\infty} \frac{2^n}{n!} = \lim_{n\to\infty} \frac{2^n}{n(n-1)\cdots 3 \cdot 2 \cdot 1}$$

$$= \lim_{n\to\infty} \frac{2 \times 2 \times \cdots 2 \text{ (n-times)}}{n(n-1) \cdots 3 \cdot 2 \cdot 1}.$$

$$= \lim_{n\to\infty} \frac{2}{n} \cdot \lim_{n\to\infty} \frac{2}{n-1} \cdots \lim_{n\to\infty} \frac{2}{2} \lim_{n\to\infty} \frac{2}{1}$$

$$\left(\because \lim_{n\to\infty} \frac{1}{n} = 0\right).$$

$= 0 \times$ anything $= 0$.

$\therefore 2^n = o(n!)$

So order (increasing among 2^n, $n!$) is 2^n, $n!$

Scanned with CamScanner

We still need to sort, $n^{\log n}$, $n \cdot \log n$, $n\sqrt{n}$

$f(n) = n^{\log n}$ $\quad g(n) = n \cdot \log n$

$(n)^{\log n} = o(n \log n)$ (or) $n \log n = \omega((n)^{\log n})$

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

(This is difficult) to

let's try.

$f(n) = n \cdot \log n \quad g(n) = n\sqrt{n}$.

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

$$= \lim_{n\to\infty} \frac{1}{\sqrt{n}} \cdot \lim_{n\to\infty} \log n$$

$$= 0 \cdot \lim_{n\to\infty} \log n$$

$$= 0$$

$\therefore n \cdot \log n = o(n\sqrt{n})$

Compare $n\sqrt{n}$ with $n^{\log n}$:-

$f(n) = n\sqrt{n} \qquad g(n) = n^{\log n}$

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

$$= \lim_{n\to\infty} \frac{n\cdot\sqrt{n}}{n^{\log n}} = \lim_{n\to\infty} \frac{1}{n^{\log n}} \cdot \lim_{n\to\infty} n\sqrt{n}$$

$$= \lim_{n\to\infty} \frac{1}{n^{\infty}} \cdot \lim_{n\to\infty} n\sqrt{n}$$

$$= 0 \cdot \lim_{n\to\infty} n\sqrt{n} = 0$$

$\therefore$ $n\sqrt{n} = o.(n^{\log n})$

$\therefore$ We've the complexity order, as

$n \log n,\ n\sqrt{n},\ n^{\log n}$.

Let's compare $n^{\log n}$ with $2^n, n!$

$f(n) = n^{\log n}$ $\quad g(n) = 2^n$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^{\log n}}{2^n}$$

$$= \lim_{n\to\infty} \frac{1}{2^n} \cdot \lim_{n\to\infty} n^{\log n}.$$

$$= \lim_{n\to\infty} \frac{1}{2^{\infty}} \cdot \lim_{n\to\infty} n^{\log \infty}$$

$$= 0.$$

$\therefore$ $n \cdot \log(n) = o.(2^n)$

We can conclude that,

The montonically increasing order,

$n \log n$, $n\sqrt{n}$, $n^{\log n}$, 2^n, $n!$

2a) $\sum_{K=2}^{n-1} K \log K \leq \frac{1}{2} n^2 \log(n) - \frac{n^2}{4}$.

From monotonically increasing function,

$\sum_{K=m}^{n} f(K) \leq \int_{m}^{n+1} f(x)\,dx$ by comparing,

we, get $f(K) = K \log K$.

$\cancel{\sum_{K=2}^{n-1} K \log K \Rightarrow \int_{2}^{n-1+1} K \log K \, dK}$.

$\sum_{K=2}^{n-1} K \log K \leq \int_{2}^{n-1+1} x \cdot \log x \cdot dx$.

$\leq \left[\log x \, \frac{x^2}{2} - \left(\int \frac{x^2}{2} \frac{1}{x} \, dx\right)\right]_2^n$

$\leq \left[\frac{x^2}{2} \log x - \frac{1}{2} \int x \cdot dx + C\right]_2^n$

$\leq \left[\frac{x^2}{2} \log x - \frac{1}{2} \frac{x^2}{2} + C\right]_2^n$

$\leq \left[\frac{x^2}{2} \log x - \frac{x^2}{4} + C\right]_2^n$

$\leq \left[\frac{1}{2} n^2 \log n - \frac{1}{4} n^2\right] - \left[\frac{2^2}{2} \log 2 - \frac{2^2}{4}\right] + C$

$$\leq \frac{1}{2} n^2 \log n - \frac{1}{4} n^2 - \underbrace{2 + c}_{\text{constant}}$$

$$\leq \frac{1}{2} n^2 \log(n) - \frac{n^2}{4}$$

$$\therefore \sum_{k=2}^{n-1} k \log k \leq \frac{1}{2} n^2 \log(n) - \frac{n^2}{4}$$

2b) $\log(n!) = O(n \log n)$

$$\log(n!) = \log(1 \times 2 \times 3 \cdots (n-1) \times n)$$

$$= \log 1 + \log 2 + \log 3 \cdots\cdots + \log n$$

$$= \sum_{x=1}^{n} \log x$$

From the monotonically increasing function,

$$f(k) = \log x$$

$$\sum_{x=1}^{n} \log x \leq \int_1^{n+1} 1 \cdot \log x \cdot dx$$

$$\leq \left[x \log x - x + c \right]_1^{n+1}$$

$$\leq (n+1(\log(n+1) - 1)) - 1 \log 1 - 1.$$

$$\leq n+1\left(\log(n+1)-1\right) - (-1)$$

$$\leq n+1 \cdot \left(\log(n+1) - 1\right) + C$$

$$\leq (n+1)(\log n - 1)$$

$$\leq n\log n + \log n - (n+1)$$

$$\leq O(n\log n)$$

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

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

Q3 (a) Loop Invariant:

In the selection sort, elements of Array A, at the start of each iteration of the loop the Sub array A[1 . . . i-1] consists of the i-1 smallest elements of the original array A, in sorted order.

Initialization:

Before the first iteration of the Outer loop, i=1. Thus the subarray A [1, . . i-1] is empty. It contains the 0 smallest elements of A.

The loop invariant holds at initialization.

Maintenance:-

→ At the start of the current iteration of the outer loop, for some i>1 the subarray A[1 . . i-1] containing the i-1 smallest elements of A in sorted order.

During the iteration, inner loop finds the minimum ~~sorted~~ element in the unsorted portion of the array A [i, . . n] and swaps it with A[i].

After this iteration, the subarray $A[1..i]$ contains the 'i' smallest elements of A in sorted order.

∴ Thus, after the iteration, loop invariant holds for 'i+1'.

Termination:- When outer-loop terminates after (n-1 iterations), $i = n$. As per the loop invariant, the subarray $A[1..n-1]$ contains the $n-1$ smallest elements of A in sorted order.

There will be no elements after $A[n]$, it must be the largest element of A.

∴ Entire array A is sorted.

∴ Algorithm is correct.

(b) No. of comparisions :-

→ Input of size n, number of comparisions can be expressed as,

$$\sum_{i=1}^{n-1} (n-i) = \frac{n(n-1)}{2}.$$

(from class notes)

Number of swaps:-

Considering a random variable X_i indicating the no. of swaps happend in iteration i.

$$E[X_1] = 1 - \frac{1}{n} \quad - i = 1.$$

$$= 1 - \frac{1}{n-1} \quad - i = 2.$$

$$E[X_i] = 1 - \frac{1}{n-i+1}$$

$$X = \sum_{1}^{n-1} X_i$$

$$E[X] = E\left[\sum_{1}^{n-1} X_i\right] = \sum_{1}^{n-1} E[X_i]$$

$$= \sum_{1}^{n-1}\left(1 - \frac{1}{n-i+1}\right) = \left(1-\frac{1}{n}\right) + \left(1 - \frac{1}{n-1}\right) + \cdots \left(1-\frac{1}{3}\right) + \left(1-\frac{1}{2}\right)$$

$$= (n-1) - \frac{1}{n} - \frac{1}{n-1} - \cdots - \frac{1}{3} - \frac{1}{2}$$

$$= n - \left(\frac{1}{n} + \frac{1}{n-1} + \cdots \frac{1}{3} + \frac{1}{2}\right)$$

$$= n - \log n + O(1).$$

Q4. Bubble Sort:-

There are two-loops in the given algorithm. 2-5 Outer loop and inner loop 3-5.

Proof of correctness needs to be written for both the loops.

Inner-loop During the start of each iteration, the smallest among the elements will be among A [j ... n]

Intialization:- If at the start, of inner loop, j = n then the sub-array A[j..n] consists of elements A[j] or A[n] i.e., A[j] = A [n]. This will be the smallest element.

Maintenance:- At the j^{th} (inner-loop) A[j] is the smallest elements in the sub-array A[j..n]. In Bubble sort, it compares the A[j] with A[j-1] and identifies smallest elements. So at j-1 iteration, A[j-1....n]

Scanned with CamScanner

has smallest element.

Termination:- when $j=1$, loop exits. The smallest element can reside in $A[i]$

Outer-loop:-

Intialization: Begining of $i=1$, iteration executes the inner loop, $j=n$ to 2. Small element will be $A[i]$.

Maintenence:- At the end of ith iteration $A[1], A[2] \ldots A[i-1]$ they've the elements. The ending case from the inner loop smallest element will be at $A[i]$.

Termination:- The last iteration $i=n-1$, elements will be in sorted order in $A[1 \ldots n-1]$. The bubble sort algorithm of the elements are sorted.

No. of comparisions :- This will be fixed. regardless of input size. Outer iterations will be $(n-1)$ comparisions. Inner loop has $\frac{(n-2)}{2}$ comparisions.

$\therefore$ Total. $(n-1) + (n-2) + \cdots 2 + 1 = \frac{n(n-1)}{2}$.

Swaps that are Expected :-

$$E\left[\sum_{i=1}^{n-1} \sum_{j>1}^{n} I_{ij}\right] = \sum_{i=1}^{n} \sum_{j>1}^{n} E[I_{i,j}].$$

$$= \frac{1}{2} \times \frac{n(n-1)}{2}$$

$$= \frac{n(n-1)}{4}.$$

Q5 Induction proof

Begins with $j = m-1$ the hypothesis holds for $j = k-1$.

Invariant :- At the end of jth iteration any element in the set S is added with probability $m-j / n-j$

Initialization:- $j = m-1$, the only element S is added with probability of $\frac{1}{n-(m-1)} = \frac{m-(m-1)}{n-(m-1)}$

Maintenence:- If holds for $j=k$, when iteration finishes, the mean of any element in set S is added with prob $\left(\frac{m-k}{n-k}\right)$.

Now $j=k-1$, for any element added with probability of $\frac{m-(k-1)}{n-(k-1)}$ after iteration with $j=k-1$ finishes. S has 2 elements.

S has $m-(k-1) = (m-k+1)$ elements.

Case 1 $\quad n = m-(k-1)$.

for $j=k-1$ iteration. the total probability will be $\frac{1}{n-k+1} + \frac{m-k}{n-k+1}$

$$= \frac{m-(k-1)}{n-(k-1)}.$$

Scanned with CamScanner

Case 2 Element either have been added in $j=k$ with probability $\frac{m-k}{n-k}$ i.e., Induction hypothesis.

If its not elected in earlier iterations and selected in last iteration..

$$S = \frac{1}{n-m+1} \cdot \frac{2}{n-m+2} \cdots \frac{m}{n}$$

$$= \frac{1}{n_{\text{given } m}}$$

$\therefore$ $O(m)$.

6) a) $T(n) = 4T(n/2) + n$.

Recurrence method,

$$T(n) = 4T(n/2) + n$$

Recursively applying same formula for $n/2$

$$\Rightarrow \quad T(n) = 4\left(4T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n$$

$$= 4^2 \cdot T(n/2^2) + 2n + n$$

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

$$= 4^2\left(4 \cdot T \cdot \left(\frac{n}{2^2 \times 2}\right) + \frac{n}{2^2}\right) + 2n + n$$

$$= 4^3 \cdot T\left(\frac{n}{2^3}\right) + 4n + 2n + n.$$

$$= 4^h \cdot T\left(\frac{n}{2^h}\right) + n\left(1 + 2^1 + 2^2 + \cdots 2^{h-1}\right)$$

[∵ Assumed $2^h = n$ the base case

$\frac{n}{2^h} = 1 \Rightarrow h = \log_2 n$]

$$\therefore T(n) = 4^{\log_2 n} \cdot T\left(\frac{n}{2^{\log_2 n}}\right) + n(2^h - 1)$$

[∵ $n(1 + 2^1 + 2^2 + \cdots 2^{h-1}) = 1 \cdot \frac{2^h - 1}{2 - 1} = 2^h - 1$.

$a \cdot \frac{r^n - 1}{r - 1}$ finite geometric series]

$$T(n) = n^{\log_2 4} T\left(\frac{n}{n}\right) + (2^{\log_2 n} - 1)\, n$$

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

$$= n^2 \cdot 1 + n^2 - n$$

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

$$= O(n^2)$$

$T(n) = 4 \cdot T(n/2) + n$ is upperbounded by n^2 i.e., $O(n^2)$ [big-oh of n^2]

b) Given $T(n) = 2 \cdot T(n/2) + n \log_2 n$.

Recursively applying for $n/2$ gives

$$T(n) = 2\left(2 \cdot T\left(\frac{n}{2^2}\right) + \frac{n}{2} \log_2 \frac{n}{2}\right) + n \log_2 n$$

$$= 2^2 T(n/2^2) + n \cdot \log_2 \frac{n}{2} + n \log_2 n$$

$$\left(\because \log_2 \frac{n}{2} = \log_2 n - \log_2 2 = \log_2 n - 1\right)$$

$$= 2^2 \cdot T(n/2^2) + n(\log_2 n - 1) + n \log_2 n$$

$$= 2^2 \cdot T(n/2^2) + n(\log_2 n - 1) + n(\log_2 n - 0)$$

$$= 2^2 \left[2 \cdot T\left(\frac{n}{2^3}\right) + \frac{n}{2^2} \log \frac{n}{2^2}\right] + n \cdot (\log n - 1) +$$

$$= 2^3 \cdot T\left(\frac{n}{2^3}\right) + n \cdot \log_2\left(\frac{n}{2^2}\right) + n(\log_2 n - 1) + n(\log_2 n - 0$$

$$= 2^3 \cdot T(n/2^3) + n(\log_2 n - 2) + n(\log_2 n - 1) + n(\log_2 n - 0$$

$$= 2^h \cdot T\left(\frac{n}{2^h}\right) + n(\log_2 n - (h-1)) + \cdots n(\log_2 n - 0$$

Base case $\frac{n}{2^h} = 1 \Rightarrow h = \log_2 n$

$$T(n) = 2^{\log_2 n} \cdot T\left(\frac{n}{2^{\log_2 n}}\right) + n(\log_2 n - (\overset{h-1}{\overline{\log_2 n - 1}}))$$

$$- - - + n \cdot (\log_2 n - 0)$$

$$= n \cdot T(1) + n \cdot \log_2 n \, (\underline{1 + 1 + 1 + \cdots + h}) - n \times$$

$$(0 + 1 + 2 - - h - 1)$$

$$= n + nh \cdot \log_2 n - n \frac{h(h-1)}{2}$$

$$= n + n \log_2 n \log_2 n - n \frac{\log_2 n (\log_2 n - 1)}{2}$$

$$= n + n \cdot \log_2^2 n - \frac{n}{2} \log_2 n (\log_2 n - 1)$$

$$= n + n \log_2^2 n - \frac{n}{2} \log_2^2 n + \frac{n}{2} \log_2 n$$

$$= O(n \cdot \log_2^2 n)$$

So Upper bounded by $n \cdot \log_2^2 n$.

Scanned with CamScanner

7. a. $T(n) = 4 \cdot T(n/2) + n^2 \cdot \sqrt{n}$.

$T(n) = a \cdot T(n/b) + f(n)$

Step 1:-

$a = 4 \quad b = 2 \quad f(n) = n^2 \cdot \sqrt{n}$

$(a \geq 1) \quad (b > 1)$

$n^{\log_b a} = n^{\log_2 4} = n^2$.

$f(n) = n^2 \sqrt{n} = n^{2.5}$

Step 2 : It falls under case 3 of Master method,

$f(n) = \Omega(n^{2+\epsilon}) \quad \forall \; \epsilon \text{ in } (0 - 0.5)$

Step 3 : $a \cdot f(n/b) = 4 \cdot f(n/b)^{2.5}$

$= 4 \cdot f\left(\frac{n}{2}\right)^{2.5}$

$= \cancel{4} \cdot \frac{n^{2.5}}{\cancel{4}\sqrt{2}}$

$\therefore a f(n/b) = \frac{1}{\sqrt{2}} n^{2.5}$

Step 4 :

$a \cdot f(n/b) \leq c \cdot f(n) \quad \exists \; c < 1$

$\frac{1}{\sqrt{2}} \cdot n^{2.5} \leq c \cdot f(n) \quad \forall \; c = \frac{1}{\sqrt{2}} < 1$

step 5 :- $T(n) = \Theta(f(n))$

Scanned with CamScanner

$\therefore T(n) = O(n^2\sqrt{n})$

Substitution Method:-

Assuming that given $T(n)$ is upper-bounded.

$$T(n) = O(n^2 \cdot \sqrt{n})$$

We've to check, whether, $T(n) = O(n^2\sqrt{n})$ is less than/equal to $c \cdot (n^2\sqrt{n})$ i.e.,

$$T(n) = O(n^2\sqrt{n}) \leq c \cdot (n^2\sqrt{n}).$$

$$= 4c \cdot \frac{n^2}{4} \cdot \sqrt{\frac{n}{2}} + n^2 \cdot \sqrt{n}$$

$$= \frac{c \cdot n^2\sqrt{n}}{\sqrt{2}} + n^2\sqrt{n}$$

$$= c\,\cancel{n^2\sqrt{n}} - \cancel{n^2\sqrt{n}\,c} \cancel{+ n^2\sqrt{n}} + \cancel{\frac{n^2\sqrt{n}\,c}{\sqrt{2}}}$$

$$= c \cdot \frac{n^2\sqrt{n}}{2} + n^2\sqrt{n} + c \cdot n^2\sqrt{n} - c\,n^2\sqrt{n}$$

$$= c \cdot n^2\sqrt{n} - n^2\left(c\sqrt{n} - \sqrt{n} - c\frac{\sqrt{n}}{2}\right)$$

$$= c \cdot n^2\sqrt{n} - n^2\sqrt{n}\left(c - 1 - \frac{c}{\sqrt{2}}\right)$$

~~Anything subtracted by $c^2 n\sqrt{n}$~~

$$\therefore T(n) \leq c \cdot n^2\sqrt{n} \quad \forall\ n \geq 1\ \&$$

$$c \geq \frac{\sqrt{2}}{\sqrt{2} - 1}$$

$\therefore\ T(n) = O(n^2\sqrt{n})$

upper-bounded by $n^2\sqrt{n}$.

b) $T(n) = 4 \cdot T(n/3) + n\log n$

Step 1:- $a=4$ $(a \geq 1)$ $b=3$ $(b>1)$ $f(n) = n\log n$

$n^{\log_b a} = n^{\log_3 4}$
$\simeq 1.262$. (from internet $n\log_3 4 = 1.262$)

Step 2:-
assume, $f(n) = O(n^{\log_b a - \epsilon})$

$f(n) = n\log n = O(n^{\log_3 4 - \epsilon})$

If ϵ is in range of $0 < \epsilon < 0.262$ then $T(n)$ falls under case 1
i.e.,
$T(n) = \Theta(n\log_b a)$

$= \Theta(n\log_3 4)$.

Substitution

-bound

7 (c) $T(n) = 2 \cdot T(n/2) + n\lg n$.

Step 1 $a = 2$. $b = 2$. $f(n) = n\log n$.

$n^{\log_b a} = n$ &

Step 2 :-

$f(n) = n\log n$ which is ~~less~~ greater than

$O(\cancel{n}) = O(n^{\log_b a - \epsilon})$

This doesn't fall under any of the Master theorem.

Substitution Method :- (Guess work)

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

$T(n) \leq 2 \cdot c \frac{n}{2} \log \frac{n}{2} + n\log n$

$\leq c \cdot n(\log n - 1) + n\log n$

$\leq cn\log n - cn + n\log n$

$\leq n\log n(c+1) - cn$.

for $n \geq 1$ and $c \geq 1$

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

7(d) $T(n) = 2T(n/4) + \sqrt{n}$.

Step 1:- $a = 2$ $b = 4$ $f(n) = \sqrt{n}$.

$$n^{\log_b a} = n^{\log_4 2} = n^{\log_{2^2} 2} = n^{\frac{1}{2}\log_2 2}$$

$$= \sqrt{n}.$$

Step 2

$$f(n) = \sqrt{n} = n^{\log_b a}.$$

Master Method case (2) applied,

$f(n) = \Theta(n^{\log_b a})$

$$\therefore T(n) = \Theta(n^{\log_b a} \cdot \log n)$$

$$T(n) = \Theta(\sqrt{n} \log n).$$