★ Order of function

- Insertion sort → $O(n^2)$.
- Merge sort → $O(n \log n)$
- counting sort → $O(n)$

sorting

Ex:-

n	n^2	$n \log_{10} n$	2^n
1	1	0	2.
10	100	10	$2^{10} = 1024$.
100	10^4	200	$2^{100} = 1.26 \times 10^{30}$
1000	10^6	3000	2^{1000} = large.
10^4	10^8	4×10^4.	V.V. large.

Order of function

$n < n \log n < n^2 < 2^n$

$\log n$; n^3 ; $\log(n!)$

~~So~~ $\Rightarrow 1 < \log_2 n < \sqrt{n} < n < n\log_2 n < n^2 < 2^n < n!$

$\star$ $O(1) < (\log^*(n)) < O(\log\log(n)) < O(\log n) < O(\log n)^c <$
$\underbrace{O(n^c)}_{(0<c<1)} < O(n) < O(n\log^* n) < [O(n\log n) = O(\log(n!))]$
$< O(n^2) < O(n^c) < \underbrace{O(c^n)}_{(c>1)} < O(n!)$

x	$\log^* x$.
$(-\infty, 1]$	0
$(1, 2]$	1
$(2, 4]$	2
$(4, 16]$	3
$[16, 65536]$	4
$(65536, 2^{65536}]$	5.

$\star$ Why does asymptotic analysis matter in the reality.

$\boxed{10^9 ns \Rightarrow 1 sec}$ $30 \times 10^9 ns \Rightarrow 30$ sec.

n	$\log_2 n$	n	$n\log_2(n)$	n^2	2^n.
10^3	9.96.	10^3	9965.78	10^6	1.07×10^{301}
10^6	19.93.	10^6	1.99×10^7	10^{12}	V.V. Large
10^9	$\underbrace{29.89}_{\approx 30 ns}$	$\underbrace{10^9}_{1 sec.}$	$\underbrace{2.98 \times 10^{10}}_{30 sec.}$	$\underbrace{10^{18}}_{10^9 sec.}$	V.V. Large.

* (a) $f(n) = n^2+n+1$
$f(n) = O(g(n))$
$f(n) = \Omega(h(n))$.
$f(n) = \Theta(k(n))$.

What are the valid function for $g(n), h(n), k(n)$?

$\Rightarrow f(n) = n^2+n+1$ $\quad O(g(n))$ $\quad$ Let $g(n) = n^2$.

$f(n) \leq C * g(n) \quad \forall n \geq n_0$.

$n^2 \leq n^2$.

$n^2+n+1 \leq n^2+n^2+n^2 \quad \forall n \geq 1$

$n^2+n+1 \leq \underset{\uparrow \; C}{3}n^2. \quad \forall n \geq 1 \quad \boxed{n_0 = 1}$

So $f(n) = O(n^2)$. $\quad \underline{\underline{g(n) = n^2}}$.

$\Rightarrow f(n) = n^2+n+1$ $\quad$ Let $g(n) = n^3$.

$n^2 \leq n^3 \quad \forall n \geq 1$

$n^2+n+1 \leq n^3+n^3+n^3. \quad \forall n \geq 1$

$n^2+n+1 \leq \underset{\uparrow \; c}{3}n^3. \quad \forall n \geq \underset{\uparrow \; n_0}{1}$. $\quad$ So $\underline{\underline{g(n) = n^3}}$.

$\Rightarrow f(n) = n^2+n+1$ $\quad$ Let $\underline{g(n) = n}$. X

$n^2+n+1 \leq C.n \quad \forall n \geq n_0$.

$\boxed{n+1+1/n \leq C.} \Rightarrow$ X $\quad f(n) \neq O(n)$.

So $g(n) = n^2$; $g(n) = n^3$; $g(n) = n^4$; ......

$\Rightarrow f(n) = \Omega(h(n))$. $\quad$ Let $h(n) = n^2$.

$C.h(n) \leq f(n)$.

$\forall n \geq n_0$.

$n^2+n+1 \geq C.n^2$.

So $\underline{\underline{h(n) = n^2}}$.

$f(x) = n^2+n+1$. $\quad$ Let $h(n) = n$.

$n^2+n+1 \geq C.n. \quad \forall n \geq n_0$.

So $\underline{h(n) = n}$ ✓

So $f(n) = \Theta(K(n))$
iff $f(n) = O(K(n))$ and $f(n) = \Omega(K(n))$.

So:- $g(n) = n^2$; $g(n) = n^3$; $g(n) = n^4$; ...... (O)
$h(n) = n^2$; $h(n) = n$. (Ω)
So $K(n) = \underline{n^2}$. (Θ)

polynomial of degree n.
So ⇒ if $f(n) = a_0 + a_1 n + a_2 n^2 + a_3 n^3 + \ldots + a_m n^m$.
(O) ⇒ $f(n) = O(n^m)$; $O(n^{m+1})$; $O(n^{m+2})$ $k \geq 0$.
(Ω) ⇒ $f(n) = \Omega(n^m)$; $\Omega(n^{m-1})$; $\Omega(n^{m-2})$; ...... $\Omega(n^{m-k})$
(Θ) ⇒ $f(n) = \Theta(n^m)$. Shortcut to remember.

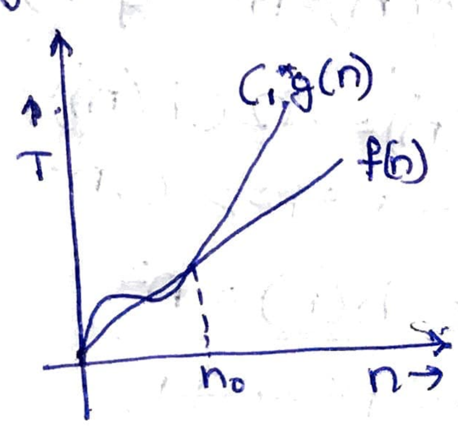
Q. $f(n) = \begin{cases} n^2 & n \leq 100 \\ n. & n > 100 \end{cases}$ $g(n) = \begin{cases} n & n < 1000 \\ n^3 & n \geq 1000. \end{cases}$

Tick correct one
✓ a) $f(n) = O(g(n))$ b) $g(n) = O(f(n))$.

→ So by Def :- $f(n) \leq Cg(n)$ $\forall$ $n \geq n_0$
$\exists$ c, n_0 → $f(n) = O(g(n))$.

So Let $n_0 = 1000$. $f(n) = n$
So. $n \geq 1000$ $g(n) = n^3$.

So $f(n) \leq Cg(n)$
$n \leq C.n^3$.
So $f(n) = O(g(n))$ ✓



Q) Let $f(n) = n^2 \log n$ and $g(n) = n(\log n)^{10}$ be 2 positive function of n. Which of the following statement is correct.

a) $f(n) = O(g(n))$ and $g(n) \neq o(f(n))$.

✓b) $g(n) = o(f(n))$ and $f(n) \neq O(g(n))$.

c) $f(n) \neq O(g(n))$ and $g(n) \neq o(f(n))$.

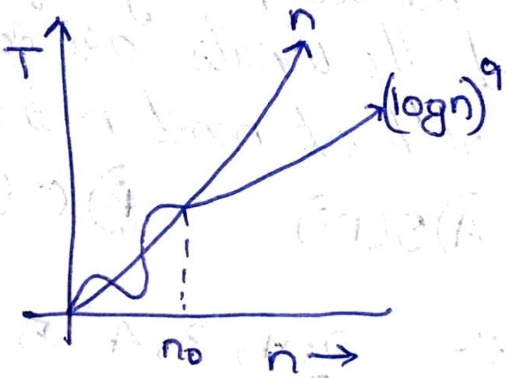
d) $f(n) = o(g(n))$ and $g(n) = o(f(n))$

→ so according to order of function

$f(n) = n^2 \log n$ $\qquad$ $g(n) = n(\log n)^{10}$.

$\Rightarrow n(\cancel{n \log n})$. $\qquad$ $(\cancel{n \log n})(\log n)^9$.

so $n > (\log n)^9$.



$\underline{\underline{c > 1}}$.

so $f(n) > g(n)$

so according to O

$g(n) = o(f(n))$.

$f(n) \neq o(g(n))$.

Q) $f_1(n) = 2^n$; $f_2(n) = n^{3/2}$; $f_3(n) = n\log_2 n$; $f_4 = n^{\log_2 n}$.

What is the increasing order of asymptotic complexity?

→ $f_3 = n \log n$ $\qquad$ $f_2 = n^{3/2}$. $\qquad$ $f_3 < f_2$.

$\cancel{n} \log n$ $\qquad$ $\cancel{n} n^{1/2}$.

$\log n < n^{1/2}$.

→ $f_2 = n^{3/2}$ $\qquad$ $f_4 = n^{\log_2 n}$ $\qquad$ $f_2 < f_4$.

$3/2 < \log_2 n$. $\qquad$ $f_3 < f_2 < f_4$.

$n^{3/2} < n^{\log_2 n}$.

$f_1 = 2^n$ > $f_4 = n^{\log_2 n}$

$n \log_2 2$ $\quad$ $\log n^{\log_2 n}$.

$\Rightarrow \log_2 n (\log_2 (n))$.

n.

n > $\Rightarrow (\log_2 n)^2$.

$f_1 > f_4$.

So the increasing order is $f_3 < f_2 < f_4 < f_1$

Q) In a permutation $a_1 \ldots a_n$ of n distinct integer an inversion is a pair (a_i, a_j) such that $i<j$ and $a_i > a_j$. What would be the worst case time complexity of the Insertion Sort algorithms if the inputs are restricted to permutation of $1 \ldots n$ with at most n inversions?

A) $O(n^2)$ $\quad$ B) $O(n \log n)$ $\quad$ C) $O(n^{1.5})$ $\quad$ ✓d) $O(n)$

Ex:- $2 \to 1, 3, 4, 5$ $\rightarrow (2,1)$ Inversion pair, 1 move to right 1 Swap.

$2 \to 3 \to 1, 4, 5$ $\rightarrow (2,1)(3,1)$ — " — " — 2 move to right 1 Swap.

$2 \to 3 \to 4 \to 5 \to 1$ $\rightarrow (2,1), (3,1), (4,1), (5,1)$ — " — " — 4 move to right 1 Swap.

Worst case.

Time Complex : $O(n + m)$

n → Testing each element.

m → # No. of Inversion in array.

So $O(n + n)$

$O(2n) \Rightarrow \underline{O(n)}$.

2 → is constant

Q) 1) $(n+K)^m = O(n^m)$ ✓
2) $2^{n+1} = O(2^n)$ ✓ where K and m are constants
3) $2^{2n+1} = O(2^n)$ X which of them are true ?

1) $(n+K)^m = n^m + {}^mC_1 n^{m-1} K^1 + {}^mC_2 n^{m-2} K^2 + \ldots + K^m$

$\Rightarrow n^m + C_1 n^{m-1} + C_2 n^{m-2} + \ldots + C_m n^0$

$\Rightarrow n^m$.

$(n+K)^m = O(n^m)$ True. ✓

2) $2^{n+1} = O(2^n)$

$2.2^n \leq C.2^n$ $\exists C, n_0$. $\forall n \geq n_0$.

So let $C=3$.

$2 \cancel{2^n} \leq 3.\cancel{2^n}$ $2 \leq 3$, $\forall n \geq 1$

$2^{n+1} = O(2^n)$ True ✓.

3). $2^{2n+1} = O(2^n)$

$2^{2n+1} \leq C \cdot 2^n$. $\exists C, n_0$

$2^{2n} 2 \leq C.2^n$. $\forall n \geq n_0$.

$2^{2n} \cancel{2} \leq \cancel{2} 2^n$ Let $C=2$

$2^{2n} \leq 2^n$

$2n < n$. So. $2^{2n+1} \neq O(2^n)$ False

Q) Consider the following function:

- $f(n) = 2^n$.
- $g(n) = n!$
- $h(n) = n^{\log n}$

$n^{\log n} < 2^n < n!$

$h(n) < f(n) < g(n)$.

Which of the following statement about the asymptotic behaviour of $f(n)$, $g(n)$ and $h(n)$ is true?

A) $f(n) = O(g(n))$; $g(n) = O(h(n))$

B) $f(n) = \Omega(g(n))$; $g(n) = O(h(n))$

C) $g(n) = O(f(n))$; $h(n) = O(f(n))$

D) $h(n) = O(f(n))$; $g(n) = \Omega(f(n))$

Q) $10, \sqrt{n}, n, \log_2 n, \frac{100}{n}$.

Order them by asymptotic complexity

$\Rightarrow$. We K.T $\frac{100}{n} < 10 < \log_2 n < \sqrt{n} \leq n$ Order of function

So if $n\uparrow$; so stays the same.

So if $n\uparrow$; $\frac{100}{n\uparrow}$ $\Rightarrow$ Value $\downarrow$ less the 10.

Let $f(n) = \frac{100}{n}$ $g(n) = 10$

$\left(\frac{100}{n}\right) = O(10)$. $\Rightarrow f(n) = O(gn)$.

$\frac{100}{n} \leq C.10$. $\forall n \geq n_0$.

$\exists C, n_0$.

Let $C=1$ $n_0 = 10$.

$\{ 10 \geq \frac{100}{n}$ $\forall n \geq 10 \}$. ✓

Q).- Multi Dequeue (Q).

```
{ m=K.
  while ((Q is not empty) and (m>0)).
     {m=m-1}
}.
```

What is the worst case time complexity of a sequence of n queue operations on an empty queue?

A) $O(n)$ ✓

B) $O(n+k)$

C) $O(n*k)$.

D) $O(n^2)$

So worst case is the order of n itself

$\rightarrow O(n)$.