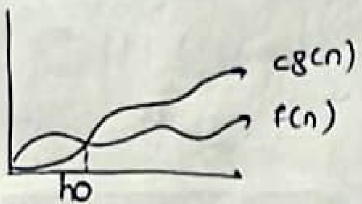
Asymptotic Notion

Analyse pseudocode :

1. worst case - O
2. Avg - Ω
3. Best - Θ

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

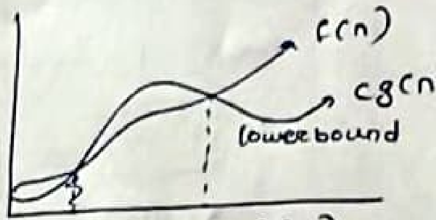
worst -



$0 \le f(n) \le cg(n) \ \forall\ n \ge n_0$

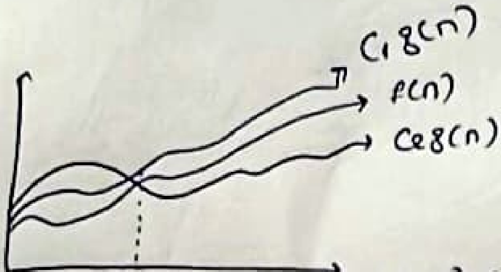
$f(n) = O(n^2)$

Best



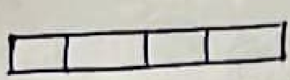
$f(n) = \Omega\, g(n)$

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



$0 \le c_2 g(n) \le f(n) \le c_1 g(n)$

eg.



← search

∴ Best case $\Omega(1)$

worst case $O(n)$

Avg case $\Theta(n)$

$f(n) = a_0 + a_1 n + a_2 n^2 + \cdots + a_n n^m$

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

$f(n) = c \cdot g(n)$

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

Transitive prop.

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

$g(n) = O\, h(n)$

Summat^n property

$f(n) = O\, h(n)$

$f(n) + g(n) = O(\max(f(n), g(n$

log prop.

$f(n) = \log_a n$

$g(n) = \log_b n$

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

$\log^2 n$
$\log n^2$ } Diff
$(\log n)^2$

---

eg. ① A {
for (i=1; i<=n; i++) — $O(1)$, $O(n+1)$ — $O(n)$
pf() — $O(n)$
}

→ $O(n) \equiv 1+n+1+n+n$

---

②
```
for (i=1, i<=n ; i=i+2)
{ pf()
}}
```
→ $O(n/2) \equiv O(n)$

```
for (i=1; i<=n; i=i*2)
```
$\log_2 n$

$\boxed{2^k > n \equiv k = \log_2 n}$

---

③
```
A() {
 i=1
 while (i<=n)
 { i=i*3;
 }}
```
→ $\log_3 n$

---

④
```
for (i=1 ; i<n ; i++)
  for (j=1; j<n ; j++)
  { pf ()
  }
```
→ $O(n^2)$

---

⑤
```
for (i=1 ; i<=n ; i++)
  for ( j=1; j<=n ; j++)
  { pf()
  }
```
→ $O(n^2)$

---

⑥
```
int i,j;
  for( i=n; i>1; i=i/2)
  { j=n
    while (j>1)
    { j=j/2 })
```
→ $\log_2 n * \log_2 n$

$(\log_2 n)^2$

① A()

```
{ i=1      s=1 ;
  while ( s<=n)
  { i++;
    s= s+i;
    PF() }}
```

i: 2 3 4 5
3 6 10 15 --- k

$\rightarrow \boxed{O(\sqrt{n})}$

$\frac{k(k+1)}{2} > n$

$\therefore k^2 > n$

②

```
A(){ int i,j,k,n;
  for ( i=1 ; i<=n ; i++)
    for ( j=1 ; j≤i ; j++)
      for ( k=1; k≤100; k++)
  }  PF()
```

i = 1, 2, 3 --- n
j = 1times, 2times, 3 ---- n
k = 1*100, 2×100, 3×100, n×100

$100+200+300+---+n\times 100$

$100(1+2+---+n)$

$100 * \frac{n(n+1)}{2}$

$\therefore \boxed{O(n^2)}$

③

```
A(){ int i,j,k;
  for ( i=1 ; i≤n ; i++)
  for ( j=1; j≤i²; j++)
  for ( k=1 ; k≤n/2 ; k++)
    PF()
}
```

i = 1, 2, 3 ---- n
j = 1, 4, 9, n
k = $\frac{n}{2}$, $4\times\frac{n}{2}$, $9\times\frac{n}{2}$, $n^2 * \frac{n}{2}$

$\frac{n}{2}\left[1+4+9 --- +n^2\right]$

$\frac{n}{2} * \frac{n(n+1)(2n+1)}{6}$

$= O(n^4)$

$\rightarrow \boxed{O(n^4)}$

④

```
for (int i=n/2 ; i<=n ; i++)    --- O(n/2)
for( j=1 ; j≤ n/2 ; j++)       --- O(n/2)
for (k=1; k≤n ; k=k*2)         --- O(log₂n)
```

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

{ for (i = ~~n/2~~ 1 ; i ≤ n ; i++)
for (j = 1 ; j ≤ n ; j = j+i)
pf()
}

→ along $\boxed{O(n \log n)}$

$i = 1 \quad 2 \quad 3 \quad \ldots\ldots$

$j = 1 * n \quad \frac{n}{2} \quad \frac{n}{3} \quad \ldots \quad 1$

$n(\underbrace{1 + 1/2 + 1/3 + 1/4 \ldots + 1}_{\text{HP series}})$

$\sum_{i=1}^{n} = \frac{1}{n}$

↑ log n + Euler const.

{ for (i = $\frac{n}{2}$; i ≤ n ; i++)
for (j = 1 ; j + $\frac{n}{2}$ ≤ n ; j++)
for (k = 1 ; k ≤ n ; k = k*2)
pf();
}

→ $\boxed{O(n^2 \log n)}$

Reccursive :

```
A(n)
{ if (n > 1)
  return A(n-1)
  else
  return 1
```

① Back substitution

$T(n) = T(n-1) + 1 \qquad n > 1$

$T(1) = 1 \qquad n = 1$

← Base condⁿ

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

$\therefore \quad T(n) = T(n-2)+2$

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

$\therefore \quad T(n) = T(n-3)+3$

$\vdots$

kth

$T(n) = T(n-k) + k \cdot 1$

Now $n-k=1$

$\therefore \quad k = n-1$

$\therefore \quad T(n) = T(n-k) + n-1$

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

$\therefore \quad O(N)$

Q $T(n) = T(n-1)+n \qquad n>1$

$T(1) = 1 \qquad n=1$

$\rightarrow \quad T(n-1) = T(n-2)+(n-1)$

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

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

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

$\vdots$

$T(n-k) = T(n-k)+(n-(k-1))+(n-(k-2))+ \cdots n$

$\therefore \quad n-k = 1$

$\boxed{k = n-1}$

$\therefore \quad = \cancel{1 + n - (n+1)}$

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

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

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

$O(N^2)$

Q $T(n) = 2T(n-1) - 1 \quad n>0$

$T(0) = 1 \quad n=0$

$\therefore T(n-1) = 2T(n-2) - 1$

$\therefore T(n) = 4T(n-2) - 1 - 1$

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

$T(n) = 8T(n-3) - 1 - 1 - 1$

$\vdots$

$T(K) = 2^K T(n-K) - K \cdot 1$

Mam

$2^K T(n-k) - 2^{K-1} - 2^{K-2} \dots - 2^0$

$n - K = 0$

$\boxed{K = n}$

$\therefore T(n) = 2^n - 2^{n-1} - 2^{n-2} \dots - 2^0$

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

$= O(1)$

Q $\left. \begin{array}{l} T(n) = 2T(n/2) + n \\ T(1) = 1 \end{array} \right\} \begin{array}{l} n>1 \\ n=1 \end{array}$

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

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

$T(n/4) =$

$O\ n(\log n)$

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

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

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

$$T(n/2^2) = 2T(n/2^3) + n/2^2$$

$$\therefore \quad T(n) = 2^2\left\{2T(n/2^3) + n/2^2\right\} + n + n$$

$$= 2^3 T(n/2^3) + n + n + n$$

$$\vdots$$

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

$$\therefore \quad \frac{n}{2^k} = 1$$

$$n = 2^k$$

$$k = \log_2 n$$

$$\therefore \quad 2^{\log_2 n} T(1) + \log_2 n . n$$

$$\therefore \quad n + \underline{n \log n}$$

$$T(n) = 2T(n/2) + n^2 \quad n > 1$$
$$1 \quad n = 1$$

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

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

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

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

$$\vdots$$

$$T(n/k) = 2^k\left(n/2^k\right) + \frac{n^2}{2^{k-1}} + \frac{n^2}{2^{k-2}} + \cdots + n^2$$

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$\boxed{\therefore k = \log n}$$

$$T(n/k) = 2^k + n^2\left(\frac{1}{2^{k-1}} + \frac{1}{2^{k-2}} + \cdots + 1\right)$$

$$2^{\log n}$$

$$n + n^2$$

$$\therefore \boxed{O(n^2)}$$

$\sqrt{n} \equiv n^{1/2}$

Q $T(n) = 2T\sqrt{n} + \log n$ $\quad n > 2$

$T(2) = 2$

$\longrightarrow$ $T(n) = 2T(n^{1/2}) + \log n$ ——— 1

$T(n^{1/2}) = 2T(n^{1/2^2}) + \log n^{1/2}$

$T(n) = 2\left[2T(n^{1/2^2}) + \log n^{1/2}\right] + \log n$

$T(n) = 2^2 T(n^{1/2^2}) + \log n^{1/2} + \log n$

$= 2^2 T(n^{1/2^2}) + \log n + \log n$

$T(n^{1/2^2}) = 2^2 T(n^{1/}$

$T(n) =$

$\vdots$

$T(n^{1/k}) = 2^k T(n^{1/2^k}) + k \log n$

$\therefore$

$n^{\frac{1}{2^k}} = 2$

$\frac{1}{2^k} \log n = \log_2 2$

$\log n = 1 \times 2^k$

$\boxed{k = \log \log n}$

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

$= 2 \log n + \log \log n * \log n$

$\boxed{O(\log n * \log \log n)}$

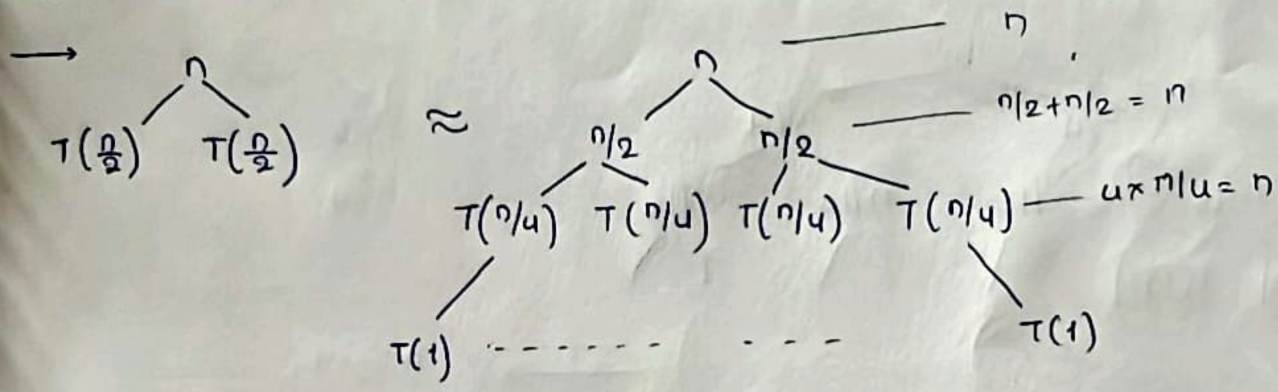
Tree method

Variations:

① $x^{\log_y n} = n^{\log_y x}$

② $x^0 + x^1 + x^2 + \cdots + x^n = \dfrac{x^{n+1} - 1}{x - 1}$

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



Total 2^k leaf nodes

$\therefore$ leaf cost $= 2^k$

$\therefore T\left(\frac{n}{2^k}\right) = T(1)$

$n = 2^k$

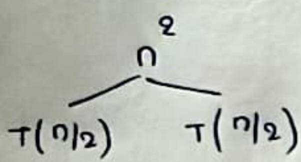
$\boxed{k = \log_2 n}$

- cost of internal node $= (k-1) * n$

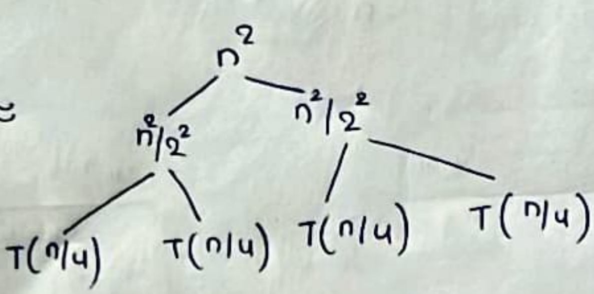
$= \log_2 n * n$

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

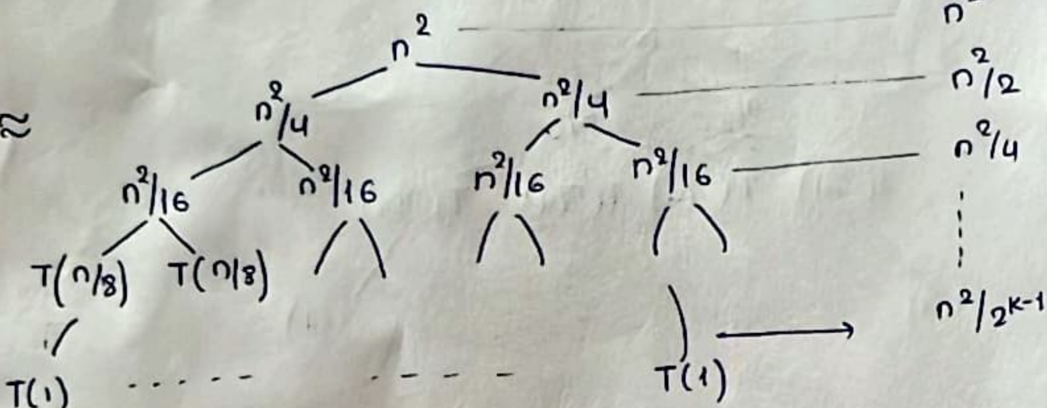
$T(1) = 1$



$\approx$



$\approx$



$n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \cdots + \frac{n^2}{2^{k-1}}$

$n^2\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{k-1}}\right)$

$n^2\left(\frac{1}{1-1/2}\right)$

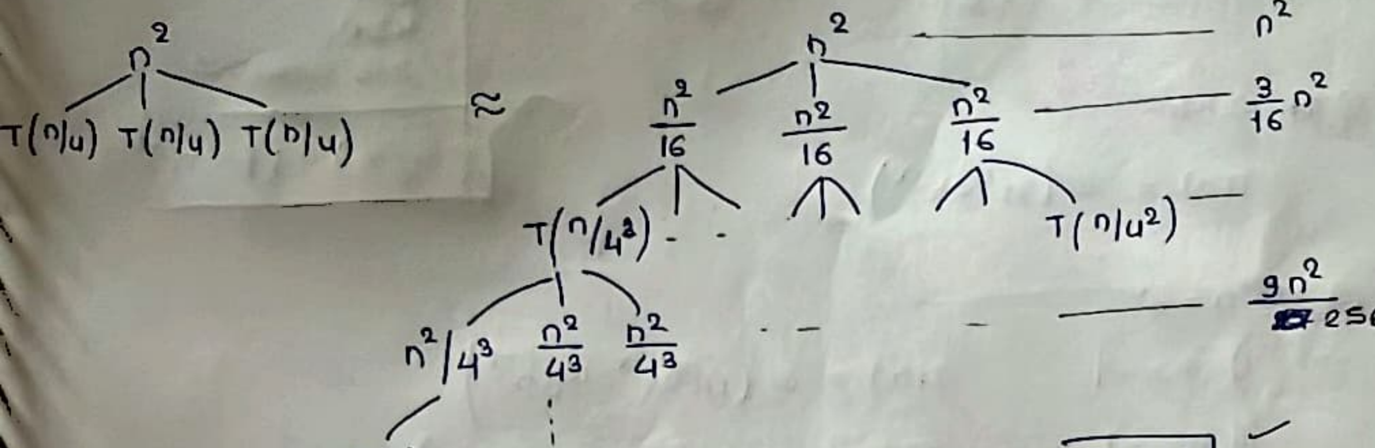
$2n^2$

$n + 2n^2$

$\therefore \boxed{O(n^2)}$

Q. $T(n) = 3T(n/4) + n^2 \quad n>1$

$T(1) = 1$



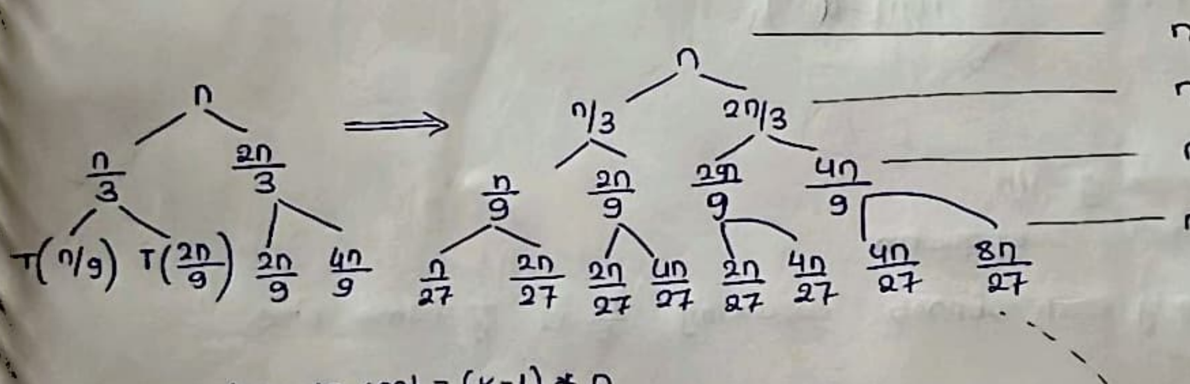
<u>Wrong</u>

$\boxed{O(n^2)}$ Ans

---

24 Jan 2024

Q $T(n) = T(n/3) + T(2n/3) + n \qquad T(1) = 1$



$\therefore$ Internal node cost $= (k-1) * n$

cost of leaf $= 2^k$ — coz each leaf node will have cost 1 & Total leaf node $= 2^k$

$T\left(n \cdot \left(\frac{2}{3}\right)^k\right)$

$\frac{2n}{3}$ — will have max Ht.

$\therefore \; n = \left(\frac{3}{2}\right)^k$

$\therefore \; \boxed{k = \log_{3/2} n}$

Total cost $= (k-1) * n + \text{cost of leaf}$

$= (\log_{3/2} n - 1)\, n + 2^{\log_{3/2} n}$

$= n\log_{3/2} n - n + n^{\log_{3/2} 2}$

$= O(n \log_{3/2}(n))$

Master Method

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + n^k \log^p n$$

$a \geq 1$
$b > 1$
$k \geq 0$
p is real No.

$\boxed{(\log^p n) \equiv (\log n)^p}$

① If $a > b^k$:

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

② If $a = b^k$

i) $p > -1$ $\quad T(n) = \Theta(n^{\log_b a} \cdot \log^{p+1} n)$

ii) $p = -1$ $\quad T(n) = \Theta(n^{\log_b a} \log \log n)$

iii) $p < -1$ $\quad T(n) = \Theta(n^{\log_b a})$

③ If $a < b^k$

i) $p \geq 0$ $\quad T(n) = \Theta(n^k \log^p n)$

ii) $p < 0$ $\quad T(n) = O(n^k)$

$n \longrightarrow$ size of i/p
$a \longrightarrow$ no of subproblem
$\frac{n}{b}$ → size of each subprob. assuming $\frac{n}{b}$ is of same size.

Q 1) $T(n) = 3T\left(\frac{n}{2}\right) + n^2$

$\longrightarrow$ $a = 3 \quad b = 2 \quad k = 2 \quad p = 0$

$b^k = 2^2 = 4$

$\therefore \boxed{a < b^k}$

case 3 (a)

$T(n) = \Theta(n^k \log^p n)$
$= \Theta(n^2 \log^0 n)$

$\boxed{T(n) = \Theta(n^2)}$

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

$\longrightarrow$ $a = 4, \quad b = 2 \quad k = 2 \quad p = 0$

$b^k = 4$

$\therefore \boxed{a = b^k}$

$\therefore$ case II $\quad T(n) = \Theta(n^{\log_b a} \log^{p+1} n)$

$T(n) = \Theta(n^{\log_2 4} \log n)$

$\boxed{T(n) = \Theta(n^2 \log n)}$

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

$a = 1 \quad b = 2 \quad k = 2 \quad p = 0$

$b^k = 4$

$\boxed{a < b^k}$

case 3 (a)

$T(n) = \Theta(n^k \log^p n)$
$= \Theta(n^2 \log^0 n)$

$\boxed{T(n) = \Theta(n^2)}$

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

$a = 2^n \quad b = 2 \quad K = n \quad P = 0$

$\therefore \boxed{a = b^k}$ case 2

$T(n) = \Theta(n^{\log_b^a} \log^{p+1} n)$

$= \Theta(n^{\log_2^{2^n}} \log n)$

$\boxed{T(n) = \Theta(n^n \log n)}$

} mam will confirm afterwards

NOT APPLICABLE

5) $T(n) = 16T(n/4) + n$

$a = 16 \quad b = 4 \quad K = 1 \quad P = 0$

$b^k = 4$

$\therefore \boxed{a > b^k}$

case 1

$T(n) = \Theta(n^k \log^p n)$

$= \Theta(n \log^0 n)$

$= \Theta(n)$

6) $T(n) = 2T(n/2) + n \log n$

$a = 2 \quad b = 2 \quad K = 1 \quad P = 1$

$\boxed{a = b^k}$ case 2 (a)

$T(n) = \Theta(n^{\log_b^a} \log^{p+1} n)$

$= \Theta(n^{\log_2^2} \log^2 n)$

$\boxed{T(n) = \Theta(n \log^2 n)}$

7) $T(n) = 2T(n/2) + \frac{n}{\log n}$

$a = 2 \quad b = 2 \quad K = 1 \quad P = -1$

$\boxed{a = b^k}$ case 2 (b)

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

$\boxed{T(n) = \Theta(n \log \log n)}$

check log property

8) $T(n) = 3T\left(\frac{n}{3}\right) + \sqrt{n}$

$a = 3 \quad b = 3 \quad K = 1/2 \quad P = 0$

$b^k = 3^{1/2} = 1.73$

$\therefore \boxed{a > b^k} \quad \therefore$ case 1

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

$= \Theta(n^{\log_3^3})$

$\boxed{T(n) = \Theta(n)}$

9) $T(n) = \sqrt{2}\,T\left(\frac{n}{2}\right) + \log n$

→ $a = \sqrt{2}$ $b = 2$ $k = 0$ $p = 1$

$b^k = 1$

$\boxed{a > b^k}$ case 1

$T(n) = \Theta\left(n^{\log_b a}\right)$

$= \Theta\left(n^{\log_2 2^{1/2}}\right)$

$= \Theta\left(n^{1/2}\right)$

$= \Theta(\sqrt{n})$

---

27 Jan 2025

1) $T(n) = 2^n T\left(\frac{n}{2}\right) + n^n$ ---- No sol^n by Master's method.

Space complexity

- variables
- prog inst^n
- funct^n call

Auxillary space + space for I/P

①

```
int s (int x, int y)
{ int q = x+y
  return q;
}
```

$\left.\begin{matrix} x \\ y \\ q \end{matrix}\right\}$ will take const space only : $\underline{O(1)}$

②

```
int add (int n)
  if (n<=0)
    return 0;
return n + add (n-1)
```

} It req stack for each f^n call : $\underline{O(n)}$

③

```
int add1 (int n)
{ int sum=0
  for (i=0 ; i<n ; i++)
     sum = sum+add (1,i+1)
  return sum;}

it add (int x . int y)
  return (x+y);
```

$\left.\begin{matrix} n \\ sum \\ i \end{matrix}\right\}$ $\underline{O(1)}$

④

```
int sum(int a[], int n)
{ int r =0;
  for (int i=0 ; i<n ; i++)
    { r += a[i];
    }
  return r;
}
```

a[] → n * 2bytes
n → 2 bytes
i → 2 bytes
return → 2 bytes

Explore 3 space complexity queist^n

Sorting Analysis

1. Bubble
2. Insertion
3. Selection ⟶ (Do it on your own, Tut work)
4. Merge
5. Quick
6. Heap

(Merge and Quick: Devide and conqure)

search
1. Linear
2. Binary

1. Bubble sort

80	40	18	22	60	54

comp

40	80	18	22	60	54

⋮

Best case - Already sorted $\Omega(n)$

worst case - Revensly sorted

$\underbrace{(n-1)+(n-2)+\cdots+1}_{\text{pass}}$

$\therefore \frac{(n-1)(n)}{2} = O(n^2)$

2. Insertion sort

worst -

key	comp	swap
1	0	0
2	1	1
3	2	2
⋮	⋮	⋮
n	(n-1)	(n-1)

Best case - O(n)

worst -

4. Merge :

Best - n log n
worst - n log n

5. quick -

Best - n log n
worst - $O(n^2)$

Merge eg

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

quick sort

$\boxed{10}$ 80 90 60 50 $\boxed{9}$

6 3 5 4 2 1

4 6 7 $\boxed{10}$ 16 12 13 14

1	n-1
n-1	1
n/2	n/2

⟶ Reccurance.

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

Quick sort

90% pivot 10%

Reccurance relat^n.

$T(n) = T\left(\frac{9n}{10}\right) + T\left(\frac{n}{10}\right) + n$ solve by Tree

n

$\frac{9n}{10}$ $\frac{n}{10}$

Matrix multiplication

$$\underset{A}{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}} \underset{B}{\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}} = \underset{C}{\begin{bmatrix} c_{11} & \\ & \end{bmatrix}}$$

$c_{11} = a_{11} \cdot b_{11} + a_{12} \cdot b_{21}$

$c_{12} = a_{11} \cdot b_{12} + a_{12} \cdot b_{22}$

$c_{21} =$

$c_{22} =$

$$\boxed{T(n) = 8T\left(\frac{n}{2}\right) + n^2}$$

$P = (A_{11} + A_{22})(B_{11} + B_{22})$

$Q = (A_{21} + A_{22}) \cdot (B_{11})$

$R = A_{11} \cdot (B_{12} - B_{22})$

$S = A_{22} \cdot (B_{21} - B_{11})$

$T = (A_{11} + A_{12}) \cdot B_{22}$

$U = (A_{21} - A_{11}) \cdot (B_{11} + B_{12})$

$V = (A_{12} - A_{22}) \cdot (B_{21} + B_{22})$

$C_{11} = P + S - T + V$

$C_{12} = R + T$

$C_{21} = Q + S$

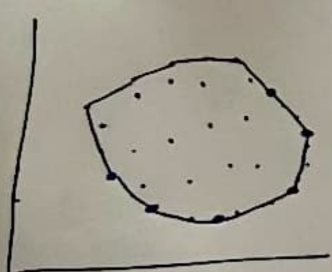
$C_{22} = P + R - Q + U$

Strassen's

$$\boxed{T(n) = 7T\left(\frac{n}{2}\right) + n^2}$$

$O(n^{2.81})$

Convexion (something like this)



Encloses all the points in the polygon.
- will use divide & conqure to construct this polygon.

max min prob in array - use divide & conqure rule - Tut