Linear Search — ~~O(n)~~ O(1) / O(n) / O(n)

Binary Search — O(1) / O(log n) / O(logn)

Insertion Sort — O(n) / O(n^2) / O(n^2)

Binary Sort — O(n) / O(n^2) / O(n^2)

Quick Sort — O(nlogn) / O(n logn) / O(n^2)

Merge Sort — ~~O(nlogn)~~

O(n log n) / O(n logn) / O(nlogn)

Strassen's — O(1) / / O($n^{2.8074}$)

$$T(n) = 7T(n/2) + O(n^2)$$

Max min Sub array —

Convex Hull — O(n logn) / O(nlogn) / O(nlogn)

0/1 Knapsack (DP) — O(n * W)

Fractional (Greedy) = O(n) / / O(nlogn)

Huffman Coding — O(n logn)

Longest common — O(m × n)

Rabin Karp — O(m+n) / O(m+n) / O(mn)

N queen — O(n!)

Master Theorem:

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$ $\qquad\qquad \Theta(n^k \log^p n)$

$a \geq 1 \quad b > 1 \qquad f(n) = \cancel{\Theta(n^k \log^p n)} \; \cancel{\Theta(n^k \log_n p)}$

✓ Find $\log_b a$ and k.

1) If $\log_b a > k$ $\qquad \Theta(n^{\log_b a})$

2) If $\log_b a = k$.

If $P > -1$ $\qquad \Theta(n^k \log^{P+1} n)$

If $P = -1$ $\qquad \Theta(n^k \log\log n)$

If $P < -1$ $\qquad \Theta(n^k)$

3) If $\log_b a < k$

if $P \geq 0$ $\qquad \Theta(n^k \log^p n)$

if $P < 0$ $\qquad \Theta(n^k)$

Eg. -

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

$a = 2, \; b = 2, \; f(n) = n = \Theta(n^k \log^p n)$

$k = 1, \; P = 0,$

$\log_b a = \log_2 2 = 1 = k.$

$\Theta(n^1 \log^1 n) = O(n \log n)$

2. $T(n) = 2T(n/2) + 1$

$a = 2, \; b = 2, \; f(n) = 1 \qquad k = 0, \; P = 0.$

$\log_b a = 1 = k$

$\Rightarrow O(n^1) = O(n)$

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

$a = 4, \ b = 2, \quad f(n) = n \quad k = 1, \ P = 0,$

$\log_2 2^2 \ \ 2 > k$

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

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

$a = 2 \quad b = 2 \qquad f(n) = n \log^{-1} n$

$\log_2 2 = 1 = k \quad k = 1 \quad P = -1$

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

5. $T(n) = 2T(\frac{n}{2}) + \frac{n^2}{\log^2 n}$

$a = 2, \ b = 2 \qquad f(n) = n^2 (\log n)^{-2}$

$k = 2, \quad P = -2$

$\log_2 2 < k$

$O(n^2)$

6. $T(n) = 9T(\frac{n}{9}) + \frac{n}{\log^2 n}$

$a = 9, \ b = 9, \ f(n) = n (\log n)^{-2}$

$\log_b a = 1 = k.$

$O(n)$

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

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

fact (n) $\longrightarrow T(n)$

if n == 1
return 1; $\Big\} \rightarrow (1)$

else return fact (n-1); $\rightarrow T(n-1)$

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

$$T(n) = \begin{cases} 1 & \\ T(n-1)+1 & n>1 \end{cases}$$

Va. $T(n) = T\left(\frac{n}{2}\right) + n^2$
Derive the algorithm using recursion.

b.(i) Solve the recurance relation
$x(n) = x(n-1) + 5$ for $n>1$ and $x(1)=0$
using substitution method.

ii) $x(n) = 3x(n-1)$ for $n>1$ $x(1)=4$

iii) $x(n) = x(n/2) + n$ for $n>1$ $x(1)=1$

iv) $x(n) = x(n-1)+n$ for $n>0$ $x(0)=0$,

$\hookrightarrow$ a.

$a=1,\ b=2$
$k=2,\ P=0$

$O(n^2)$

$n \rightarrow n/2 \rightarrow n/4 \rightarrow n/6 \rightarrow \cdots$

$\downarrow \quad\quad \downarrow \quad\quad \downarrow$

$n^2 \quad \left(\frac{n}{2}\right)^2 \quad \left(\frac{n}{4}\right)^2$

$T(n)\ \ n^2 + \frac{n^2}{2^2} + \frac{n^2}{2^4} + \cdots$

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

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

$= \frac{4}{3} n$

b. i) $x(n) = x(n-1) + 5$

$= x(n-2) + 2 \cdot 5$

$= x(n-3) + 3 \cdot 5$

$\vdots$

$= x(n-k) + k \cdot 5$

Base Condition

$n-k=1 \Rightarrow x(1) = 0$,

$\Rightarrow k = n-1$

$= x(1) + 5(n-1)$

$= 5n - 5$

$= 5n - 5$

$\theta(n)$

ii) $x(n) = 3x(n-1)$

$= 3^2 x(n-2)$

$= 3^k x(n-k)$

$n-k=1$ $= 3^{n-1} \cdot 4$

$\Rightarrow k = n-1$

$= \frac{3^n}{3}$

$O(3^n)$