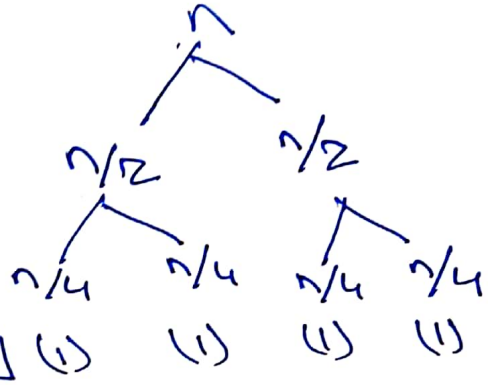
Lab 3

→ Algorithm complexity in terms of 'n'

$O(1) < O(\lg n) < O(n) < O(n \lg n) < O(n^2) < O(2^n) < O(n!)$

- $O(1)$ → Constant Time
- $O(\lg n)$ → Divide & Conquer.
- $O(n)$ → Linear Time
- $O(n^2)$ → Quadratic time

→ Merge Sort:-



or

$n = 2^k$

$\log n = \log 2^k$

$\log n = k \log 2$

$k = \log n$

→ $T(n) = c$ (Divide) + (Merge)

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

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

$= 4T(n/4) + c.n + c.n$

$= 4T(2T(n/8) + c.n/4) + cn + cn$

$= 8T(n/8) + cn + cn + cn$

$= 2^k T(n/2^k) + k.cn$

Until $k = 1$

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

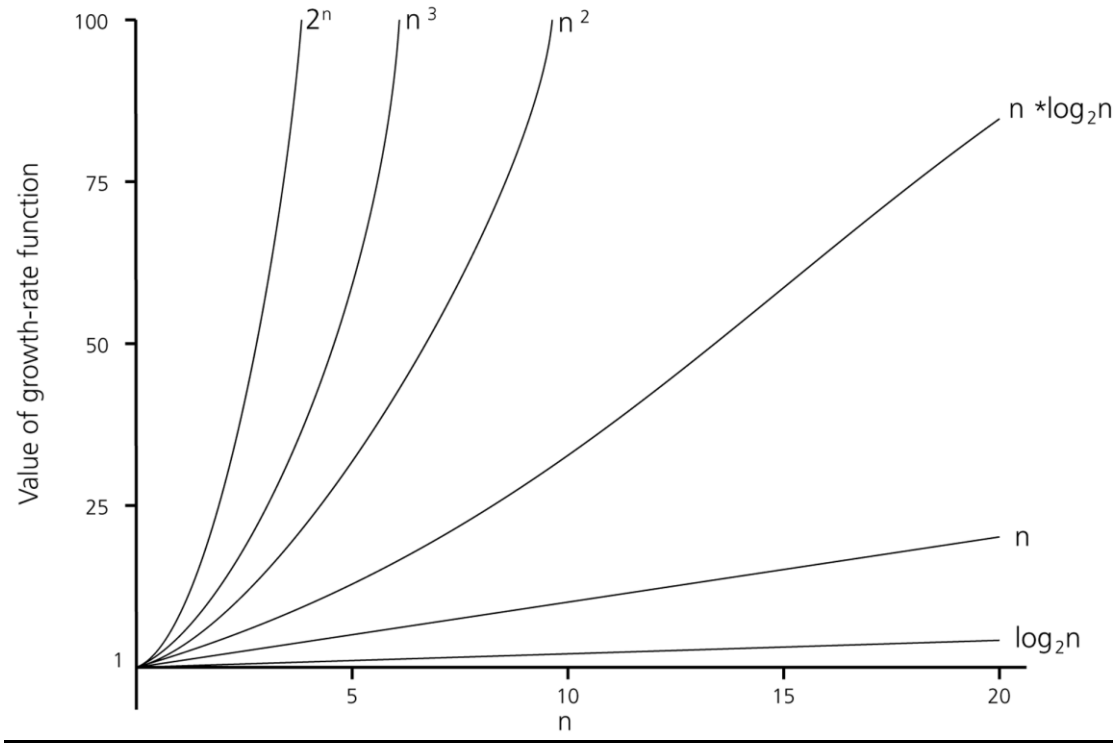
$= n + n \log n$

$= O(n \log n)$.

→ Binary Search.
→ Closest Pair of Points $O(n^2)$, $O(n \lg n)$
→ Strassen's Algo $O(n^3)$ vs $O(n^{2.8})$
→ Cooley-Tukey FFT
→ Karatsuba Algo

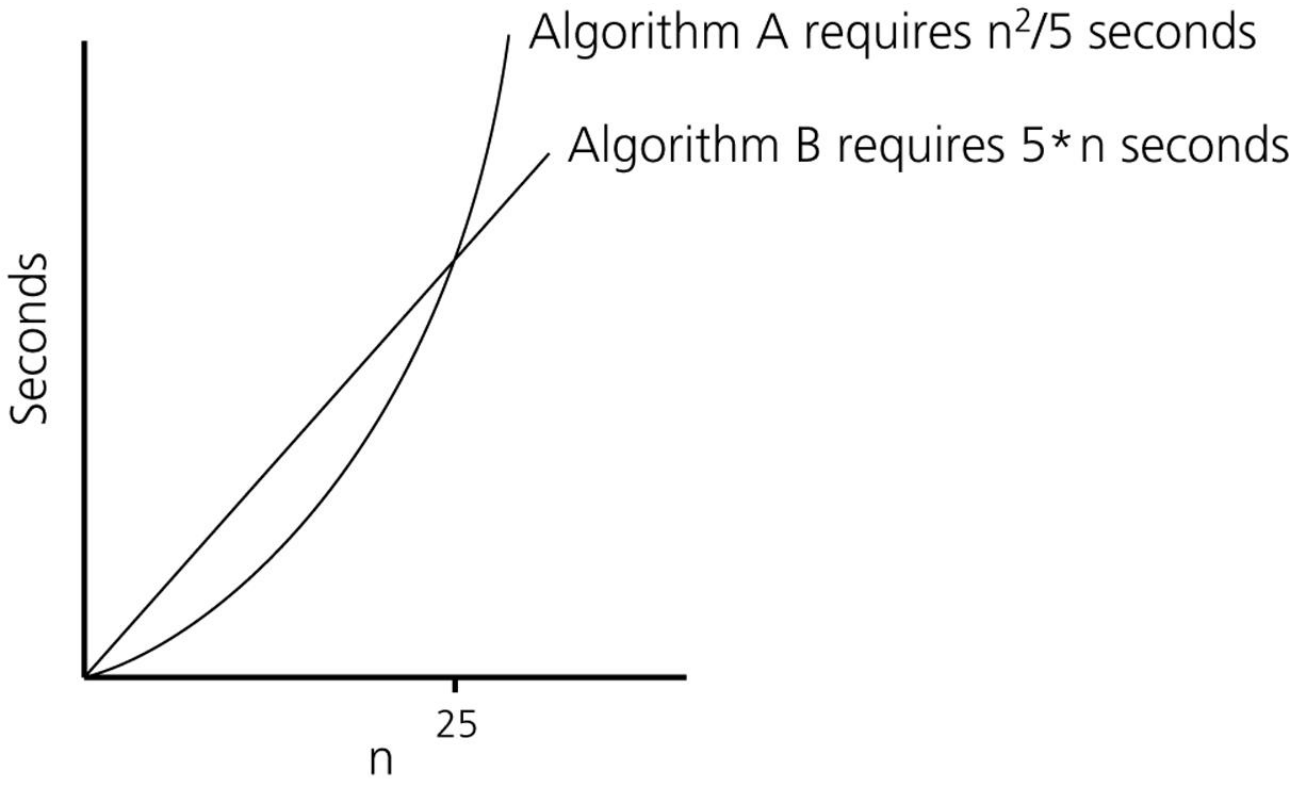
Growth Function

(b)



(a)

Function	n = 10	100	1,000	10,000	100,000	1,000,000
1	1	1	1	1	1	1
$\log_2 n$	3	6	9	13	16	19
n	10	10^2	10^3	10^4	10^5	10^6
$n * \log_2 n$	30	664	9,965	10^5	10^6	10^7
n^2	10^2	10^4	10^6	10^8	10^{10}	10^{12}
n^3	10^3	10^6	10^9	10^{12}	10^{15}	10^{18}
2^n	10^3	10^{30}	10^{301}	$10^{3,010}$	$10^{30,103}$	$10^{301,030}$


Algorithm A requires $n^2/5$ seconds
Algorithm B requires 5*n seconds
Seconds
25
n