

Assignment - 1

① Asymptotic notations are used to tell the complexity of an algorithm when the input is very large.

i) Big O notation (O) - It represents the upper bound of a function. For e.g. if a function $y(n)$ is $O(g(n))$, it means that $f(n)$ grows no faster than $g(n)$.

ii) Omega notation (Ω) - Lower bound of a function growth. Ex - $f(n) = n^2 + 3n$ is $\Omega(n^2)$.

iii) Theta notation (Θ) - Both upper & lower bounds indicating it's asymptotic behaviour.
Ex - $f(n) = n^2 + 3n$ is $\Theta(n^2)$.

② $O(\log n)$ for(*int i=0; i<n; i+=c*)
*c = c * 2*
 for(*int i=0; i<n*)

③ T
 $\left\{ \begin{array}{l} i = c \times 2 \\ y \end{array} \right.$

$$\begin{aligned} T_K &= a \cdot 2^{n-1} \\ n &= 2^{n-1} \end{aligned}$$

Taking log both sides
 $\log_2^n = (K-1) \log_2 2 + \log_2^n = n-1$

$$K = \log_2^n + 1$$

Time complexity $\rightarrow O(\log^n)$

③ $T(n) = \begin{cases} 3T(n-1) & \text{if } n > 0 \\ 3T(n-1) & \text{put } n = n-1 \end{cases}$ otherwise

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

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

so on

$$T(K) = 3^K T(n-K)$$

$$\text{Put } n-K=0 \Rightarrow n=K$$

$$T(n) = 3^n T(0)$$

$$\underline{T(n) = 3^n}$$

④ $T(n) = \begin{cases} 2T(n-1) - 1 & \text{if } n > 0 \\ \text{otherwise} \end{cases}$

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

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

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

& so on

if we expand this —

$$T(n) = 2^k T(n-k) - k$$

We continue this until $n-k=0$, which means $K=n$.

$$T(n) = 2^n T(0) - n$$

Since $T(0)$ is given as 1:

$$\underline{T(n) = 2^n - n}$$

So the solution to the recurrence relation

$$T(n) = 2T(n-1) - 1 \Leftrightarrow \underline{T(n) = 2^n - n}$$

Q) $\text{int } i=1, s=1;$
 $\text{while } (s \leq n)$

α

$i++;$

$s = s + i;$

$\text{Print } ("#");$

3

~~$i \leftarrow n$~~

~~$i \leftarrow 1$~~

$i = 1, 2, 3, 4, 5 \dots$

$s = 1, 3, 6, 10, 15 \dots$

$s = \frac{i(i+1)}{2}$

$\frac{i(i+1)}{2} \leq n$

$i \leftarrow n$
 $i \leftarrow \sqrt{n}$

$i(i+1) \leq 2n$

$i^2 + i - 2n \leq 0$

$i \leq -1 + \frac{\sqrt{1+8n}}{2}$

Complexity = $O(\sqrt{n})$

Q) void function (int n)

$\text{int } i^0, \text{count}=0;$

$\text{for } (i=1; i+i \leq n; i++)$

count++;

3

$i^0 = 1, 2, 3, 4 \dots \sqrt{n}$

$i^0 = 1, 4, 9, 16 \dots n^2$

Complexity = $O(\sqrt{n})$

(7) void function(int n)

 int i, j, k, count = 0

$\frac{n}{2}$ after ($i = \frac{n}{2}; i \leq n; i++$)
 $\log n$ after ($j = 1; j \leq n; j = j + 2$)
 $\log n$ for ($k = 1; k \leq n; k = k + 2$)
 count ++;

$$\frac{n \times \log n}{2} \times \log n$$

$$\text{complexity} = O(n \log^2 n)$$

(8) function (int n)

 if ($n == 1$) return;
 for ($i = 1 + \log n$)
 for ($j = 1 + \log n$)
 point ("*");

y
y

function($n - 3$); $\rightarrow T(n - 3)$

function($n - 3$); $\rightarrow T(n - 3)$

5

$$T(n) = O(n^4) + \underline{T(n - 3)}$$

$$\overbrace{T}^{C = O(n^4)}$$

(9) void

(10)

⑨

void function(int n)
{
 for (i=1 to n)
 for (j=1; j <= n; j = j + i)
 print ("*");

i = 1, 2, 3, 4 —
j = 1, 3, 6, 10 —
 $i = 1 \text{ to } n \text{ times}$
 $i = 2 + \frac{n}{2} + 1 \text{ times}$
 $i = 3 \quad \frac{n}{3} \text{ times}$

$$1 + \frac{n}{2} + \frac{n}{3} + \dots - 1$$

complexity $O(n \log n)$

⑩

n^k & C^n
Assume $n >= 1$ & $C > 1$ $c = ?$
for n^k growth rate $\propto k$
for C^n growth rate $\propto C$
Set $n = 2$
 $n = 2$

$$n^k = 2^k$$

$$C^n = 2^n$$

$$C^2 > 2^k$$

$$C^2 > 2^k$$

Set $n = 3$
 $n = 1$

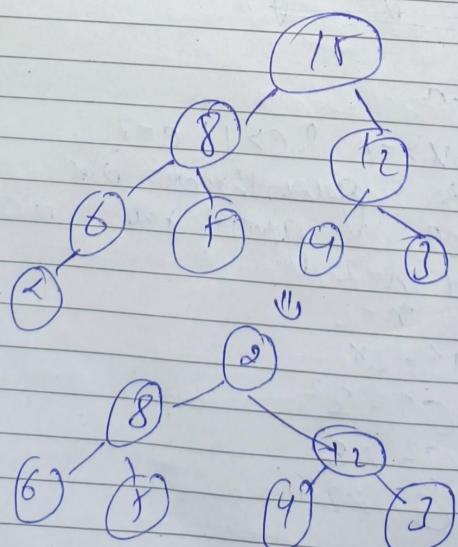
$$3^2 = 9$$

$$2^1 = 2$$

$9 > 2$ C^n grows faster than n^k
 relationship $C^n > n^k$ holds for large
 values of n . (6)

- (11) The time complexity for extracting the min. element using 'extract min()' depends on the underlying data structure. In the case of a binary heap, which is commonly used for priority queues, the complexity is $O(\log n)$.

(12)



$n=3$

$k=1$

$n=2$

$k=2$

$n=1$

$k=3$

$n=0$

$k=4$

$n=-1$

$k=5$

$n=-2$

$k=6$

$n=-3$

$k=7$

$n=-4$

$k=8$

$n=-5$

$k=9$

$n=-6$

$k=10$

$n=-7$

$k=11$

$n=-8$

$k=12$

$n=-9$

$k=13$

$n=-10$

$k=14$

$n=-11$

$k=15$

$n=-12$

$k=16$

$n=-13$

$k=17$

$n=-14$

$k=18$

$n=-15$

$k=19$

$n=-16$

$k=20$

$n=-17$

$k=21$

$n=-18$

$k=22$

$n=-19$

$k=23$

$n=-20$

$k=24$

$n=-21$

$k=25$

$n=-22$

$k=26$

$n=-23$

$k=27$

$n=-24$

$k=28$

$n=-25$

$k=29$

$n=-26$

$k=30$

$n=-27$

$k=31$

$n=-28$

$k=32$

$n=-29$

$k=33$

$n=-30$

$k=34$

$n=-31$

$k=35$

$n=-32$

$k=36$

$n=-33$

$k=37$

$n=-34$

$k=38$

$n=-35$

$k=39$

$n=-36$

$k=40$

$n=-37$

$k=41$

$n=-38$

$k=42$

$n=-39$

$k=43$

$n=-40$

$k=44$

$n=-41$

$k=45$

$n=-42$

$k=46$

$n=-43$

$k=47$

$n=-44$

$k=48$

$n=-45$

$k=49$

$n=-46$

$k=50$

$n=-47$

$k=51$

$n=-48$

$k=52$

$n=-49$

$k=53$

$n=-50$

$k=54$

$n=-51$

$k=55$

$n=-52$

$k=56$

$n=-53$

$k=57$

$n=-54$

$k=58$

$n=-55$

$k=59$

$n=-56$

$k=60$

$n=-57$

$k=61$

$n=-58$

$k=62$

$n=-59$

$k=63$

$n=-60$

$k=64$

$n=-61$

$k=65$

$n=-62$

$k=66$

$n=-63$

$k=67$

$n=-64$

$k=68$

$n=-65$

$k=69$

$n=-66$

$k=70$

$n=-67$

$k=71$

$n=-68$

$k=72$

$n=-69$

$k=73$

$n=-70$

$k=74$

$n=-71$

$k=75$

$n=-72$

$k=76$

$n=-73$

$k=77$

$n=-74$

$k=78$

$n=-75$

$k=79$

$n=-76$

$k=80$

$n=-77$

$k=81$

$n=-78$

$k=82$

$n=-79$

$k=83$

$n=-80$

$k=84$

$n=-81$

$k=85$

$n=-82$

$k=86$

$n=-83$

$k=87$

$n=-84$

$k=88$

$n=-85$

$k=89$

$n=-86$

$k=90$

$n=-87$

$k=91$

$n=-88$

$k=92$

$n=-89$

$k=93$

$n=-90$

$k=94$

$n=-91$

$k=95$

$n=-92$

$k=96$

$n=-93$

$k=97$

$n=-94$

$k=98$

$n=-95$

$k=99$

$n=-96$

$k=100$

$n=-97$

$k=101$

$n=-98$

$k=102$

$n=-99$

$k=103$

$n=-100$

$k=104$

$n=-101$

$k=105$

$n=-102$

$k=106$

$n=-103$

$k=107$

$n=-104$

$k=108$

$n=-105$

$k=109$

$n=-106$

$k=110$

$n=-107$

$k=111$

$n=-108$

$k=112$

$n=-109$

$k=113$

$n=-110$

$k=114$

$n=-111$

$k=115$

$n=-112$

$k=116$

$n=-113$

$k=117$

$n=-114$

$k=118$

$n=-115$

$k=119$

$n=-116$

$k=120$

$n=-117$

$k=121$

$n=-118$

$k=122$

$n=-119$

$k=123$

$n=-120$

$k=124$

$n=-121$

$k=125$

$n=-122$

$k=126$

$n=-123$

$k=127$

$n=-124$

$k=128$

$n=-125$

$k=129$

$n=-126$

$k=130$

$n=-127$

$k=131$

$n=-128$

$k=132$

$n=-129$

$k=133$

$n=-130$

$k=134$

$n=-131$

$k=135$

$n=-132$

$k=136$

$n=-133$

$k=137$

$n=-134$

$k=138$

$n=-135$

$k=139$

$n=-136$

$k=140$

$n=-137$

$k=141$

$n=-138$

$k=142$

$n=-139$

$k=143$

$n=-140$

$k=144$

$n=-141$

$k=145$

$n=-142$

$k=146$

$n=-143$

$k=147$

$n=-144$

$k=148$

$n=-145$

$k=149$

$n=-146$

$k=150$

$n=-147$

$k=151$

$n=-148$

$k=152$

$n=-149$

$k=153$

$n=-150$

$k=154$

$n=-151$

$k=155$

$n=-152$

$k=156$

$n=-153</$