

Name :- Saumya Negi  
Section :- I  
Roll No :- 38

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

TCS-409

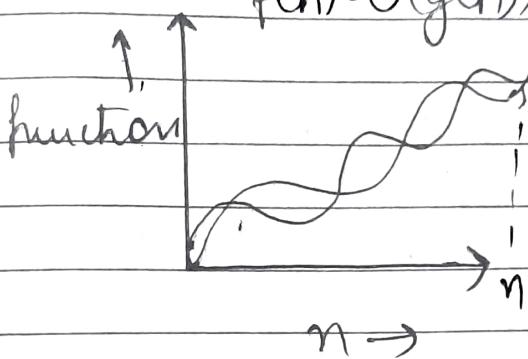
## Tutorial 1

Q1 Asymptotic notations means tending to infinity. They are used to tell the complexity when input is very large

Dif. types of asymptotic notations

i) Big Oh (O)

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



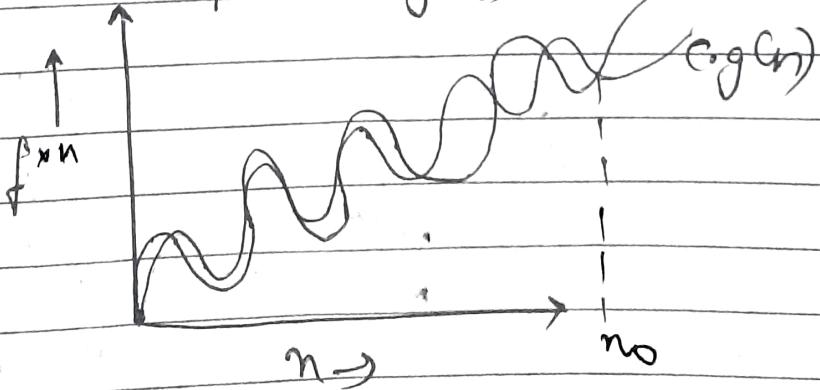
Ex for  $c_1 = 1; i <= n; i++$

{ print ("\*"); —  $o(i)$  }

$$T(n) = O(n)$$

ii) Big Omega ( $\Omega$ )

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



Ex  $f(n) = 2n^2 + 3n + 5$ ,  $g(n) = n^2$

$$0 \leq c \cdot g(n) \leq f(n)$$

$$0 \leq c \cdot n^2 \leq 2n^2 + 3n + 5$$

$$c \leq \frac{2}{n^2} + \frac{3}{n} + \frac{5}{n^2}$$

On putting  $n = \infty$ ,  $\frac{3}{n} \rightarrow 0$ ,  $\frac{5}{n^2} \rightarrow 0$

2)  $c = 2$

$2 \cdot 2n^2 \leq 2n^2 + 3n + 5$

On putting  $n = 2$

$2 \leq 2 + 3 + 5$

$2 \leq 10$  True

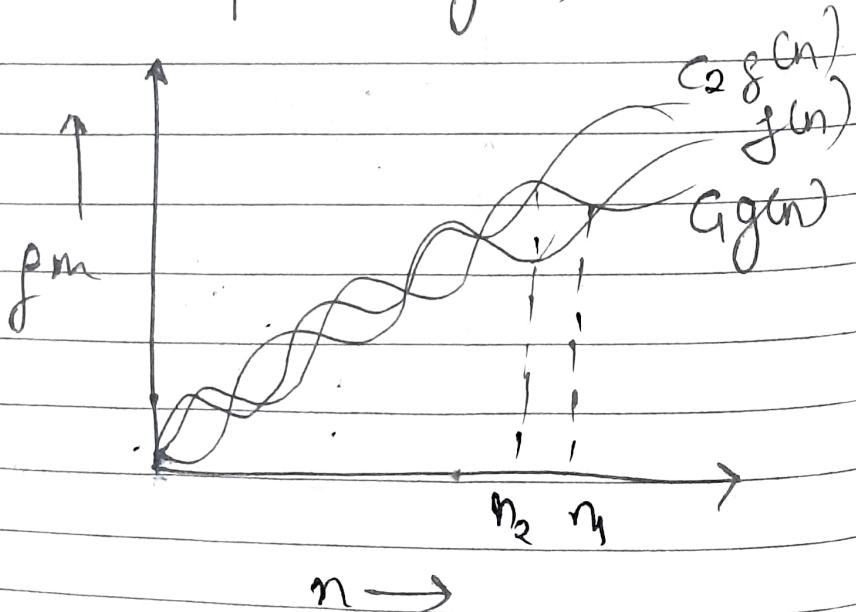
2)  $[c = 2, n = 2]$

$0 \leq 2n^2 \leq 2n^2 + 3n + 5$

$f(n) = \Omega(n^2)$

3) Big Theta ( $\Theta$ )

$f(n) = \Theta(g(n))$



Ex  $f(n) = 10 \log_2 n + 4, g(n) = \log_2 n$

$$f(n) \leq c_2 \cdot g(n)$$

$$\therefore 10 \log_2 n + 4 \leq 10 \log_2 n + \log_2 n$$

$$10 \log_2 n + 4 \leq 11 \log_2 n$$

(Q 21)

$$\therefore 4 \leq 11 \log_2 n - 10 \log_2 n$$

$$4 \leq \log_2 n$$

$$16 \leq n$$

$$\therefore n \geq 16$$

$$n \geq 16$$

(Q 21)

$$f(n) \geq c_1 g(n)$$

$$10 \log_2 n + 4 \geq 4 \log_2 n$$

$$c_1 = 1, n \geq 0$$

$$\therefore n_1 = 2, n_2 = 16$$

$$\therefore \log_2 n \leq 10 \log_2 n + 4 \leq 11 \log_2 n$$

(Q 21, Q 21)

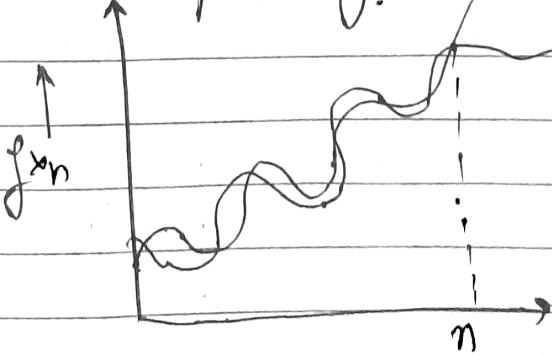
$$\therefore \Theta(\log_2 n)$$

4) small  $O$ (n) -

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

$$O(g(n))$$

$$g(n)$$

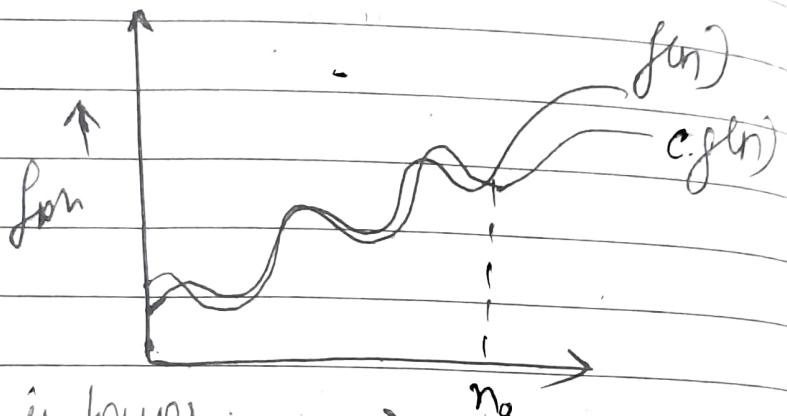


Page

$f(n) = O(g(n))$   
 iff,  $f(n) \leq c \cdot g(n)$   
 &  $n > n_0$ , and  $c$  constant,  $c > 0$

5) small omega ( $\omega$ ) -

$$f(n) = \omega(g(n))$$



$g(n)$  is lower  $n \rightarrow$   
 bound of  $f(n)$

$$f(n) = \omega(g(n))$$

when,

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

&  $n > n_0$

$$\& c > 0$$

Q2 values of  $i = 1, 2, 4, 6, 16, \dots, n$   
 & terms

Is this a GP with  $a_1, r^2$

Now,

$$k^m \text{ term} : - t_k = a^r^{K_1}$$

$$n = 1 \cdot 2^{K_1}$$

$$n = 2^{K_1}$$

taking  $\log_2$  on both sides

$$2) \log_2 n \geq \log_2 2^K$$

$$\log_2 n \geq (K-1) \log_2 2$$

$$\log_2 n \geq K-1 + \log_2 n \quad (\because \log_2 2 = 1)$$

∴ Time complexity  $T(n) \geq \mathcal{O}(K)$

$$\geq \mathcal{O}(K + \log_2 n)$$

$$\geq \mathcal{O}(\log_2 n)$$

$$Q3 \quad T(n) = 3T(n-1) \dots \text{--- } ①$$

put  $n = n-1$  in eq  $\text{--- } ①$

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

$$T(n-1) = 3T(n-2) \text{ --- } ②$$

Put value of  $T(n-1)$  from eq  $\text{--- } ②$  in eq  $\text{--- } ①$

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

$$T(n) = 9T(n-2) \text{ --- } ③$$

Put  $n = n-2$  in eq  $\text{--- } ③$

$$T(n-2) = 3T(n-3) \text{ --- } ④$$

Put value of  $T(n-2)$  in eq  $\text{--- } ③$

$$T(n) = 3[9T(n-3)]$$

$$T(n) = 27T(n-3) \text{ --- } ⑤$$

On Generalising eq  $\text{--- } 5$

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

Put  $n = k = 0$

$$2) T(n) = 3^k T(0)$$

$$= 3^k \quad (\because T(0) = 1)$$

$$\therefore T(n) = \mathcal{O}(3^n)$$

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

Step 1:

$$T(n) = 2T(n-1) - 1 \quad \dots \quad (1)$$

Put  $n = n-1$  in eq<sup>n</sup> (1)

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

$$T(n-1) = 2T(n-2) - 1 \quad \dots \quad (2)$$

Put value of  $T(n-1)$  from (2) in (1)

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

$$T(n) = 4T(n-2) - 2 - 1 \quad \dots \quad (3)$$

Put  $n = n-2$  in eq<sup>n</sup> (1)

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

Put value of  $T(n-2)$  in eq<sup>n</sup> (3)

$$T(n) = 4[2T(n-3) - 1] - 2 - 1$$

$$T(n) = 8T(n-3) - 4 - 2 - 1$$

On Generalising

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

Put  $n-K=0 \Rightarrow n=k$ ,  $T(0)=1$  (Given)

$$\begin{aligned} T(n) &= 2^n T(0) - 2^{n-1} - 2^{n-2} - \dots - 1 \\ &= 2^n - \underbrace{[2^{n-1} + 2^{n-2} + \dots + 1]}_{k \text{ terms}} \end{aligned}$$

$$\Rightarrow Q = 2^{n-1}, R = 1$$

$$\text{Sum of GP} = \frac{2^{n-1} \left[ 1 - \left(\frac{1}{2}\right)^{n-1} \right]}{1 - \frac{1}{2}}$$

$$= 2^n - 2$$

$$\Rightarrow T(n) = 2^n - [2^n - 2] = 2$$

$$= O(2)$$

$$\boxed{T(n) = O(1)}$$

Ques-5 What should be the time complexity of -

int i=1, s=L;

while (s<=n) {

    i++; s=s+i;

    printf("#");

}

Sol:

i	s
1	1
2	3
3	6
4	10
5	15
6	21
7	28
8	36
9	45
10	55
11	66
12	78
13	91
14	104
15	118
16	132
17	147
18	162
19	178
20	194
21	210
22	227
23	244
24	261
25	278
26	295
27	312
28	330
29	348
30	366
31	384
32	402
33	420
34	438
35	456
36	474
37	492
38	510
39	528
40	546
41	564
42	582
43	600
44	618
45	636
46	654
47	672
48	690
49	708
50	726
51	744
52	762
53	780
54	798
55	816
56	834
57	852
58	870
59	888
60	906
61	924
62	942
63	960
64	978
65	996
66	1014
67	1032
68	1050
69	1068
70	1086
71	1104
72	1122
73	1140
74	1158
75	1176
76	1194
77	1212
78	1230
79	1248
80	1266
81	1284
82	1302
83	1320
84	1338
85	1356
86	1374
87	1392
88	1410
89	1428
90	1446
91	1464
92	1482
93	1500
94	1518
95	1536
96	1554
97	1572
98	1590
99	1608
100	1626
101	1644
102	1662
103	1680
104	1698
105	1716
106	1734
107	1752
108	1770
109	1788
110	1806
111	1824
112	1842
113	1860
114	1878
115	1896
116	1914
117	1932
118	1950
119	1968
120	1986
121	2004
122	2022
123	2040
124	2058
125	2076
126	2094
127	2112
128	2130
129	2148
130	2166
131	2184
132	2202
133	2220
134	2238
135	2256
136	2274
137	2292
138	2310
139	2328
140	2346
141	2364
142	2382
143	2400
144	2418
145	2436
146	2454
147	2472
148	2490
149	2508
150	2526
151	2544
152	2562
153	2580
154	2598
155	2616
156	2634
157	2652
158	2670
159	2688
160	2706
161	2724
162	2742
163	2760
164	2778
165	2796
166	2814
167	2832
168	2850
169	2868
170	2886
171	2904
172	2922
173	2940
174	2958
175	2976
176	2994
177	3012
178	3030
179	3048
180	3066
181	3084
182	3102
183	3120
184	3138
185	3156
186	3174
187	3192
188	3210
189	3228
190	3246
191	3264
192	3282
193	3300
194	3318
195	3336
196	3354
197	3372
198	3390
199	3408
200	3426
201	3444
202	3462
203	3480
204	3498
205	3516
206	3534
207	3552
208	3570
209	3588
210	3606
211	3624
212	3642
213	3660
214	3678
215	3696
216	3714
217	3732
218	3750
219	3768
220	3786
221	3804
222	3822
223	3840
224	3858
225	3876
226	3894
227	3912
228	3930
229	3948
230	3966
231	3984
232	4002
233	4020
234	4038
235	4056
236	4074
237	4092
238	4110
239	4128
240	4146
241	4164
242	4182
243	4200
244	4218
245	4236
246	4254
247	4272
248	4290
249	4308
250	4326
251	4344
252	4362
253	4380
254	4398
255	4416
256	4434
257	4452
258	4470
259	4488
260	4506
261	4524
262	4542
263	4560
264	4578
265	4596
266	4614
267	4632
268	4650
269	4668
270	4686
271	4704
272	4722
273	4740
274	4758
275	4776
276	4794
277	4812
278	4830
279	4848
280	4866
281	4884
282	4902
283	4920
284	4938
285	4956
286	4974
287	4992
288	5010
289	5028
290	5046
291	5064
292	5082
293	5100
294	5118
295	5136
296	5154
297	5172
298	5190
299	5208
300	5226
301	5244
302	5262
303	5280
304	5298
305	5316
306	5334
307	5352
308	5370
309	5388
310	5406
311	5424
312	5442
313	5460
314	5478
315	5496
316	5514
317	5532
318	5550
319	5568
320	5586
321	5604
322	5622
323	5640
324	5658
325	5676
326	5694
327	5712
328	5730
329	5748
330	5766
331	5784
332	5802
333	5820
334	5838
335	5856
336	5874
337	5892
338	5910
339	5928
340	5946
341	5964
342	5982
343	6000
344	6018
345	6036
346	6054
347	6072
348	6090
349	6108
350	6126
351	6144
352	6162
353	6180
354	6198
355	6216
356	6234
357	6252
358	6270
359	6288
360	6306
361	6324
362	6342
363	6360
364	6378
365	6396
366	6414
367	6432
368	6450
369	6468
370	6486
371	6504
372	6522
373	6540
374	6558
375	6576
376	6594
377	6612
378	6630
379	6648
380	6666
381	6684
382	6702
383	6720
384	6738
385	6756
386	6774
387	6792
388	6810
389	6828
390	6846
391	6864
392	6882
393	6900
394	6918
395	6936
396	6954
397	6972
398	6990
399	7008
400	7026
401	7044
402	7062
403	7080
404	7098
405	7116
406	7134
407	7152
408	7170
409	7188
410	7206
411	7224
412	7242
413	7260
414	7278
415	7296
416	7314
417	7332
418	7350
419	7368
420	7386
421	7404
422	7422
423	7440
424	7458
425	7476
426	7494
427	7512
428	7530
429	7548
430	7566
431	7584
432	7602
433	7620
434	7638
435	7656
436	7674
437	7692
438	7710
439	7728
440	7746
441	7764
442	7782
443	7800
444	7818
445	7836
446	7854
447	7872
448	7890
449	7908
450	7926
451	7944
452	7962
453	7980
454	7998
455	8016
456	8034
457	8052
458	8070
459	8088
460	8106
461	8124
462	8142
463	8160
464	8178
465	8196
466	8214
467	8232
468	8250
469	8268
470	8286
471	8304
472	8322
473	8340
474	8358
475	8376
476	8394
477	8412
478	8430
479	8448
480	8466
481	8484
482	8502
483	8520
484	8538
485	8556
486	8574
487	8592
488	8610
489	8628
490	8646
491	8664
492	8682
493	8700
494	8718
495	8736
496	8754
497	8772
498	8790
499	8808
500	8826

$$\begin{aligned} k &= t_k - t_{k-1} - \dots \quad \text{--- (1)} \\ \Rightarrow k &= m - t_{k-1} \end{aligned}$$

loop runs  $k$ -times

$$\text{Time Complexity} = O(1+1+1+n-t_{n-1})$$

but  $t_{n-1} = c$  (constant)

$$\therefore \text{Time Complexity} = O(3+n-1) \\ = O(n)$$

Ques-7: Time Complexity of

void function (int  $n$ )

{ int  $i, j, k$ ; count = 0;

for ( $i = n/2$ ;  $i \leq n$ ;  $i++$ )

for ( $j = 1$ ;  $j \leq n$ ;  $j = j*2$ )

for ( $k = 1$ ,  $k \leq n$ ;  $k = k*2$ )

count++

3.

Sol

$$i \rightarrow n/2, \frac{n+2}{2}, \frac{n+4}{2}, \frac{n+6}{2}, \dots \text{upto } n$$

$$= \frac{n+0 \times 2}{2} + \frac{n+1 \times 2}{2} + \frac{n+2 \times 2}{2}, \dots \text{upto } n$$

General form:

$$t_k = \frac{n + k \times 2}{2}$$

$$\text{total terms} = k+1$$

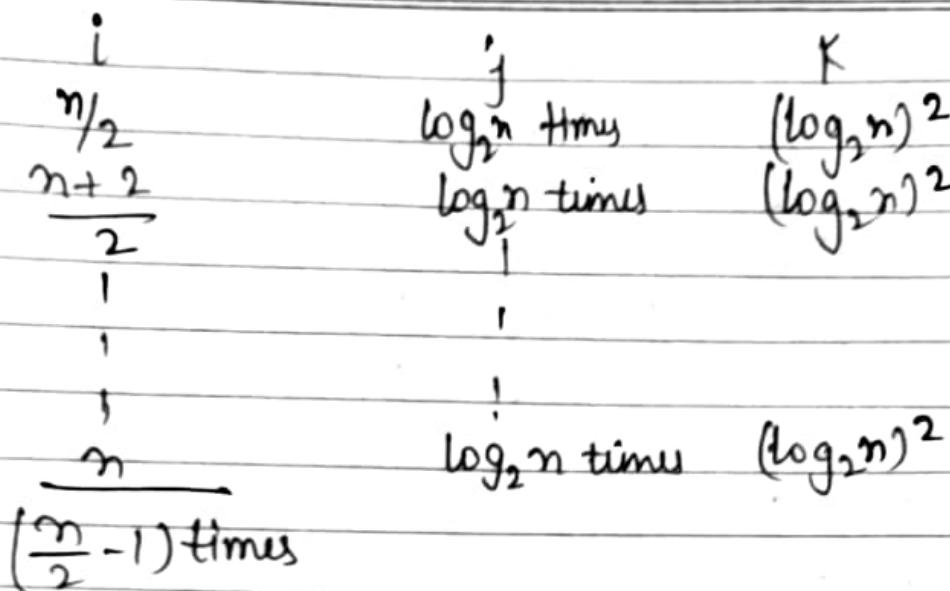
$$t_{k+1} = n$$

$$\Rightarrow \frac{n + (k+1) \times 2}{2} = n$$

$$n + 2k + 2 = 2n$$

$$2k = n - 2$$

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



$$\Rightarrow \left(\frac{n}{2} - 1\right)(\log_2 n)^2$$

$$= O\left(\frac{n}{2} \log_2^2 n - \log_2 n\right)$$

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

Ques-6 Time complexity of -

void function(int n) {  
 O(1) — int i, count = 0;  $\underbrace{O(1)}$   
 for (i = 1; i  $\leq$  n; i++)  
 count++; — O(1)}

Q.

$$i * i$$

$$1^2$$

$$2^2$$

$$3^2$$

$$4^2$$

$$i * i = \underbrace{1^2, 2^2, 3^2, 4^2, \dots, n}_{k \text{ terms}}$$

$$\Rightarrow k^{\text{th}} \text{ term}; t_k = k^2$$

$$k^2 = n$$

$$k = n^{1/2}$$

$$\begin{aligned}
 \text{Time complexity} &= O(1+1+1+n^{1/2}) \\
 &= O(n^{1/2}) \\
 &= O(\sqrt{n})
 \end{aligned}$$

Ques-8: Time complexity of function( $\text{int } n$ ) {

    if ( $n == 1$ ) return; —  $O(1)$

    for ( $i = 1$  to  $n$ ) { —  $O(n)$

        for ( $j = 1$  to  $n$ ) { —  $O(n)$

            printf("\*"); —  $O(1)$

}

    } function( $n-3$ );

3.

Sol: for function call

$\underbrace{n, n-3, n-6, n-9, \dots, 1}_{K \text{ terms}}$

K terms

AP with  $d = -3$ ,  $a = n$

$$a_n = a + (n-1)d$$

$$1 = n + (K-1)(-3)$$

$$\frac{1-n}{(-3)} = K-1$$

$$K-1 = \frac{n-1}{3}$$

$$K = \frac{n-1+3}{3}$$

$$K = \frac{n+2}{3}$$

Hence 1 function have a recursive call  $\frac{n+2}{3}$  times

$$\Rightarrow \text{Time Complexity} = \left(\frac{n+2}{3}\right)(n)(n)$$
$$= O(n^3)$$

Ques-9 Time Complexity of

void function (int n) {

    for (i=1 to n) {

        for (j=1; j<=n; j=j+i)

            printf ("\*");

    }

}

for  $i=1 \rightarrow j=1, 2, 3, 4, \dots, n = n$

for  $i=2 \rightarrow j=1, 3, 5, 7, \dots, n = n/2$

for  $i=3 \rightarrow j=1, 4, 7, \dots, n = n/3$

for  $i=n \Rightarrow j=1, \dots, n = 1$

$$\Rightarrow \sum_{j=n}^1 n + n/2 + n/3 + n/4 + \dots + 1$$

$$\sum_{j=n}^1 n \left[ 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} \right]$$

$$\sum_{j=n}^1 n \log n$$

$$T(n) = [n \log n]$$

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

Sol:

As given  $n^K$  and  $c^n$   
relation b/w  $n^K$  and  $c^n$  is  $\boxed{n^K = O(c^n)}$

as  $n^K \leq ac^n \quad \forall n \geq n_0$  for a constant  $a > 0$

for  $n_0 = 1$

$c = 2$

$\Rightarrow \boxed{1^K \leq 2^1}$

$\therefore \boxed{n_0 = 1, \text{ and } c = 2}$