

Name - Prabhdeep Singh
 Section - G
 Roll no - 04

classmate
 Date _____
 Page _____

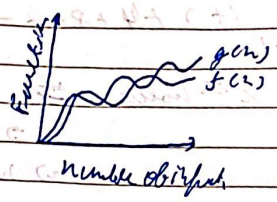
design and analysis of algorithm

Tutorial 1

Sol.

Asymptotic notation - help you find the complexity of an algorithm when input is very large

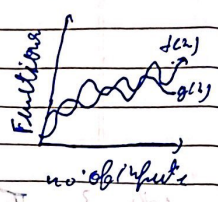
1) Big O



$f(n) = O(g(n))$
 if $n \geq n_0$
 $\forall n \geq n_0$
 for some constant $C > 0$
 $\Rightarrow g(n)$ is tight upper bound of $f(n)$

2) Big omega (Ω)

$f(n) = \Omega(g(n))$
 $g(n)$ is tight lower bound of $f(n)$
 $f(n) = \Omega(g(n))$
 if $f(n) \geq g(n)$
 $\forall n \geq n_0$ for some constant $C > 0$.



3) Theta (Θ)

$f(n) = \Theta(g(n))$
 $g(n)$ is both tight upper and lower bound of $f(n)$
 $f(n) = \Theta(g(n))$
 if
 $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
 $\forall n \geq (\max(n_1, n_2))$ for some constant $c_1 > 0$ and $c_2 > 0$

4) Small O (o)

$f(n) = o(g(n))$
 $g(n)$ is upper bound of $f(n)$
 $f(n) = o(g(n))$

Final

Soln 2

for $(i \geq 1, \text{ to } n) \{ i = i * 2 \}$ for $(i = 1 \text{ to } n) \{ i = 1, 2, 4, 8, \dots, n \}$
 $\{ i = i * 2; \}$ (100%)

$$\Rightarrow \sum_{i=1}^n 1 + 2 + 4 + 8 + \dots + n$$

for K^{th} value $\Rightarrow T_K = 2^{K-1}$

$$\Rightarrow 1 \times 2^{K-1}$$

$$\Rightarrow n = 2^{K-1}$$

$$2n = 2^K$$

$$\text{so } 2n = K \cdot \log 2$$

$$\log 2 + \log n = K \log 2$$

$$\log n = K$$

$$O(K) = O(1 + \log n)$$

$$= O(\log n)$$

Soln 3 $T(n) = \{ 3T(n-1) \text{ if } n > 0, \text{ otherwise } 1 \}$

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

$$\text{Put } n = n-1$$

$$T(n-1) = 3T(n-2) \text{ --- (2)}$$

from (1) and (2)

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

$$T(n) = 9T(n-2) \text{ --- (3)}$$

putting $n = n-2$ in (3)

$$T(n) = 3(T(n-3)) \text{ --- (4)}$$

$$T(n) = 27T(n-3)$$

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

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

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

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

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

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

$$\left. \begin{aligned} T(n) &= 2T(n-1) \\ T(n-1) &= 2T(n-2) \\ T(n-2) &= 2T(n-3) \\ T(0) &= 2T(0) \end{aligned} \right\} \text{h levels}$$

$$T(0) = 1$$

Substituting above values we get

$$T(n) = 2^n \times T(0)$$

$$T(n) = 2^n \times 1$$

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

Soln 5

int i = 1, s = 1; cout

while (s <= n)

{

i++;

s = s + i; i = i + 1;

cout << " ";

}

i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000

$$S = 1 + 3 + 6 + 10 + 15 + \dots$$

$$\text{sum of } S = 1 + 3 + 6 + 10 + \dots + n - 1 \quad \text{--- (1)}$$

$$\text{also } S = 1 + 3 + 6 + 10 + \dots + n \quad \text{--- (2)}$$

from (1) - (2),

$$0 = 1 + 2 + 3 + 4 + \dots + n - n$$

$$TK = 1 + 2 + 3 + 4 + \dots + n$$

$$TK = \frac{1}{2} K(K+1)$$

For K iteration, $1 + 2 + \dots + K \leq n$

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

$$K \approx \sqrt{2n}$$

$$K \approx O(\sqrt{n})$$

sol 46

void function (int n) {

int i; count = 0;

for (i = 1; i <= n; i++)

count++;

}

or i.e. $C = n$ $i <= \sqrt{n}$ $i = 1, 2, 3, 4, \dots, \sqrt{n}$ $\sum_{i=1}^n 1 + 2 + 3 + 4 + \dots + \sqrt{n}$

$$\Rightarrow T(n) = \sqrt{n} \times (\sqrt{n} + 1)$$

$$T(n) = n \times \sqrt{n}$$

$$T(n) = O(n^{3/2})$$

Ans 7

void function (int n)

{

int i, j, k, count = 0;

for (i = n/2; i <= n; i++)

for (j = 1; j <= i; j = j * 2)

for (k = 1; k <= i; k = k * 2)

count++;

}

for $k = k * 2$ $k = 1, 2, 4, 8, \dots, n$ $\Rightarrow GP \Rightarrow a = 1, r = 2$

$$a \frac{(r^n - 1)}{r - 1}$$

$$= 1 \frac{(2^k - 1)}{2 - 1}$$

$$n = 2^k \Rightarrow \log n = k$$

i	j	k
1	$\log n$	$\log n * \log n$
2	$\log n$	$\log n * \log n$
...
n	$\log n$	$\log n * \log n$

$$\Rightarrow O(n * \log n + \log n)$$

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

7. fibonacci (int n)

```

{
    if (n == 1) return; // O(1)
    for (i = 1 to n) { // i = 1, 2, 3, 4 ... n → O(n)
        for (j = 1 to n) { // j = 1, 2, 3, 4 ... n → O(n)
            print ("x");
        }
    }
}

```

function (n-3) : $T(n/3)$

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

$$a=1, b=3, f(n) = n^2$$

$$c = \log_3 3 = 0$$

$$\Rightarrow n \geq 1 \Rightarrow O(n) \geq 2, 2)$$

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

9. void function (int n) {
 for (i = 1 to n) { 110 (n)
 for (j = 1; j ≤ n; j = j + 1)
 Print + ("*") ; 110 (1)
 }
 }

for i = 1 ⇒ j = 1, 2, 3, 4, ... n
 for i = 2 ⇒ j = 1, 2, 3, ... n/2
 for i = 3 ⇒ j = 1, 2, 3, ... n/3
 for i = 4 ⇒ j = 1, 2, 3, ... n/4

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

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

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

Sol: 10

Relation B/W n^k and n^c is

$$n^k = O(n^c)$$

(as $n^k \leq a n^c$ for $n \geq n_0$ and some constant $a > 0$)

for $n_0 = 1$

$$1 \leq 2$$

$$\Rightarrow 1^k \leq 2^c$$

$n_0 = 1$ and $1 \leq 2$