Original **take U forward**’s Video : <https://www.youtube.com/watch?v=WNtzUR_MwUQ>  
  
Original Roadmap (**SDE-PROBLEMS** Sheet) : [https://bit.ly/takeUforward](https://docs.google.com/document/d/1SM92efk8oDl8nyVw8NHPnbGexTS9W-1gmTEYfEurLWQ/edit)

Dream FAANG Google Classroom  
(Code : **mfphb56**) (Invite [Link](https://classroom.google.com/c/MTE3NTIzNzQzNzI1?cjc=mfphb56))

Dream FAANG Roadmap

(***Mobile Version*** *:* [*Mind-Map of Dream FAANG Roadmap*](https://whimsical.com/dream-faang-roadmap-7bz8sqzrB2bBVMhXefGZKt@2Ux7TurymN6FUTWi5NrZ))

|  |  |
| --- | --- |
| **Last Updated On :** | **11:54 pm IST, 04/April/2021** |

Author : [Kunal Chand](http://www.linkedin.com/in/kunal-chand-ab718b175/)

|  |
| --- |
| **INDEX** [INDEX](#_8adfzrfts3jl)  [Week 1 - Arrays](#_1kmfif8zk9ow)  [Editorial](#_blyo2kx0hu4x)  [(1) Find the duplicate in an array of N+1 integers.](#_wvo7na8mkve1)  [(2) Sort an array of 0’s 1’s 2’s without using extra space or sorting algo (Sort Colors)](#_oaas6rnxpv88)  [(3) Repeat and Missing Number](#_r1fngucq4z19)  [(4) Merge two sorted Arrays without extra space](#_yx51d9vrdve5)  [(5) Kadane’s Algorithm](#_c8ss94qi457w)  [(6) Merge Overlapping Subintervals (Merge Intervals)](#_6qu52c9zcgv8)  [Practice (Week 1)](#_8t7mr0ku9mtr)  [Q1. Find All Numbers Disappeared in an Array](#_t6r2gznpijyg)  [Q2. Smallest Missing Integer](#_t4af6lqg9yvc)  [Q3. Flipping Game](#_kef2lektxhtr)  [Q4. Insert Interval](#_g5h9rkz55dby)  [Q5. Product of Array Except Self](#_vcicug2pkvz1)  [Q6. Duplicate Zeros](#_pclar9w4qdku)  [Q7. Wiggle Sort II](#_bhfgscpbrt5q)  [Q8. Recursive Insertion Sort](#_xiy7x0qqb3xm)  [Week 2 - Multi Dimensional Arrays](#_zhz839fu9l97)  [Editorial](#_1grxxcvy3oou)  [(1) Set Matrix Zeros](#_gh1x25zfqtzh)  [(2) Pascal Triangle](#_crnjclvjm4oh)  [(3) Next Permutation](#_1iba5pqa0rzd)  [(4) Inversion of Array (Using Merge Sort)](#_cepfdfubkv8y)  [(5) Stock Buy and Sell](#_yn50gr15is5j)  [(6) Rotate Matrix](#_suh3bs2xiwof)  [Practice (Week 2)](#_4u60nxe93cw0)  [Q1. Game of Life](#_fsomk293vhei)  [Q2. Best Time to Buy and Sell Stock II](#_u8tz8pen31sj)  [Q3. Spiral Matrix Traversal](#_t9f1resc13h2)  [Q4. Max Sum of Rectangle No Larger Than K](#_3bnro3xhctkd)  [Q5. NEKO's Maze Game](#_clzi2afr5ozg)  [Week 3 - Math](#_o2djixmg6xmc)  [Editorial](#_3hqhtwxf70h2)  [(1) Search in a 2D matrix](#_ku33zhj15ubt)  [(2) Pow(X,n)](#_vg703n9ypow)  [(3) Majority Element (>N/2 times) [SEARCH FOR MORE Explanations]](#_xzvzk2apei9u)  [(4) Majority Element (>N/3 times) [SEARCH FOR MORE Explanations]](#_9jht3hdpbyea)  [(5) Grid Unique Paths](#_d6ywjrzahjnk)  [(6) Reverse Pairs [SEARCH FOR MORE Explanations]](#_d4ss1x8xwgia)  [Practice (Week 3)](#_qnqgqu6no7an)  [Q1. Excel Sheet Column Number & Title Conversion](#_yju6jw3k9tl1)  [Q2. Factorial Trailing Zeroes](#_pnjb2ougb3o6)  [Q3. Find GCD in Log N](#_exqw8k89gfbx)  [Q4. Sqrt(x) / Valid Perfect Square](#_vm5wxmt25csn)  [Q5. Roman & Integer Conversion](#_sbn5y92acfrm)  [Q6. Maximum Swap](#_7txehzkjc27u)  [Q7. Tower of Hanoi](#_jqp6zqi83y60)  [Q8. Count Primes (Sieve of Eratosthenes)](#_86pdbfnn9gkj)  [Week 4 - Hashing](#_nl15yxdsu448)  [Editorial](#_pg6rmb6lthhx)  [(1) 2 Sum problem](#_9mvkof39jwc5)  [(2) 4 Sum problem](#_5tzt7q2q3pli)  [(3) Longest Consecutive Sequence](#_p5s5qr68c1ef)  [(4) Longest Subarray with K sum](#_je1e3sl34qk8)  [(5) Count number of subarrays with given XOR](#_e3r6d6ulzh1i)  [(6) Longest substring without repeat](#_vx508buq8ujt)  [Practice (Week 4)](#_6sbmry4r1hg1)  [Q1. Valid Sudoku](#_jfjmwjrae9za)  [Q2. Contiguous Array [WRITE A CLEAN EXPLANATION]](#_a83muaxvr482)  [Q3. Insert Delete GetRandom in O(1) Time](#_vin9iz7h27p)  [Q4. Design HashSet/HashMap](#_d4tsbclgut6x)  [Q5. Eugene and an array [WRITE A CLEAN EXPLANATION]](#_80sx5hfec15c)  [Q6. Longest subarray with sum divisible/not divisible by K](#_r4o79j4snfxp)  [Q7. Subarray Sums Divisible by K [WRITE A CLEAN EXPLANATION]](#_igl9frv91l6u)  [Q8. Flip Columns For Maximum Number of Equal Rows [WRITE A CLEAN EXPLANATION]](#_a1rep0t8jjb2)  [Week 5 - Linked List](#_vixhtvbj8vc8)  [Editorial](#_vzydu7uhl24v)  [(1) Reverse a LinkedList](#_es49u9mz85q6)  [(2) Find middle of LinkedList](#_qx633ls0w28w)  [(3) Merge two sorted Linked List](#_7dmus2d85ye9)  [(4) Remove N-th node from back of LinkedList](#_z0z0t7ya0zpc)  [(5) Delete a given Node when a node is given. (0(1) Explanation)](#_idpfq0pl4kx1)  [(6) Add two numbers as LinkedList](#_o4o2fgghffeq)  [Practice (Week 5)](#_ytyultpewsjy)  [Q1. Reverse Linked List II](#_lzvg3wxmqhlr)  [Q2. Partition List](#_l4rewg4z0p2t)  [Q3. Merge k Sorted Lists](#_k3tdyq9epjfr)  [Q4. Add Two Numbers II](#_p24t4bwewu9)  [Q5. Odd Even Linked List](#_7e2tss8elfjk)  [Q6. Remove Linked List Elements](#_c0iv7nj2mfsm)  [Q7. Remove Duplicates from Sorted List](#_obrrgx4335c2)  [Week 6 - Advanced Linked List](#_14ba96knze5d)  [Editorial](#_vzfxieaw2h80)  [(1) Find intersection point of Y LinkedList](#_4adxsssntd8s)  [(2) Detect a cycle in Linked List](#_5xq3yb1iz4f5)  [(3) Reverse a LinkedList in groups of size k. (Reverse Nodes in k-Group)](#_vi90l1ts7d58)  [(4) Check if a LinkedList is palindrome or not.](#_sg50s8heu31u)  [(5) Find the starting point of the Loop of LinkedList](#_thmx5ghn3tgr)  [(6) Flattening of a LinkedList](#_p0xsdftw2jdz)  [(7) Rotate a LinkedList](#_u7h34yflojwm)  [Practice (Week 6)](#_uu1hc1qusa46)  [Q1. Linked List in Zig-Zag fashion](#_jgs8r39iifdg)  [Q2. Reorder List](#_wofjnjuayr4m)  [Q3. Remove Zero Sum Consecutive Nodes from Linked List](#_l7v9qau5p08a)  [Q4. Sort List](#_n2bnyg5h72t)  [Q5. Insertion Sort List](#_4haxy15forcg)  [Q6. Remove Duplicates from Sorted List II](#_fpwmscqf3q11)  [Q7. QuickSort on Doubly Linked List [SEARCH MORE EXPLANATIONS]](#_scf5aynv7otv)  [Week 7 - Two Pointer](#_tpmv76v9v62i)  [Editorial](#_idhzrkw34d2d)  [(1) Clone a Linked List with random and next pointer.](#_gfdztchsfvek)  [(2) 3 sum](#_mtedh9xanqdh)  [(3) Trapping rainwater](#_munrqicghr4d)  [(4) Remove Duplicate from Sorted array](#_9jayjenyfj9b)  [(5) Max continuous number of 1’s](#_lf1dzv5lvw2g)  [Practice (Week 7)](#_69rv1o6sn1lv)  [Q1. Subarray Product Less Than K](#_naenz6tqkrf5)  [Q2. Subarrays with K Different Integers](#_j68d2johzx15)  [Q3. Subarrays with Sum K [SEARCH FOR MORE EXPLANATIONS]](#_ae3wz39yn4h6)  [Q4. Smallest Subarray with Sum at least K [SEARCH FOR MORE EXPLANATIONS]](#_a50s41zqxps)  [Q5. Remove Duplicates from Sorted Array II](#_8968g4nd2dvr)  [Q6. Move Zeroes](#_8a5y6060lqfe)  [Q7. 3Sum Closest](#_gb27mgrocinc)  [Q8. Container With Most Water](#_77rlkd4i39zy)  [Q9. Minimum Window Substring [SEARCH FOR MORE EXPLANATION]](#_liy1w43je03)  [Week 8 - Greedy](#_2nqyag9q9cly)  [Editorial](#_vkxr4y7tj74m)  [(1) N meeting in one room](#_evav73t4dw1w)  [(2) Minimum number of platforms required for a railway](#_vmt6diudlnim)  [(3) Job sequencing Problem (Maximum Profit in Job Scheduling)](#_76p1y1q7hgu3)  [(4) Fractional Knapsack Problem](#_xwrlb6i3iaow)  [(5) Greedy algorithm to find minimum number of coins](#_trh5y427b19u)  [(6) Activity Selection [SEARCH FOR MORE EXPLANATION]](#_ck923n2jk0wa)  [Practice (Week 8)](#_arivqvkq8v3)  [Q1. Non-overlapping Intervals [SEARCH FOR MORE EXPLANATION]](#_qenlkgq7e2ii)  [Q2. Remove Covered Intervals [SEARCH FOR MORE EXPLANATION]](#_5dpfy4sqmtdh)  [Q3. Chef and Bitwise Product [SEARCH FOR MORE EXPLANATION]](#_apqhxbaj5syv)  [Q4. Gas Station](#_8xsv2ueb1t29)  [Week 9 - Recursion](#_584qj8dz6wtp)  [Editorial](#_xt0k1pi6zifn)  [(1) Combination sum-1](#_s932wk72w6f2)  [(2) Combination sum-2](#_meev12l7943x)  [(3) Palindrome Partioning](#_lfklf01k4zdc)  [(4) Subset Sum-1 (Subset Sums) [Search For More Explanation]](#_hctv6r92fhv6)  [(5) Subset Sum-2 [Search For More Explanation]](#_1u8ike3fislv)  [(6) K-th permutation Sequence (Permutation Sequence)](#_s26xw0n96m44)  [Week 10 - Backtracking](#_k3vuuolimbu6)  [Editorial](#_qiywdrj2snw1)  [(1) N queens Problem](#_f98qb0nl2w6u)  [(2) Sudoko](#_ugo72f33btiz)  [(3) M coloring Problem (Graph prob)](#_2sh81ulsq477)  [(4) Rat in a Maze](#_7pger8aughn)  [(5) Print all Permutations of a string/array](#_7itzh3b7purl)  [(6) Word Break (print all ways) (Word Break II)](#_7jzq7i5v2qxg)  [Week 11 - Divide and Conquer](#_xm16b6vjomgb)  [Editorial](#_rd1g637vebv)  [(1) 1/N-th root of an integer (use binary search) (square root, cube root, ..)](#_3yj193yricnf)  [(2) Matrix Median](#_s8mv36xitsq6)  [(3) Find the element that appears once in sorted array, and rest element appears twice (Binary search)](#_96awmv3z0a83)  [(4) Search element in a sorted and rotated array/ find pivot where it is rotated](#_xlcdb7pfecs5)  [(5) Median of 2 sorted arrays](#_4iju4isxgn1q)  [(6) K-th element of two sorted arrays](#_ayfcgvcnvg1d)  [Week 12 - Bits / Bit Manipulation](#_f76xv3hocl55)  [Editorial](#_fmr4ud5cz27)  [(1) Check if a number if a power of 2 or not in O(1)](#_l0k0zplk5s7j)  [(2) Count total set bits](#_74dtyo8xiba)  [(3) Divide Integers without / operator](#_i5s51bpkc1uw)  [(4) Power Set (this is very important)](#_5lo259ivtqv9)  [(5) Find MSB in o(1)](#_z6yg0ixjwi55)  [(6) Find square of a number without using multiplication or division operators.](#_6u0vzm54fcft)  [Practice (Week 12)](#_b9ty91m3sgdc)  [Q1. Find elements that occurs once while other elements occur M times](#_1ops11rb3lvm)  [Week 13 - Stack and Queue](#_r6mvxc25te7r)  [Editorial](#_mp6gf4sfocvn)  [(1) Implement Stack / Implement Queue [In Progress]](#_fe78b23ku8yy)  [(2) BFS](#_ucjp0i3nawss)  [(3) Implement Stack using Queue [In Progress]](#_cqm0r8nafctx)  [(4) Implement Queue using Stack](#_rgbauklopk5)  [(5) Check for balanced parentheses](#_jm7opfp3i41s)  [(6) Next Greater Element](#_opexqcekrha6)  [Week 14 - Advance Stack & Queue](#_3msrksnbiccv)  [Editorial](#_ttpqjvj1w4n7)  [(1) Next Smaller Element](#_u2e2q0z60tlm)  [(2) LRU cache (vvvv. imp)](#_fufy54qe0apx)  [(3) Largest rectangle in histogram](#_joe45qgoxty)  [(4) Sliding Window maximum](#_d9syzi36bps0)  [(5) Implement Min Stack [IN PROGRESS]](#_c66b4hrjxipf)  [(6) Rotten Orange (Using BFS)](#_lf8iism8svbe)  [Week 15 - String](#_1tr85o2hhaoi)  [Editorial](#_a9bxy2dkunr0)  [(1) Reverse Words in a String](#_js4v3da9671h)  [(2) Longest Palindrome in a string](#_jq0ww65usvhs)  [(3) Roman Number to Integer and vice versa](#_ksagvyd7kyn7)  [(4) Implement ATOI/STRSTR](#_gx8cjg2x21x)  [(5) Longest Common Prefix](#_4k57yujgg2zz)  [(6) Rabin Karp (Longest Duplicate Substring)](#_kzewokq699nq)  [Practice (Week 15)](#_ns2uhqvgh0sg)  [Q1. Ways to split string into two palindromes (Rabin Karp Application)](#_sgnm71islabl)  [Week 16 - Advance String](#_a6p7hu6qllyy)  [Editorial](#_1u198oxfuhtj)  [(1) Prefix Function/Z-Function](#_xxqc3rgr5bw3)  [(2) KMP algo](#_nesvzfhiibhe)  [(3) Minimum characters needed to be inserted in the beginning to make it palindromic.](#_py6o4if4lr54)  [(4) Check for Anagrams [IN PROGRESS]](#_cojvwix3w6ag)  [(5) Count and Say](#_h63itdx4lsad)  [(6) Compare version numbers](#_7znmn7l66szl)  [Week 17 - Binary Tree (Easy)](#_niih9133bnhw)  [Editorial](#_xf2e7ntt4l9m)  [(1) Inorder Traversal (with recursion and without recursion)](#_a9bxw3bcqfdk)  [(2) Preorder Traversal (with recursion and without recursion)](#_z69k2hrdaltd)  [(3) Postorder Traversal (with recursion and without recursion)](#_70bfp2h3mm8m)  [(4) LeftView Of Binary Tree](#_4tartjtg4mqd)  [(5) Bottom View of Binary Tree](#_7pqtbxmhve1l)  [(6) Top View of Binary Tree](#_bnc6beixchy9)  [Week 18 - Binary Tree (Medium)](#_w33me1zb4szv)  [Editorial](#_e4k4sgafhw96)  [(1) Level order Traversal / Level order traversal in spiral form](#_kxshhg6j6lkc)  [(2) Height of a Binary Tree](#_vt53ds7fbzqf)  [(3) Diameter of Binary Tree](#_31wflmny2dav)  [(4) Check if Binary tree is height balanced or not](#_q976m3tov1cv)  [(5) LCA in Binary Tree](#_cxiq6in9ftkm)  [(6) Check if two trees are identical or not](#_jkyi4r8rx1sd)  [Week 19 - Binary Tree (Advance)](#_wfxf6g9l3p9g)  [Editorial](#_3qw4ujehb9ni)  [(1) Maximum path sum](#_gbntj4dre5cs)  [(2) Construct Binary Tree from inorder and preorder](#_bqwdj2eoceeu)  [(3) Construct Binary Tree from Inorder and Postorder](#_lz61en7lnd63)  [(4) Symmetric Binary Tree](#_a2pgbgs5hkww)  [(5) Flatten Binary Tree to LinkedList](#_gnm0924qpdto)  [(6) Check if Binary Tree is mirror of itself or not](#_29jl2qa5xp)  [Week 20 - Binary Search Tree](#_t37106mw5j2d)  [Editorial](#_hp9n3er2q5g9)  [(1) Populate Next Right pointers of Tree](#_dq6g3jufcmt8)  [(2) Search given Key in BST](#_q04aozdcim82)  [(3) Construct BST from given keys.](#_5dgusc3qa44e)  [(4) Check is a BT is BST or not](#_97pjz3nfc8oi)  [(5) Find LCA of two nodes in BST](#_jdbcxsua7vnd)  [(6) Find the inorder predecessor/successor of a given Key in BST.](#_vpk4usbt49rf)  [Week 21 - Advance Binary Search Tree](#_m320oph3onqi)  [Editorial](#_jg83m1qij8th)  [(1) Floor and Ceil in a BST](#_h33p7t7mq50g)  [(2) Find K-th smallest and K-th largest element in BST (2 different Questions)](#_xku0m7o6qdl4)  [(3) Find a pair with a given sum in BST](#_ons5vw6wudj7)  [(4) BST iterator](#_dphj2b3ceiy0)  [(5) Size of the largest BST in a Binary Tree](#_1jdhlcwa2jmg)  [(6) Serialize and deserialize Binary Tree](#_ovi83599he0m)  [Week 22 - Mixed Questions](#_mdykep8qx0jk)  [Editorial](#_qr5mrj4qjngn)  [(1) Binary Tree to Double Linked List](#_bwrkiensxtvs)  [(2) Find median in a stream of running integers.](#_4dlygwecjkw9)  [(3) K-th largest element in a stream.](#_knn0sts4bof7)  [(4) Distinct numbers in Window.](#_x1pdap6tbqrz)  [(5) K-th largest element in an unsorted array.](#_qhex4wnxiaux)  [(6) Flood-fill Algorithm](#_nf9nrv5o9i0o)  [Practice (Week 22)](#_a4lfbthgewd)  [Q1. Brick Wall](#_d39a9fwne2l6)  [Q2. Minimum Area Rectangle](#_usmiaa9l5g2g)  [Q3. Maximum Equal Frequency](#_r0lonixyb5z0)  [Week 23 - Graph](#_nj5nzl965g1z)  [Editorial](#_h9531ykp6cn1)  [(1) Clone a graph (Not that easy as it looks)](#_1axz29jm9wp3)  [(2) DFS](#_kh82t2c3vxr3)  [(3) BFS](#_vmo6bqdyrrwo)  [(4) Detect A cycle in Undirected Graph/Directed Graph](#_a6yj1rob7mmu)  [(5) Topo Sort](#_75p0c4wk059e)  [(6) Number of islands (Do in Grid and Graph both)](#_boeab09kdp9q)  [(7) Bipartite Check](#_x9g23jke6cpf)  [Week 24 - Advance Graph](#_q5zza619inw6)  [Editorial](#_67b8oov2o38j)  [(1) SCC(using KosaRaju’s algo)](#_afc5g93e1st1)  [(2) Djisktra’s Algorithm](#_2yejqfb50gqu)  [(3) Bellman Ford Algo](#_99c36u9scslp)  [(4) Floyd Warshall Algorithm](#_p95thvyqn4pf)  [(5) MST using Prim’s Algo](#_ix5jad7yknh0)  [(6) MST using Kruskal’s Algo](#_kocqh1677xnd)  [Week 25 - Dynamic Programming](#_1tgq6ye49zy6)  [Editorial](#_y5clofiq52fz)  [(1) Max Product Subarray](#_1l98be9di2py)  [(2) Longest Increasing Subsequence](#_8oznmf1hnbi5)  [(3) Longest Common Subsequence](#_yqgqnjb8lsff)  [(4) 0-1 Knapsack](#_g4h3web2eqci)  [(5) Edit Distance](#_wlvg8hc938u0)  [(6) Maximum sum increasing subsequence](#_xxzo86kuun5e)  [(7) Matrix Chain Multiplication](#_3tf048p9n6b5)  [Week 26 - Advance Dynamic Programming](#_x6if7knqk0ud)  [Editorial](#_aabxnzzdf8ds)  [(1) Maximum sum path in matrix, (count paths, and similar type do, also backtrack to find the maximum path)](#_1gh9y7hvtqh1)  [(2) Coin change [IN PROGRESS]](#_dscn9kyo5wrv)  [(3) Subset Sum [IN PROGRESS]](#_6um9hmlfyter)  [(4) Rod Cutting](#_kza9stsrwuer)  [(5) Egg Dropping](#_2ajbs9p8ih9y)  [(6) Word Break](#_v2nnmrhhp5b9)  [(7) Palindrome Partitioning (MCM Variation)](#_g1diwjhdrcxt)  [Practice (Week 26)](#_ct0y389z5va4)  [Q1. Subset Sum Partition](#_ky4fbaeptczd)  [Week 27 - Heap](#_2yocjaf3bxw2)  [Practice (Week 27)](#_zd1inwddr9z6)  [Q1. Top K Frequent Words/Elements](#_ftgth9lmagsd)  [Week 28 - OS (Operating System)](#_ovx82f85bob6)  [(1) Love Babbar Cheat Sheet](#_juc8xedfh16y)  [Sub-Section Name](#_jnim805scuvp)  [(1) Resource name is written over here](#_xifkuoskgsak)  [Week 29 - DBMS (Database Management System)](#_riddpgtmtcqb)  [(1) Love Babbar RoadMap](#_398jqto1vs7v)  [Sub-Section Name](#_c3m6pedu3886)  [(1) Resource name is written over here](#_m7mzypd4qjpa)  [Week 30 - CN (Computer Networking)](#_5ugod13j48rt)  [(1) Resource name is written over here](#_rwph7p198xw6)  [Sub-Section Name](#_frxearjx6owk)  [(1) Resource name is written over here](#_cnptpc6yc5r8)  [Week 31 - OOP (Object Oriented Programming)](#_6pwhwj82x5nu)  [(1) Resource name is written over here](#_rts8g7v17zwk)  [Sub-Section Name](#_c59butk89ltk)  [(1) Resource name is written over here](#_qfvfctbo1t5j)  [Week 32 - System Design](#_man3npihet01)  [Sub-Section Name](#_lxxplq85d9hk)  [(1) Resource name is written over here](#_fko0z8k91dwi) |

# Week 1 - Arrays

## Editorial

#### (1) Find the duplicate in an array of N+1 integers.

**Question** :

**Question A** (Find the Duplicate Number):

<https://leetcode.com/problems/find-the-duplicate-number/>

**Question B** (Find All Duplicates in an Array):

<https://leetcode.com/problems/find-all-duplicates-in-an-array/>

**Solutions** :

**Solution A** :

take U forward’s Video :

<https://www.youtube.com/watch?v=32Ll35mhWg0>

Algorithms Made Easy’s Video :

<https://www.youtube.com/watch?v=0U4e11Z7Vcs>

GFG Editorial (5 Approaches) :

<https://www.geeksforgeeks.org/find-repetitive-element-1-n-1/>

LeetCode Article (3 Approaches) :

<https://leetcode.com/problems/find-the-duplicate-number/solution/>

**Solution B** :

Jason Chang’s Video :

<https://www.youtube.com/watch?v=OM7yWsiRVGI>

Anish Malla’s Video :

<https://www.youtube.com/watch?v=kRrSeAZRD6E>

Nick White’s Video :

<https://www.youtube.com/watch?v=aMsSF1Il3IY>

Michael Muinos’s Video :

<https://www.youtube.com/watch?v=lYxEdtR5_xQ>

#### (2) Sort an array of 0’s 1’s 2’s without using extra space or sorting algo (Sort Colors)

**Question** :

<https://leetcode.com/problems/sort-colors/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=oaVa-9wmpns>

Aalekh Jain’s Video :

<https://www.youtube.com/watch?v=yjnErPy6ul8>

Algorithms Made Easy’s Video :

<https://www.youtube.com/watch?v=4SNs8G-yByo>

GFG Editorial :  
 <https://www.geeksforgeeks.org/sort-an-array-of-0s-1s-and-2s/>

#### (3) Repeat and Missing Number

**Question** :

<https://practice.geeksforgeeks.org/problems/find-missing-and-repeating/0>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=5nMGY4VUoRY>

GFG Editorial :  
 <https://www.geeksforgeeks.org/find-a-repeating-and-a-missing-number/>

#### (4) Merge two sorted Arrays without extra space

**Question** :

<https://practice.geeksforgeeks.org/problems/merge-two-sorted-arrays/0>

**Solutions** :

Dream FAANG’s Video (with Intuition O(nlogn + mlogm)) :

<https://youtu.be/JdS87OM_CPg?t=97>

take U forward’s Video :  
<https://www.youtube.com/watch?v=hVl2b3bLzBw>

GFG Editorial :  
 <https://www.geeksforgeeks.org/merge-two-sorted-arrays-o1-extra-space/>

#### (5) Kadane’s Algorithm

**Question** :

<https://leetcode.com/problems/maximum-subarray/>

OR

<https://www.interviewbit.com/problems/max-sum-contiguous-subarray/>

OR

<https://practice.geeksforgeeks.org/problems/kadanes-algorithm-1587115620/1>

**Solutions** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=w_KEocd__20>

Michael Muinos’s Video :

<https://www.youtube.com/watch?v=tmakGVOGV3A&ab_channel=MichaelMuinos>

Back To Back SWE’s Video :   
 <https://www.youtube.com/watch?v=2MmGzdiKR9Y&ab_channel=BackToBackSWE>

mycodeschool’s Video (3 Approaches):

<https://www.youtube.com/watch?v=ohHWQf1HDfU>

QuanticDev’s Video (with Proof):

<https://www.youtube.com/watch?v=4csAswCkXZM>

GFG Editorial :  
 <https://www.geeksforgeeks.org/largest-sum-contiguous-subarray/>

#### (6) Merge Overlapping Subintervals (Merge Intervals)

**Question** :

<https://leetcode.com/problems/merge-intervals/>

**Solutions** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=2JzRBPFYbKE>

Giuseppe Picciano’s Video :

<https://www.youtube.com/watch?v=SXPvqPFX_VE>

leetuition’s Video :  
 <https://www.youtube.com/watch?v=ygaBzC_qY0w>

GFG Editorial :   
 <https://www.geeksforgeeks.org/merging-intervals/>

## Practice (Week 1)

#### Q1. Find All Numbers Disappeared in an Array

**Problem** :

<https://leetcode.com/problems/find-all-numbers-disappeared-in-an-array>

**Solutions** :

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=CTBEcmzLAuA>

#### Q2. Smallest Missing Integer

**Problem** :

**Problem A** (First Missing Positive):

<https://leetcode.com/problems/first-missing-positive/>

**Problem B** (Positive Mex):

<https://www.codechef.com/LTIME83B/problems/MEXUM>

**Solutions** :

**Solution A** :

Rachit Jain’s Video :

<https://www.youtube.com/watch?v=-lfHWWMmXXM>

Michael Muinos’s Video :

<https://www.youtube.com/watch?v=9SnkdYXNIzM>

Knapsak’s Video :

<https://www.youtube.com/watch?v=vDYzpUULJ8E>

**Solution B** :

Algorythm’s Video :

<https://www.youtube.com/watch?v=__sclWOOhOU>

#### Q3. Flipping Game

**Problem** :

<https://codeforces.com/problemset/problem/327/A>

**Solutions** :

GFG Editorial :  
 <https://www.geeksforgeeks.org/maximize-number-0s-flipping-subarray/>

alGOds’s Video (Kadane’s Algorithm Approach) :

<https://www.youtube.com/watch?v=cLVpE5q_-DE>

The Coding Guy’s Video (Kadane’s Algorithm Approach) :

<https://www.youtube.com/watch?v=FtiWd8PIpxI>

Stoover Coding’s Video (Dynamic Programming Approach) :

<https://www.youtube.com/watch?v=zF0UgUXGnuU>

#### Q4. Insert Interval

**Problem** :

<https://leetcode.com/problems/insert-interval/>

**Solutions** :

CS with KV’s Video :

<https://www.youtube.com/watch?v=RCJW_y_Cogk>

#### Q5. Product of Array Except Self

**Problem** :

<https://leetcode.com/problems/product-of-array-except-self/>

**Solutions** :

Algorythm’s Video :

<https://www.youtube.com/watch?v=PEzX2if5zZA>

Nick White’s Video :

<https://www.youtube.com/watch?v=tSRFtR3pv74>

Errichto’s Video :

<https://www.youtube.com/watch?v=E0FqAbHjf4E>

LeetCode Article :

<https://leetcode.com/problems/product-of-array-except-self/solution/>

#### Q6. Duplicate Zeros

**Problem** :

<https://leetcode.com/problems/duplicate-zeros/>

**Solutions** :

LeetCode Article :

<https://leetcode.com/problems/duplicate-zeros/solution/>

#### Q7. Wiggle Sort II

**Problem** :

<https://leetcode.com/problems/wiggle-sort-ii/>

**Solutions** :

Coding Blocks’s Video :

<https://www.youtube.com/watch?v=di7gNqxfU1g>

AfterAcademy Editorial :

<https://afteracademy.com/blog/wave-array>

GFG Editorial :

<https://www.geeksforgeeks.org/sort-array-wave-form-2/>

Medium Article :

<https://medium.com/enjoy-algorithm/sort-an-array-in-wave-form-9c88ed34b7f2>

#### Q8. Recursive Insertion Sort

**Problem** :

<https://afteracademy.com/problems/recursive-insertion-sort>

**Solutions** :

AfterAcademy Editorial :

<https://afteracademy.com/blog/recursive-insertion-sort>

GFG Editorial :

<https://www.geeksforgeeks.org/recursive-insertion-sort/>

# Week 2 - Multi Dimensional Arrays

## Editorial

#### (1) Set Matrix Zeros

**Question** :

<https://leetcode.com/problems/set-matrix-zeroes/>

**Solutions** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=M65xBewcqcI>

Errichto’s Video :

<https://www.youtube.com/watch?v=6_KMkeh5kEc>

#### (2) Pascal Triangle

**Question** :

**Question A** (Pascal's Triangle):

<https://leetcode.com/problems/pascals-triangle/>

**Question B** (Pascal’s Triangle II):

<https://leetcode.com/problems/pascals-triangle-ii/>

**Solutions** :

**Solution A** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=6FLvhQjZqvM>

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=7pOzP9m_bX8>

Michael Muinos’s Video :  
 <https://www.youtube.com/watch?v=VJBUH3chC64>

GFG Editorial :  
 <https://www.geeksforgeeks.org/pascal-triangle/>

**Solution B** :

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=tTYU4PAiqOE>

#### (3) Next Permutation

**Question** :

<https://leetcode.com/problems/next-permutation/>

**Solutions** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=LuLCLgMElus>

Back To Back SWE’s Video :

<https://www.youtube.com/watch?v=quAS1iydq7U>

GFG Editorial :

<https://www.geeksforgeeks.org/find-the-next-lexicographically-greater-word-than-a-given-word/>

#### (4) Inversion of Array (Using Merge Sort)

**Question** :

<https://practice.geeksforgeeks.org/problems/inversion-of-array/0>

**Solutions** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=kQ1mJlwW-c0>

GFG Editorial :

<https://www.geeksforgeeks.org/counting-inversions/>

#### (5) Stock Buy and Sell

**Question** :

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock/>

**Solutions** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=eMSfBgbiEjk>

Inside code’s Video :  
<https://www.youtube.com/watch?v=hOLSBR7eN4g>

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=mmIMpgh67vg>

Jesse Dietrichson’s Video :

<https://www.youtube.com/watch?v=X6i_I-JiB4Y>

#### (6) Rotate Matrix

**Question** :

<https://leetcode.com/problems/rotate-image/>

OR

<https://www.interviewbit.com/problems/rotate-matrix/>

**Solutions** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=Y72QeX0Efxw>

My Programming Sandbox’s Video (Long Approach) :

<https://www.youtube.com/watch?v=Jtu6dJ0Cb94>

Michael Muinos’s Video (Short Approach) :

<https://www.youtube.com/watch?v=J-Ihez5cwCM>

Nick White’s Video (Short Approach) :

<https://www.youtube.com/watch?v=SA867FvqHrM>

GFG Editorial (Long Approach) :

<https://www.geeksforgeeks.org/inplace-rotate-square-matrix-by-90-degrees/>

GFG Editorial (Short Approach) :

<https://www.geeksforgeeks.org/rotate-matrix-90-degree-without-using-extra-space-set-2/>

## Practice (Week 2)

#### Q1. Game of Life

**Problem** :

<https://leetcode.com/problems/game-of-life/>

**Solutions** :

LeetCode Article :

<https://leetcode.com/problems/game-of-life/solution/>

happygirlzt’s Video :

<https://www.youtube.com/watch?v=sUqYZvfZ9UE>

GFG Editorial :

<https://www.geeksforgeeks.org/program-for-conways-game-of-life-set-2/>

#### Q2. Best Time to Buy and Sell Stock II

**Problem** :

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/>

**Solutions** :

Inside code’s Video :

<https://www.youtube.com/watch?v=GtFhszN6VIg>

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=Q-8JkdUliVM>

LeetCode Article :

<https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/solution/>

GFG Editorial :

<https://www.geeksforgeeks.org/stock-buy-sell/>

Errichto’s Video :

<https://www.youtube.com/watch?v=MTnFIF2I2gw>

#### Q3. Spiral Matrix Traversal

**Problem** :

**Problem A** (Spiral Matrix):

<https://leetcode.com/problems/spiral-matrix/>

**Problem B** (Spiral Matrix III):

<https://leetcode.com/problems/spiral-matrix-iii/>

**Solutions** :

**Solution A** :

mycodeschool’s Video :

<https://www.youtube.com/watch?v=siKFOI8PNKM>

GFG Editorial :

<https://www.geeksforgeeks.org/print-a-given-matrix-in-spiral-form/>

**Solution B** :

LeetCode Article :

<https://leetcode.com/articles/spiral-matrix-iii/>

#### Q4. Max Sum of Rectangle No Larger Than K

**Problem** :

<https://leetcode.com/problems/max-sum-of-rectangle-no-larger-than-k/>

**Solutions** :

Back To Back SWE’s Video (Using Kadane’s Algorithm in 2D Grid):

<https://www.youtube.com/watch?v=-FgseNO-6Gk>

#### Q5. NEKO's Maze Game

**Problem** :

<https://codeforces.com/contest/1292/problem/A>

**Solutions** :

Errichto’s Video :

<https://www.youtube.com/watch?v=mhrvlor1qH0>

# Week 3 - Math

## Editorial

#### (1) Search in a 2D matrix

**Question** :

**Question A** (Search a 2D Matrix):

<https://leetcode.com/problems/search-a-2d-matrix/>

**Question B** (Search a 2D Matrix II):

<https://leetcode.com/problems/search-a-2d-matrix-ii/>

**Solutions** :

**Solution A** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=ZYpYur0znng>

Amell Peralta’s Video :

<https://www.youtube.com/watch?v=dHJDhsvBd8c>

Back To Back SWE’s Video (General Approach) :

<https://www.youtube.com/watch?v=FOa55B9Ikfg>

**Solution B** :

take U forward’s Video :  
 <https://www.youtube.com/watch?v=ZYpYur0znng>

Amell Peralta’s Video :

<https://www.youtube.com/watch?v=Ohke9-qwAKU>

GitHub Article (Based on Amell Peralta’s Explanation) :

<https://github.com/eMahtab/search-2D-matrix-ii>

Giuseppe Picciano’s Video (With Visual Explanation):

<https://www.youtube.com/watch?v=OeHVnd_MTuc>

Back To Back SWE’s Video (General Approach) :

<https://www.youtube.com/watch?v=FOa55B9Ikfg>

#### (2) Pow(X,n)

**Question** :

<https://leetcode.com/problems/powx-n/>

**Solutions** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=l0YC3876qxg>

mycodeschool’s Video (Exponentiation - Using Recursion):

<https://www.youtube.com/watch?v=wAyrtLAeWvI>

Algorithms Made Easy’s Video :

<https://www.youtube.com/watch?v=JMHL9geRAKI>

GFG Editorial :   
 <https://www.geeksforgeeks.org/write-a-c-program-to-calculate-powxn/>

#### (3) Majority Element (>N/2 times) [SEARCH FOR MORE Explanations]

**Question** :

<https://leetcode.com/problems/majority-element/>

**Solutions** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=AoX3BPWNnoE>

#### (4) Majority Element (>N/3 times) [SEARCH FOR MORE Explanations]

**Question** :

<https://leetcode.com/problems/majority-element-ii/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=yDbkQd9t2ig>

#### (5) Grid Unique Paths

**Question** :

<https://leetcode.com/problems/unique-paths/>

**Solutions** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=t_f0nwwdg5o>

Don't Memorise’s Video (Mathematical Optimization Intuition) :

<https://www.youtube.com/watch?v=fpnNaAU0iPk>

LeetCode Discussion (Mathematical Approach) :

<https://leetcode.com/problems/unique-paths/discuss/22958/Math-solution-O(1)-space/175698>

Michael Muinos’s Video (DP Approach) :

<https://www.youtube.com/watch?v=4Zq2Fnd6tl0>

Amell Peralta’s Video (DP Approach | Space Complexity - O(n^2)) :

<https://www.youtube.com/watch?v=rdu3YVZ3KD4>

Amell Peralta’s Video (DP Approach | Space Complexity - O(n)) :

<https://www.youtube.com/watch?v=RZz5M3iidBI>

Khushboo Goel’s Video (DP Approach | Space Complexity - O(n)) :

<https://www.youtube.com/watch?v=fEcyKrdIkho>

GFG Editorial :

<https://www.geeksforgeeks.org/count-possible-paths-top-left-bottom-right-nxm-matrix/>

#### (6) Reverse Pairs [SEARCH FOR MORE Explanations]

**Question** :

<https://leetcode.com/problems/reverse-pairs/>

**Solutions** :

take U forward’s Video :  
 <https://www.youtube.com/watch?v=S6rsAlj_iB4>

## Practice (Week 3)

#### Q1. Excel Sheet Column Number & Title Conversion

**Problem** :

**Problem A** (Excel Column Number):

<https://leetcode.com/problems/excel-sheet-column-number/>

**Problem B** (Excel Sheet Column Title):

<https://leetcode.com/problems/excel-sheet-column-title/>

**Solutions** :

**Solution A** :

GFG Editorial :

<https://www.geeksforgeeks.org/find-excel-column-number-column-title/>

IDeserve’s Video :

<https://www.youtube.com/watch?v=77HYaBDcGuQ>

**Solution B** :

GFG Editorial :

<https://www.geeksforgeeks.org/find-excel-column-name-given-number/>

IDeserve’s Video :

<https://www.youtube.com/watch?v=77HYaBDcGuQ>

#### Q2. Factorial Trailing Zeroes

**Problem** :

<https://leetcode.com/problems/factorial-trailing-zeroes/>

**Solutions** :

Scaler Academy’s Video :

<https://www.youtube.com/watch?v=wkvVdggCSeo>

GFG Editorial :

<https://www.geeksforgeeks.org/count-trailing-zeroes-factorial-number/>

#### Q3. Find GCD in Log N

**Problem** :

<https://practice.geeksforgeeks.org/problems/gcd-of-two-numbers/0>

**Solutions** :

mycodeschool’s Video :

<https://www.youtube.com/watch?v=7HCd074v8g8>

Gaurav Sen’s Video :

<https://www.youtube.com/watch?v=80pOI0_BXyk>

GFG Editorial :

<https://www.geeksforgeeks.org/euclidean-algorithms-basic-and-extended/>

#### Q4. Sqrt(x) / Valid Perfect Square

**Problem** :

**Problem A** (Sqrt(x)):

<https://leetcode.com/problems/sqrtx/>

**Problem B** (Valid Perfect Square):

<https://leetcode.com/problems/valid-perfect-square/>

**Solutions** :

**Solution A** :

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=VYtEKhxKd1Q>

Scaler Academy’s Video :

<https://www.youtube.com/watch?v=fItuKa_tIpY>

**Solution B** :

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=-oqZCmhJ2Zs>

#### Q5. Roman & Integer Conversion

**Problem** :

**Problem A** (Roman to Integer):

<https://leetcode.com/problems/roman-to-integer/>

**Problem B** (Integer to Roman):

<https://leetcode.com/problems/integer-to-roman/>

**Solutions** :

**Solution A** :

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=9rfe5nXL05Q>

GFG Editorial :

<https://www.geeksforgeeks.org/converting-roman-numerals-decimal-lying-1-3999/>

**Solution B** :

Michael Muinos’s Video :

<https://www.youtube.com/watch?v=yzB4M-UXqgI>

Krishna Teaches’s Video :

<https://www.youtube.com/watch?v=zNlm28-ZdaE>

GFG Editorial :

<https://www.geeksforgeeks.org/converting-decimal-number-lying-between-1-to-3999-to-roman-numerals/>

#### Q6. Maximum Swap

**Problem** :

<https://leetcode.com/problems/maximum-swap/>

**Solutions** :

Ren Zhang’s Video (Time - O(n) & Space - O(1)) :

<https://www.youtube.com/watch?v=arecn8VuQL0>

GFG Editorial (Time - O(n) & Space - O(1)) :  
<https://www.geeksforgeeks.org/largest-number-with-one-swap-allowed/>  
  
happygirlzt’s Video (Time - O(n) & Space - O(n)) :

<https://www.youtube.com/watch?v=YM4_hj7AWrI>

LeetCode Article (Time - O(n) & Space - O(n)) :

<https://leetcode.com/problems/maximum-swap/solution/>

#### Q7. Tower of Hanoi

**Problem** :

<https://practice.geeksforgeeks.org/problems/help-the-old-man/0>

**Solutions** :

Inside code’s Video (Visual Recursive Approach) :

<https://www.youtube.com/watch?v=UR1lOWMvAIA>

Reducible’s Video (Visual Recursive Approach) :

<https://www.youtube.com/watch?v=rf6uf3jNjbo>

AlgoData’s Video (Visual Recursive Approach) :

<https://www.youtube.com/watch?v=fffbT41IuB4>

Medium Article (Recursive Approach) :

<https://medium.com/@jamalmaria111/tower-of-hanoi-js-algorithm-3f667fa46f0f>

GFG Editorial (Recursive Approach):

<https://www.geeksforgeeks.org/c-program-for-tower-of-hanoi/>

GFG Editorial (Time Complexity Analysis of Recursive Approach):

<https://www.geeksforgeeks.org/time-complexity-analysis-tower-hanoi-recursion/>

GFG Editorial (**Iterative** Approach):

<https://www.geeksforgeeks.org/iterative-tower-of-hanoi/>

Khan Academy Article :

<https://www.khanacademy.org/computing/computer-science/algorithms/towers-of-hanoi/a/towers-of-hanoi>

Medium Article (**3 Approaches**) :

<https://medium.com/datadriveninvestor/tower-of-hanoi-solve-and-optimize-with-memoization-f215a1bd201f>

#### Q8. Count Primes (Sieve of Eratosthenes)

**Problem** :  
 <https://practice.geeksforgeeks.org/problems/sieve-of-eratosthenes5242/1>

OR

<https://leetcode.com/problems/count-primes/>

**Solutions** :

GeeksforGeeks’s Video (Best **Sieve of Eratosthenes** Explanation):

<https://www.youtube.com/watch?v=NZ7-ntEgt6g>

GeeksforGeeks’s Video (**Segmented Sieve** Explanation):

<https://www.youtube.com/watch?v=j0M8SF6daSs>

GFG Editorial (**Simple Sieve** of Eratosthenes (O(N\*log(logN))):  
 <https://www.geeksforgeeks.org/sieve-of-eratosthenes/>

GFG Editorial : Why **Sieve of Eratosthenes** has time complexity of **O(N\*log(logN))**?

<https://www.geeksforgeeks.org/how-is-the-time-complexity-of-sieve-of-eratosthenes-is-nloglogn/>

GFG Editorial (**Segmented Sieve** (Only Space Optimal)):  
 <https://www.geeksforgeeks.org/segmented-sieve/>

AND

<https://www.geeksforgeeks.org/segmented-sieve-print-primes-in-a-range/>

---------------------------------------------------------------------------------------------------------------------

Terrible Whiteboard’s Video (**Sieve of Eratosthenes** Explanation) :

<https://www.youtube.com/watch?v=PypkiVlTRa4>

mycodeschool’s Video (**Sieve of Eratosthenes** Explanation) :

<https://www.youtube.com/watch?v=eKp56OLhoQs>

GFG Editorial (**Advance Sieve** of Eratosthenes (O(n))):  
 <https://www.geeksforgeeks.org/sieve-eratosthenes-0n-time-complexity/>

GFG Editorial (**Bitwise Sieve**):

<https://www.geeksforgeeks.org/bitwise-sieve/>

# Week 4 - Hashing

## Editorial

#### (1) 2 Sum problem

**Question** :

<https://leetcode.com/problems/two-sum/>

**Solutions** :

take U forward’s Video :  
<https://www.youtube.com/watch?v=dRUpbt8vHpo>

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=U8B984M1VcU>

Amell Peralta’s Video :

<https://www.youtube.com/watch?v=QW61VBhH10A>

Jesse Dietrichson’s Video :

<https://www.youtube.com/watch?v=LB62Atgt0xM>

LeetCode Article :

<https://leetcode.com/problems/two-sum/solution/>

#### (2) 4 Sum problem

**Question** :

<https://leetcode.com/problems/4sum/>

OR

<https://www.interviewbit.com/problems/4-sum/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=4ggF3tXIAp0>

LeetCode Article :

<https://leetcode.com/problems/4sum/solution/>

#### (3) Longest Consecutive Sequence

**Question** :

<https://leetcode.com/problems/longest-consecutive-sequence/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=qgizvmgeyUM>

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=xdMyL--dOqE>

Byte By Byte’s Video (with Time Complexity Analysis) :

<https://www.youtube.com/watch?v=rpku4iVaDNU>

Knapsak’s Video (with Code Walkthrough) :

<https://www.youtube.com/watch?v=Awc7f5mCtks>

LeetCode Article :

<https://leetcode.com/articles/longest-consecutive-sequence/>

GFG Editorial :

<https://www.geeksforgeeks.org/longest-consecutive-subsequence/>

#### (4) Longest Subarray with K sum

**Question** :

**Question A** (Longest Subarray with 0 sum):

<https://practice.geeksforgeeks.org/problems/largest-subarray-with-0-sum/1>

**Question B** (Maximum Size Subarray Sum Equals k **or** Longest Sub-Array with Sum K):

<https://www.lintcode.com/problem/maximum-size-subarray-sum-equals-k/description>   
 **or**  
<https://practice.geeksforgeeks.org/problems/longest-sub-array-with-sum-k/0>

**Solutions** :

**Solution A** :

take U forward’s Video :

<https://www.youtube.com/watch?v=xmguZ6GbatA>

GFG Editorial :

<https://www.geeksforgeeks.org/find-the-largest-subarray-with-0-sum/>

**Solution B** :

GFG Editorial :

<https://www.geeksforgeeks.org/longest-sub-array-sum-k/>

#### (5) Count number of subarrays with given XOR

**Question** :

<https://www.interviewbit.com/problems/subarray-with-given-xor/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=lO9R5CaGRPY>

GFG Editorial :

<https://www.geeksforgeeks.org/count-number-subarrays-given-xor/>

#### (6) Longest substring without repeat

**Question** :

<https://leetcode.com/problems/longest-substring-without-repeating-characters/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=qtVh-XEpsJo>

Michael Muinos’s Video :

<https://www.youtube.com/watch?v=4i6-9IzQHwo>

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=WKTgajDkVcA>

Scaler Academy’s Video :

<https://www.youtube.com/watch?v=pJZF_VCxG9I>

GFG Editorial :

<https://www.geeksforgeeks.org/length-of-the-longest-substring-without-repeating-characters/>

## Practice (Week 4)

#### Q1. Valid Sudoku

**Problem** :

<https://leetcode.com/problems/valid-sudoku/>

**Solutions** :

Nick White’s Video :

<https://www.youtube.com/watch?v=Pl7mMcBm2b8>

#### Q2. Contiguous Array [WRITE A CLEAN EXPLANATION]

**Problem** :

<https://leetcode.com/problems/contiguous-array/>

**Solutions** :  
 Knowledge Center’s Video (**Intuitive** Explanation):

<https://www.youtube.com/watch?v=VM5Mh3-UFPg>

daose’s Video (**Intuitive** Explanation):

<https://www.youtube.com/watch?v=63ogoiDrd4g>

Nick White’s Video :

<https://www.youtube.com/watch?v=nSEO5zOwP7g>

LeetCode Article :

<https://leetcode.com/problems/contiguous-array/solution/>

GFG Editorial :

<https://www.geeksforgeeks.org/largest-subarray-with-equal-number-of-0s-and-1s/>

#### Q3. Insert Delete GetRandom in O(1) Time

**Problem** :

**Problem A** (Insert Delete GetRandom O(1)):

<https://leetcode.com/problems/insert-delete-getrandom-o1/>

**Problem B** (Insert Delete GetRandom O(1) - Duplicates allowed):

<https://leetcode.com/problems/insert-delete-getrandom-o1-duplicates-allowed/>

**Solutions** :

**Solution A** :

Aalekh Jain’s Video :

<https://www.youtube.com/watch?v=AynZwcnFHwE>

Michael Muinos’s Video :

<https://www.youtube.com/watch?v=TD2g8UjXMLA>

Algorithms Made Easy’s Video :

<https://www.youtube.com/watch?v=yeTmZxk6-v4>

UBlog’s Video :

<https://www.youtube.com/watch?v=kp3E4N7H1AA>

GFG Editorial :

<https://www.geeksforgeeks.org/design-a-data-structure-that-supports-insert-delete-search-and-getrandom-in-constant-time/>

**Solution B** :

LeetCode Article:

<https://leetcode.com/problems/insert-delete-getrandom-o1-duplicates-allowed/solution/>

GFG Editorial :

<https://www.geeksforgeeks.org/design-a-data-structure-that-supports-insert-delete-getrandom-in-o1-with-duplicates/>

#### Q4. Design HashSet/HashMap

**Problem** :

**Problem A** (Design HashSet):

<https://leetcode.com/problems/design-hashset/>

**Problem B** (Design HashMap): [SEARCH FOR MORE EXPLANATION]

<https://leetcode.com/problems/design-hashmap/>

**Solutions** :

**Solution A** :

Java Brains’s Video :

<https://www.youtube.com/watch?v=NrMaQL_4Npo>

FelixTechTips’s Video :

<https://www.youtube.com/watch?v=7y4p_ZPsttI>

GFG Editorial (Internal Working of HashSet in Java) :

<https://www.geeksforgeeks.org/internal-working-of-sethashset-in-java/>

**Solution B** :

Ranjith ramachandran’s Video :

<https://www.youtube.com/watch?v=c3RVW3KGIIE>

GFG Editorial (Internal Working of HashMap in Java) :

<https://www.geeksforgeeks.org/internal-working-of-hashmap-java/>

#### Q5. Eugene and an array [WRITE A CLEAN EXPLANATION]

**Problem** :

<https://codeforces.com/problemset/problem/1333/C>

**Solutions** :

ProgrammerSought Editorial :

<https://www.programmersought.com/article/36934911835/>

Stefan Dascalescu’s Video :

<https://www.youtube.com/watch?v=XR_ZQvr9zyU>

#### Q6. Longest subarray with sum divisible/not divisible by K

**Problem** :

**Problem A** (Longest subarray with sum divisible by K): [WRITE A CLEAN EXPLANATION]

<https://practice.geeksforgeeks.org/problems/longest-subarray-with-sum-divisible-by-k1259/1>

**Problem B** (Length of longest subarray whose sum is **not** divisible by integer K):

<https://www.geeksforgeeks.org/length-of-longest-subarray-whose-sum-is-not-divisible-by-integer-k/>

**Solutions** :

**Solution A** :

GFG Editorial :

<https://www.geeksforgeeks.org/longest-subarray-sum-divisible-k/>

**Solution B** :

GFG Editorial :

<https://www.geeksforgeeks.org/length-of-longest-subarray-whose-sum-is-not-divisible-by-integer-k/>

#### Q7. Subarray Sums Divisible by K [WRITE A CLEAN EXPLANATION]

**Problem** :

<https://leetcode.com/problems/subarray-sums-divisible-by-k/>

**Solutions** :

LeetCode Comment (**Intuitive** Explanation):

<https://leetcode.com/problems/subarray-sums-divisible-by-k/discuss/217962/Java-Clean-O(n)-Number-Theory-%2B-Prefix-Sums>

GFG Editorial :

<https://www.geeksforgeeks.org/count-sub-arrays-sum-divisible-k/>

#### Q8. Flip Columns For Maximum Number of Equal Rows [WRITE A CLEAN EXPLANATION]

**Problem** :

<https://leetcode.com/problems/flip-columns-for-maximum-number-of-equal-rows/>

**Solutions** :

LeetCode Comment (@sourov\_roy):

<https://leetcode.com/problems/flip-columns-for-maximum-number-of-equal-rows/discuss/303847/Simple-C%2B%2B-Solution-with-comments>

Programming Live with Larry’s Video (Live Explanation):

[https://www.youtube.com/watch?v=xj3ltfSh9Io](https://www.youtube.com/watch?v=xj3ltfSh9Io&ab_channel=ProgrammingLivewithLarry)

# Week 5 - Linked List

## Editorial

#### (1) Reverse a LinkedList

**Question** :

<https://leetcode.com/problems/reverse-linked-list/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=iRtLEoL-r-g>

Log2Base2’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=4NWR385qeY4>

Coding with Conner’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=K-fqMtwbaPs>

Jesse Dietrichson’s Video (Recursive Approach) :

<https://www.youtube.com/watch?v=S92RuTtt9EE>

Joy Liu - Computer Psyc’s Video (Visual Recursive Approach) :

<https://www.youtube.com/watch?v=TzvmgkiDKkc>

CodeWhoop’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=PQIHq0vfADI>

mycodeschool’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=sYcOK51hl-A>

mycodeschool’s Video (Recursive Approach) :

<https://www.youtube.com/watch?v=KYH83T4q6Vs>

GFG Editorial :

<https://www.geeksforgeeks.org/reverse-a-linked-list/>

#### (2) Find middle of LinkedList

**Question** :

<https://leetcode.com/problems/middle-of-the-linked-list/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=sGdwSH8RK-o>

CodeWhoop’s Video :

<https://www.youtube.com/watch?v=M497FJW9mPk>

GFG Editorial :

<https://www.geeksforgeeks.org/write-a-c-function-to-print-the-middle-of-the-linked-list/>

AfterAcademy Editorial :

<https://afteracademy.com/blog/middle-of-the-linked-list>

#### (3) Merge two sorted Linked List

**Question** :

<https://leetcode.com/problems/merge-two-sorted-lists/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=Xb4slcp1U38>

Back To Back SWE’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=GfRQvf7MB3k>

Terrible Whiteboard’s Video (Iterative Approach with Code) :

<https://www.youtube.com/watch?v=orCMI6WjoIw>

Fisher Coder’s Video (Recursive Approach) :

<https://www.youtube.com/watch?v=bdWOmYL5d1g>

GFG Editorial :

<https://www.geeksforgeeks.org/merge-two-sorted-linked-lists/>

#### (4) Remove N-th node from back of LinkedList

**Question** :

<https://leetcode.com/problems/remove-nth-node-from-end-of-list/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=Lhu3MsXZy-Q>

Giuseppe Picciano’s Video (One Pass Approach) :

<https://www.youtube.com/watch?v=_-V4jJB3t9Q>

Fisher Coder’s Video (One Pass Approach) :

<https://www.youtube.com/watch?v=Kka8VgyFZfc>

LeetCode Article (Two Pass Approach & One Pass Approach) :

<https://leetcode.com/problems/remove-nth-node-from-end-of-list/solution/>

#### (5) Delete a given Node when a node is given. (0(1) Explanation)

**Question** :

[https://leetcode.com/problems/delete-node-in-a-linked-list/](https://leetcode.com/problems/delete-node-in-a-linked-list)

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=icnp4FJdZ_c>

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=3XGaTq-bRiU>

CodeWhoop’s Video :

<https://www.youtube.com/watch?v=owMmoSdqIfI>

LeetCode Article :

<https://leetcode.com/problems/delete-node-in-a-linked-list/solution/>

#### (6) Add two numbers as LinkedList

**Question** :

<https://leetcode.com/problems/add-two-numbers/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=LBVsXSMOIk4>

Terrible Whiteboard’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=1Spw7DEtB14>

Suboptimal Engineer’s Video (Recursive Approach) :

<https://www.youtube.com/watch?v=G6X7Fn2IDPE>

## Practice (Week 5)

#### Q1. Reverse Linked List II

**Problem** :

<https://leetcode.com/problems/reverse-linked-list-ii/>

**Solutions** :

leetuition’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=wk8-_M-2fzI>

Jyotinder Singh’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=BE0hruM5O5U>

Amell Peralta’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=LnnJTODA77I>

LeetCode Article (Recursive + Iterative Approach) :

<https://leetcode.com/problems/reverse-linked-list-ii/solution/>

Medium Article (Recursive Approach):

<https://medium.com/journey-to-becoming-an-algoat/reverse-a-linked-list-ii-part-2-ed0b0c376761>

Algorithms Casts’s Video (Iterative + Recursive Approach) :

<https://www.youtube.com/watch?v=sGNJidRPlUM>

#### Q2. Partition List

**Problem** :

<https://leetcode.com/problems/partition-list/>

**Solutions** :

LeetCode Article :

<https://leetcode.com/problems/partition-list/solution/>

Keep On Coding’s Video :

<https://www.youtube.com/watch?v=vsPsU8DCfGg>

#### Q3. Merge k Sorted Lists

**Problem** :

<https://leetcode.com/problems/merge-k-sorted-lists/>

**Solutions** :

Back To Back SWE’s Video (Min Heap Algorithm Approach) :

<https://www.youtube.com/watch?v=ptYUCjfNhJY>

Michael Muinos’s Video (Divide and Conquer Approach) :

<https://www.youtube.com/watch?v=BBt9FB5Yt0M>

Jyotinder Singh’s Video (Min Heap Algorithm Approach) :

<https://www.youtube.com/watch?v=OzWCsfl60sM>

Amell Peralta’s Video (Priority Queue Min Heap Code Walkthrough) :

<https://www.youtube.com/watch?v=tDn9O7UQ4E8>

Anwar Mamat’s Video (3 Approach) :

<https://www.youtube.com/watch?v=hqgpMUhwVOQ>

Techie Codes’s Video (Using Min Heap Data Structure) :

<https://www.youtube.com/watch?v=8VpsIL-cvPE>

GFG Editorial (2 Approach) :

<https://www.geeksforgeeks.org/merge-k-sorted-linked-lists/>

GFG Editorial (1 Approach) :

<https://www.geeksforgeeks.org/merge-k-sorted-linked-lists-set-2-using-min-heap/>

LeetCode Article (5 Approach) :

<https://leetcode.com/problems/merge-k-sorted-lists/solution/>

#### Q4. Add Two Numbers II

**Problem** :

<https://leetcode.com/problems/add-two-numbers-ii/>

**Solutions** :  
 Francesco Manicardi’s Video (Short & Clean Code):  
 <https://www.youtube.com/watch?v=z0B5u_HUd2Q>

GFG Editorial (Using Recursion) :

<https://www.geeksforgeeks.org/sum-of-two-linked-lists/>

Amell Peralta’s Video (Using Stack) :

<https://www.youtube.com/watch?v=aLxAUxCbMLk>

#### Q5. Odd Even Linked List

**Problem** :

<https://leetcode.com/problems/odd-even-linked-list/>

**Solutions** :

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=ie1rKf7bpHw>

LeetCode Article :

<https://leetcode.com/problems/odd-even-linked-list/solution/>

AfterAcademy Editorial :

<https://afteracademy.com/blog/odd-even-linked-list>

GFG Editorial :

<https://www.geeksforgeeks.org/rearrange-a-linked-list-such-that-all-even-and-odd-positioned-nodes-are-together/>

#### Q6. Remove Linked List Elements

**Problem** :

<https://leetcode.com/problems/remove-linked-list-elements/>

**Solutions** :

Terrible Whiteboard’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=nVpgHAZdhQQ>

Medium Article (Iterative + Recursive Approach) :

<https://medium.com/@jimdaosui/remove-linked-list-elements-6ec6b7560327>

GFG Editorial (Iterative Approach) :

<https://www.geeksforgeeks.org/delete-occurrences-given-key-linked-list/>

#### Q7. Remove Duplicates from Sorted List

**Problem** :

<https://leetcode.com/problems/remove-duplicates-from-sorted-list/>

**Solutions** :

Dream FAANG’s Video (**Recursive** Approach) :

<https://www.youtube.com/watch?v=aCAZN1x1q6w>

Amell Peralta’s Video (**Recursive** Approach) :

<https://www.youtube.com/watch?v=TsdAEkB76_0>

GFG Editorial (**Recursive** Approach) :

<https://www.geeksforgeeks.org/remove-duplicates-sorted-linked-list-using-recursion/>

Terrible Whiteboard’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=gfgJjrkR-W4>

Helper Func’s Video (Iterative Approach for more than 2 duplicates) :

<https://www.youtube.com/watch?v=_pRN6SzRYLU>

Amell Peralta’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=qbsuumkSia0>

# Week 6 - Advanced Linked List

## Editorial

#### (1) Find intersection point of Y LinkedList

**Question** :

<https://leetcode.com/problems/intersection-of-two-linked-lists/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=u4FWXfgS8jw>

Programmer Mitch’s Video (Optimal Approach) :

[https://www.youtube.com/watch?v=gaMeDay XbM](https://www.youtube.com/watch?v=gMeDsyEXnbM)

Medium Article (3 Approaches) :  
 <https://medium.com/swlh/intersection-of-two-linked-lists-a920fe2ec7c2>

Terrible Whiteboard’s Video (Optimal Approach) :

<https://www.youtube.com/watch?v=c7dOI-hDa2Q>

GFG Editorial (7 Approaches):

[https://www.gfg.org/write-a-function-to-get-the-intersection-point-of-two-linked-lists/](https://www.geeksforgeeks.org/write-a-function-to-get-the-intersection-point-of-two-linked-lists/)

mycodeschool’s Video (3 Approaches with Complexity Analysis) :

<https://www.youtube.com/watch?v=gE0GopCq378>

Vivekanand Khyade - Algorithm Every Day’s Video (Using Stack) :

<https://www.youtube.com/watch?v=ycIMmSmkQbo>

#### (2) Detect a cycle in Linked List

**Question** :

**Question A** (Linked List Cycle):

<https://leetcode.com/problems/linked-list-cycle/>

**Question B** (Linked List Cycle II):

<https://leetcode.com/problems/linked-list-cycle-ii/>

**Question C** (Detect and Remove Loop in a Linked List):

<https://afteracademy.com/problems/detect-and-remove-loop-in-a-linked-list>

**Solutions** :

**Solution A** :

take U forward’s Video :

<https://www.youtube.com/watch?v=354J83hX7RI>

Terrible Whiteboard’s Video:

<https://www.youtube.com/watch?v=sMqEwkpvJvQ>

codecram’s Video (Also explains No-Cycle Detection):

<https://www.youtube.com/watch?v=0Ih3MOgGAY8>

HackerRank’s Video:

<https://www.youtube.com/watch?v=MFOAbpfrJ8g>

CodeWhoop’s Video:  
 <https://www.youtube.com/watch?v=-V9BbUt8SbI>

Byte By Byte’s Video (2 Approaches):

<https://www.youtube.com/watch?v=dvOilHNRzZs>

Abdul Bari Data Structure’s Video:  
 <https://www.youtube.com/watch?v=C200KC7iXy4>

GFG Editorial (4 Approaches):

<https://www.geeksforgeeks.org/detect-loop-in-a-linked-list/>

**Solution B** :

take U forward’s Video :

<https://www.youtube.com/watch?v=QfbOhn0WZ88>

Evergreen's coderzone’s Video (Time - O(n) & Space - O(1)) :

<https://www.youtube.com/watch?v=iZVBVCpmugI>

Evergreen's coderzone’s Video (Time - O(n) & Space - O(n)) :

<https://www.youtube.com/watch?v=pTRQyV0EEOA>

Lets Algo together’s Video :

<https://www.youtube.com/watch?v=Qq-vnKmzJR0>

GFG Editorial (3 Approaches):   
 <https://www.geeksforgeeks.org/find-first-node-of-loop-in-a-linked-list/>

Medium Article :

<https://medium.com/@rajwar67/explanation-on-finding-the-starting-node-of-a-loop-in-linked-list-74c2f3d1590b>

**Solution C** :

UBlog’s Video (**Reason why slow-fast pointer approach works**):

<https://www.youtube.com/watch?v=8NBkGnY5iJ8>

CodesDope Blog :

<https://www.codesdope.com/blog/article/detect-and-remove-loop-in-a-linked-list/>

GFG Editorial (4 Approaches):   
 <https://www.geeksforgeeks.org/detect-and-remove-loop-in-a-linked-list/>

#### (3) Reverse a LinkedList in groups of size k. (Reverse Nodes in k-Group)

**Question** :

<https://leetcode.com/problems/reverse-nodes-in-k-group/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=Of0HPkk3JgI>

Coding with Conner’s Video :

<https://www.youtube.com/watch?v=jTWEmztptCQ>

GFG Editorial (Using Recursion) :

<https://www.geeksforgeeks.org/reverse-a-list-in-groups-of-given-size/>

GFG Editorial (Using Stack) :

<https://www.geeksforgeeks.org/reverse-linked-list-groups-given-size-set-2/>

#### (4) Check if a LinkedList is palindrome or not.

**Question** :

<https://leetcode.com/problems/palindrome-linked-list/>

**Solutions** :  
 take U forward’s Video :

<https://www.youtube.com/watch?v=-DtNInqFUXs>

GFG Editorial (3 Approaches) :

<https://www.geeksforgeeks.org/function-to-check-if-a-singly-linked-list-is-palindrome/>

Fisher Coder’s Video (Time Complexity - O(n) & Space Complexity - O(1)) :

<https://www.youtube.com/watch?v=bOGh_3MTrdE>

Amell Peralta’s Video (Time Complexity - O(n) & Space Complexity - O(n)) :

<https://www.youtube.com/watch?v=BTzWJUIoAIQ>

#### (5) Find the starting point of the Loop of LinkedList

**Question** :

<https://leetcode.com/problems/linked-list-cycle-ii/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=QfbOhn0WZ88>

Evergreen's coderzone’s Video (Time - O(n) & Space - O(1)) :

<https://www.youtube.com/watch?v=iZVBVCpmugI>

Evergreen's coderzone’s Video (Time - O(n) & Space - O(n)) :

<https://www.youtube.com/watch?v=pTRQyV0EEOA>

Lets Algo together’s Video :

<https://www.youtube.com/watch?v=Qq-vnKmzJR0>

GFG Editorial (3 Approaches):   
 <https://www.geeksforgeeks.org/find-first-node-of-loop-in-a-linked-list/>

Medium Article :

<https://medium.com/@rajwar67/explanation-on-finding-the-starting-node-of-a-loop-in-linked-list-74c2f3d1590b>

#### (6) Flattening of a LinkedList

**Question** :

**Question A** (Flattening a Singly Linked List):

<https://practice.geeksforgeeks.org/problems/flattening-a-linked-list/1>

**Question B** (Flatten a Multilevel Doubly Linked List):

<https://leetcode.com/problems/flatten-a-multilevel-doubly-linked-list/>

**Solutions** :

**Solution A** :

take U forward’s Video :

<https://www.youtube.com/watch?v=ysytSSXpAI0>

GFG Editorial (Merge Singly Lists via Recursion) :

<https://www.geeksforgeeks.org/flattening-a-linked-list/>

GFG Editorial (Using Heap) :

<https://www.geeksforgeeks.org/flattening-a-linked-list-set-2/>

**Solution B** :

Algorithms Made Easy’s Video (Recursive + Iterative Approach) :

<https://www.youtube.com/watch?v=A7leG9StaJ8>

Akshay Goyal’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=pjWqjqGDOlw>

Tech With Paul’s Video (Iterative Approach) :

<https://www.youtube.com/watch?v=ugBx_T1RHuc>

Maged Helmy’s Video (Recursive Approach) :

<https://www.youtube.com/watch?v=QWoX2-s8KLE>

#### (7) Rotate a LinkedList

**Question** :

**Question A** (Clockwise Rotation):

<https://leetcode.com/problems/rotate-list/>

**Question B** (Anti-Clockwise Rotation):

<https://practice.geeksforgeeks.org/problems/rotate-a-linked-list/1>

**Solutions** :

**Solution A** :

take U forward’s Video :

<https://www.youtube.com/watch?v=9VPm6nEbVPA>

Jyotinder Singh’s Video:

<https://www.youtube.com/watch?v=A7nNLqFqmn8>

Amell Peralta’s Video:

<https://www.youtube.com/watch?v=VX5Fz9z4-CE>

GFG Editorial:

<https://www.geeksforgeeks.org/clockwise-rotation-of-linked-list/>

**Solution B** :

BORNTOCODE’s Video:

<https://www.youtube.com/watch?v=NC2hGWsyeLo>

GFG Editorial:

<https://www.geeksforgeeks.org/rotate-a-linked-list/>

## Practice (Week 6)

#### Q1. Linked List in Zig-Zag fashion

**Problem** :

<https://practice.geeksforgeeks.org/problems/linked-list-in-zig-zag-fashion/1>

**Solutions** :

O(n) Logic [Part 1]:

<https://www.geeksforgeeks.org/converting-an-array-of-integers-into-zig-zag-fashion/>

GFG Editorial (Implementation of Logic) [Part 2] :

<https://www.geeksforgeeks.org/linked-list-in-zig-zag-fashion/>

#### Q2. Reorder List

**Problem** :

<https://leetcode.com/problems/reorder-list/>

**Solutions** :

Nick White’s Video :

<https://www.youtube.com/watch?v=xRYPjDMSUFw>

Algorithms Made Easy’s Video :

<https://www.youtube.com/watch?v=rEYm4yLHSHQ>

#### Q3. Remove Zero Sum Consecutive Nodes from Linked List

**Problem** :

<https://leetcode.com/problems/remove-zero-sum-consecutive-nodes-from-linked-list/>

**Solutions** :

leetuition’s Video :

<https://www.youtube.com/watch?v=tss5biw6ctI>

LeetCode Discussion Comment :

<https://leetcode.com/problems/remove-zero-sum-consecutive-nodes-from-linked-list/discuss/366350/C%2B%2B-O(n)-(explained-with-pictures)>

GFG Editorial :

<https://www.geeksforgeeks.org/delete-continuous-nodes-with-sum-k-from-a-given-linked-list/>

#### Q4. Sort List

**Problem** :

<https://leetcode.com/problems/sort-list/>

**Solutions** :

Amell Peralta’s Video (Merge Sort via Recursion) :

<https://www.youtube.com/watch?v=vH-o_6rwCEE>

GFG Editorial (Merge Sort via Recursion) :

<https://www.geeksforgeeks.org/merge-sort-for-linked-list/>

GFG Editorial (Iterative Merge Sort) :

<https://www.geeksforgeeks.org/iterative-merge-sort-for-linked-list/>

GFG Editorial (QuickSort via Recursion) :

<https://www.geeksforgeeks.org/quicksort-on-singly-linked-list/>

Why is QuickSort preferred for Arrays and Merge Sort for Linked Lists? :

<https://www.geeksforgeeks.org/why-quick-sort-preferred-for-arrays-and-merge-sort-for-linked-lists/>

AfterAcademy Editorial (Merge Sort) :

<https://afteracademy.com/blog/sort-list-merge-sort>

#### Q5. Insertion Sort List

**Problem** :

<https://leetcode.com/problems/insertion-sort-list/>

**Solutions** :

Sahil Arora’s Video :

<https://www.youtube.com/watch?v=vyeHALAClic>

AfterAcademy Editorial :

<https://afteracademy.com/blog/sort-a-linked-list-using-insertion-sort>

Quinston Pimenta’s Video :

<https://www.youtube.com/watch?v=_5_v2E0OWVs>

#### Q6. Remove Duplicates from Sorted List II

**Problem** :

<https://leetcode.com/problems/remove-duplicates-from-sorted-list-ii/>

**Solutions** :

Ren Zhang’s Video (Iterative + Recursive Approach) :

<https://www.youtube.com/watch?v=ODwu-L7nH9A>

alGOds’s Video (Recursive Approach) :

<https://www.youtube.com/watch?v=j7W70djR5ow>

GFG Editorial (Iterative Approach) :

<https://www.geeksforgeeks.org/remove-occurrences-duplicates-sorted-linked-list/>

#### Q7. QuickSort on Doubly Linked List [SEARCH MORE EXPLANATIONS]

**Problem** :

<https://practice.geeksforgeeks.org/problems/quicksort-on-doubly-linked-list/1>

**Solutions** :

CodesDope Blog :

<https://www.codesdope.com/blog/article/quicksort-on-doubly-linked-list/>

GFG Editorial :

<https://www.geeksforgeeks.org/quicksort-for-linked-list/>

# Week 7 - Two Pointer

## Editorial

#### (1) Clone a Linked List with random and next pointer.

**Question** :

<https://leetcode.com/problems/copy-list-with-random-pointer/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=VNf6VynfpdM>

Medium Article (By Divya Godayal):

<https://medium.com/spotthedifference/deep-copy-a-linked-list-b90d8376223f>

Vivekanand Khyade’s Video (Time - O(n) & Space - O(n)) :

<https://www.youtube.com/watch?v=EHpS2TBfWQg>

Maged Helmy’s Video (Time - O(n) & Space - O(1)) :

<https://www.youtube.com/watch?v=L2wOEvjCjwA>

Rachit Jain’s Video (2 Approaches):

<https://www.youtube.com/watch?v=xbpUHSKoALg>

Back To Back SWE’s Video (2 Approaches):

<https://www.youtube.com/watch?v=OvpKeraoxW0>

Ren Zhang’s Video (2 Approaches):

<https://www.youtube.com/watch?v=DEKr0efEGTM>

GFG Editorial (Time - O(n) & Space - O(1)) :

<https://www.geeksforgeeks.org/clone-linked-list-next-random-pointer-o1-space/>

GFG Editorial (Time - O(n) & Space - O(n)) :

<https://www.geeksforgeeks.org/clone-linked-list-next-arbit-pointer-set-2/>

#### (2) 3 sum

**Question** :

<https://leetcode.com/problems/3sum/>

OR

<https://www.interviewbit.com/problems/3-sum-zero/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=onLoX6Nhvmg>

CoderSnacks’s Video (**Intuitive** Explanation) :

<https://www.youtube.com/watch?v=jXZDUdHRbhY>

Krishna Teaches’s Video (Visual Code Walkthrough) :

<https://www.youtube.com/watch?v=QLec4VB4Ol0>

Khushboo Goel’s Video (Visual Explanation):

<https://www.youtube.com/watch?v=fj1g_-BwCMk>

Giuseppe Picciano’s Video (Visual Explanation):

<https://www.youtube.com/watch?v=erEHQO0xljc>

GFG Editorial :

<https://www.geeksforgeeks.org/find-triplets-array-whose-sum-equal-zero/>

#### (3) Trapping rainwater

**Question** :

<https://leetcode.com/problems/trapping-rain-water/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=m18Hntz4go8>

Knapsak’s Video (3 Approaches):

<https://www.youtube.com/watch?v=RV7jsfvJ33U>

LogicHeap’s Video (**Using Stack**):

<https://www.youtube.com/watch?v=lhzrp3Nbj-w>

GFG Editorial (4 Approaches):

<https://www.geeksforgeeks.org/trapping-rain-water/>

Algorithms Made Easy’s Video (4 Approaches):

<https://www.youtube.com/watch?v=EdR3V5DBgyo>

ForAllEpsilon’s Video (Part **1/3**):

<https://www.youtube.com/watch?v=HmBbcDiJapY>

ForAllEpsilon’s Video (Part **2/3**) [Time O(n) | Space O(n)]:

<https://www.youtube.com/watch?v=VZpJxINSvfs>

ForAllEpsilon’s Video (Part **3/3**) [Time O(n) | Space O(1)]:

<https://www.youtube.com/watch?v=XqTBrQYYUcc>

Terrible Whiteboard’s Video (DP Approach):

<https://www.youtube.com/watch?v=fTD6Se3ZtEo>

Time Complexity Infinity’s Video (DP Approach):

<https://www.youtube.com/watch?v=zdDeV5v_iUE>

thecodingworld’s Video (Visual DP Approach):

<https://www.youtube.com/watch?v=W-lWBEVE7Uc>

LeetCode Article (4 Approaches):

<https://leetcode.com/problems/trapping-rain-water/solution/>

#### (4) Remove Duplicate from Sorted array

**Question** :

<https://leetcode.com/problems/remove-duplicates-from-sorted-array/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=Fm_p9lJ4Z_8>

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=rlfsnRY0S9k>

Jesse Dietrichson’s Video :

<https://www.youtube.com/watch?v=4ZIJ6fGB1e0>

CodesDope Blog :

<https://www.codesdope.com/blog/article/remove-duplicate-elements-from-sorted-array/>

labuladong’s Article :

<https://labuladong.gitbook.io/algo-en/iv.-high-frequency-interview-problem/removeduplicatesfromsortedarray>

#### (5) Max continuous number of 1’s

**Question** :

<https://leetcode.com/problems/max-consecutive-ones/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=Mo33MjjMlyA>

Nick White’s Video (Using Global & Local Maxima):

<https://www.youtube.com/watch?v=PLa4tYQhqoU>

Michael Geng’s Video (Using Two Pointer Approach):

<https://www.youtube.com/watch?v=4q0UyW6XB60>

Coding Brunette’s Video (Code Walkthrough via LeetCode Debugger):

<https://www.youtube.com/watch?v=60TTTZWH9uY>

GFG Editorial :

<https://www.geeksforgeeks.org/maximum-consecutive-ones-or-zeros-in-a-binary-array/>

The Brown Box’s Video (**Only** Visual Walkthrough):

<https://www.youtube.com/watch?v=MxAybn96rlQ>

Coding Kevin BKH’s Video (**Only** Visual Walkthrough):

<https://www.youtube.com/watch?v=_l5aRR04Yuc>

## Practice (Week 7)

#### Q1. Subarray Product Less Than K

**Problem** :

<https://leetcode.com/problems/subarray-product-less-than-k/>

**Solutions** :

Anish Malla’s Video :

<https://www.youtube.com/watch?v=4775IgUKfww>

GFG Editorial :

<https://www.geeksforgeeks.org/number-subarrays-product-less-k/>

LeetCode Comment (Explaining how “right - left + 1” works):  
<https://leetcode.com/problems/subarray-product-less-than-k/solution/717864>

#### Q2. Subarrays with K Different Integers

**Problem** :

<https://leetcode.com/problems/subarrays-with-k-different-integers/>

**Solutions** :

GFG Editorial :

<https://www.geeksforgeeks.org/count-of-subarrays-having-exactly-k-distinct-elements/>

GFG Comment (Explaining how “right -left + 1” works) :  
 <https://pasteboard.co/JtLmnnk.png>

#### Q3. Subarrays with Sum K [SEARCH FOR MORE EXPLANATIONS]

**Problem** :

**Problem A** (Subarray Sum Equals K): [SEARCH FOR MORE EXPLANATIONS]

<https://leetcode.com/problems/subarray-sum-equals-k/>

**Problem B** (Binary Subarrays With Sum): [SEARCH FOR MORE EXPLANATIONS]

<https://leetcode.com/problems/binary-subarrays-with-sum/>

**Solutions** :

**Solution A** :

Medium Article (By Divya Godayal) :

<https://medium.com/spotthedifference/number-of-subarrays-having-sum-exactly-equal-to-k-f943ea367bfa>

Knapsak’s Video (Visual + Code):

<https://www.youtube.com/watch?v=6poxiip7sBY>

daose’s Video :

<https://www.youtube.com/watch?v=D5_AudnzHTI>

Jyotinder Singh’s Video (Using HashMap):

<https://www.youtube.com/watch?v=UhE-Srvawo8>

Akshay Goyal’s Video (Using HashMap):

<https://www.youtube.com/watch?v=N6EzbSxD6Bg>

LeetCode Article :

<https://leetcode.com/problems/subarray-sum-equals-k/solution/>

GFG Editorial :

<https://www.geeksforgeeks.org/number-subarrays-sum-exactly-equal-k/>

**Solution B** :

Medium Article (By Divya Godayal) :

<https://medium.com/spotthedifference/number-of-subarrays-having-sum-exactly-equal-to-k-f943ea367bfa>

Algorithms Casts’s Video (Map Approach + Two Pointer Approach) :

<https://www.youtube.com/watch?v=56qUe5E0QNc>

Francesco Manicardi’s Video (Two Pointer Approach) :

<https://www.youtube.com/watch?v=riE1wrGKxN4>

#### Q4. Smallest Subarray with Sum at least K [SEARCH FOR MORE EXPLANATIONS]

**Problem** :

**Problem A** (Minimum Size Subarray Sum):

<https://leetcode.com/problems/minimum-size-subarray-sum/>

**Problem B** (Shortest Subarray with Sum at Least K):

<https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k/>

**Solutions** :

**Solution A** :

Medium Article (2 Approaches):

<https://medium.com/@lenchen/leetcode-209-minimum-size-subarray-sum-ab92c2de4e94>

LeetCode Comment (With Visual Walkthrough):

<https://leetcode.com/problems/minimum-size-subarray-sum/discuss/277445/Python-Sliding-Window-Approach-(with-comments)>

LeetCode Comment (By lee215):

<https://leetcode.com/problems/minimum-size-subarray-sum/discuss/433123>

**Solution B** :

LeetCode Comment (Using Deque):

1) Intuition :

<https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k/discuss/189039/Detailed-intuition-behind-Deque-solution>

2) Code (By lee215):

<https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k/discuss/143726/C%2B%2BJavaPython-O(N)-Using-Deque>

GFG Editorial (Using Binary Search + Prefix Sum + HashMap) [**O(N\*LogN)**]:

<https://www.geeksforgeeks.org/smallest-subarray-from-a-given-array-with-sum-greater-than-or-equal-to-k/>

LeetCode Comment (Intuitive Heap Solution):

<https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k/discuss/572877/C%2B%2B-Intuition-behind-the-heap-solution>

LeetCode Comment (Intuitive Greedy Approach) [**FAILS FEW EDGE CASES**]:

<https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k/discuss/152764/Beats-99.9(14-ms)-solutions-More-intutive-than-queue-based-solution>.

#### Q5. Remove Duplicates from Sorted Array II

**Problem** :

[h](https://leetcode.com/problems/subarray-product-less-than-k/)<https://leetcode.com/problems/remove-duplicates-from-sorted-array-ii/>

**Solutions** :

Amell Peralta’s Video :

<https://www.youtube.com/watch?v=-jHoA0e-IJ0>

dev.to Article (Generalized Approach):  
<https://dev.to/varunu28/a-leetcode-a-day-remove-duplicates-from-sorted-array-ii-2b0>

Francesco Manicardi’s Video (Extremely Short Code):

<https://www.youtube.com/watch?v=OTh-l-COvt4>

#### Q6. Move Zeroes

**Problem** :

<https://leetcode.com/problems/move-zeroes/>

**Solutions** :

Terrible Whiteboard’s Video (One Pass Solution):  
 <https://www.youtube.com/watch?v=0rPuILjoVsg>

shirin setayesh’s Video (Two Pass Solution):

<https://www.youtube.com/watch?v=vs4rvPT1_mM>

On The Spot STEM’s Video (**Intuition** for One Pass Solution):

<https://www.youtube.com/watch?v=S6h1a1knsoQ>

Brennan Fradelis’s Video (One Pass Solution + Python Code):

<https://www.youtube.com/watch?v=QOth4_VJkJY>

studytonight.com Editorial (Two Pass Solution):

<https://www.studytonight.com/post/leetcode-solution-move-zeroes-problem>

cherryljr’s Article (2 Approaches)

<https://github.com/cherryljr/LeetCode/blob/master/Move%20Zeroes.java>

AfterAcademy Editorial (3 Approaches):

<https://afteracademy.com/blog/move-all-the-zeroes-to-the-end>

Dev.to Article (3 Approaches):

<https://dev.to/13point5/leetcode-challenge-move-zeroes-512p>

LeetCode Article (3 Approaches) :

<https://leetcode.com/problems/move-zeroes/solution/>

#### Q7. 3Sum Closest

**Problem** :

<https://leetcode.com/problems/3sum-closest/>

OR

<https://www.interviewbit.com/problems/3-sum/>

**Solutions** :  
 Krishna Teaches’s Video :  
 <https://www.youtube.com/watch?v=hHMz-9sXE1g>

Leslie Tang’s Video (Visual Logic Walkthrough):

<https://www.youtube.com/watch?v=cJOkAOgfRr8>

LeetCode Article (2 Approaches) :

<https://leetcode.com/problems/3sum-closest/solution/>

GFG Editorial (1 Approach) :

<https://www.geeksforgeeks.org/find-a-triplet-in-an-array-whose-sum-is-closest-to-a-given-number/>

Han Jiang’s Medium Article (Part 1 | Brute Force Approach):

<https://medium.com/@hanjiang_54259/leetcode-16-3sum-closest-in-javascript-part-1-1493777c11b9>

Han Jiang’s Medium Article (Part 2 | Optimised Approach):

<https://medium.com/@hanjiang_54259/leetcode-16-3sum-closest-in-javascript-part-2-ffa883a3cd2d>

#### Q8. Container With Most Water

**Problem** :

<https://leetcode.com/problems/container-with-most-water/>

**Solutions** :

Inside code’s Video :

<https://www.youtube.com/watch?v=I7fFgU6n4x8>

Michael Muinos’s Video :

<https://www.youtube.com/watch?v=JMmKtYH5VOE>

Time Complexity Infinity’s Video :

<https://www.youtube.com/watch?v=k5fbSqb9sCI>

Coding with Conner’s Video :

<https://www.youtube.com/watch?v=x6ZZ3JmgRKE>

Krishna Teaches’s Video (Visual Code Walkthrough) :

<https://www.youtube.com/watch?v=aJOvDxY6AQw>

Algorithms Casts’s Video (**Intuitive** Explanation):

<https://www.youtube.com/watch?v=O3BmSJWY6nU>

Weili Yang’s Video (**Intuitive** Explanation):

<https://www.youtube.com/watch?v=XASDpSFty74>

Joy Liu - Computer Psyc’s Video (**Proof**):

<https://www.youtube.com/watch?v=cPwXGcZQ1mA>

GFG Editorial :

<https://www.geeksforgeeks.org/container-with-most-water/>

LeetCode Article (with links to proof) :

<https://leetcode.com/problems/container-with-most-water/solution/>

#### Q9. Minimum Window Substring [SEARCH FOR MORE EXPLANATION]

**Problem** :

<https://leetcode.com/problems/minimum-window-substring/>

**Solutions** :

Back To Back SWE’s Video :

<https://www.youtube.com/watch?v=eS6PZLjoaq8>

# Week 8 - Greedy

## Editorial

#### (1) N meeting in one room

**Question** :

**Question A** (N meetings in one room):

<https://practice.geeksforgeeks.org/problems/n-meetings-in-one-room/0>

**Question B** (Meeting Rooms II):

<https://www.interviewbit.com/problems/meeting-rooms/>   
 OR  
<https://www.lintcode.com/problem/meeting-rooms-ii/description>

**Question C** (Meeting Rooms):

<https://www.lintcode.com/problem/meeting-rooms/description>

**Solutions** :

**Solution A** :

take U forward’s Video :

<https://www.youtube.com/watch?v=II6ziNnub1Q>

GFG Editorial :

<https://www.geeksforgeeks.org/find-maximum-meetings-in-one-room/>

**Solution B** :  
 Ankur Agrawal’s Video (**Intuitive** Explanation):

<https://www.youtube.com/watch?v=DFEf8_fjb_0>

Coding With Jaz’s Video (Using Heap):

<https://www.youtube.com/watch?v=9ZsUM1ed05c>

Coding With Jaz’s Video (with **Follow-Up** Questions)

<https://www.youtube.com/watch?v=wO4x9NnCOfU>

Amell Peralta’s Video:

<https://www.youtube.com/watch?v=RBlcUlUkDCU>

Phani Thaticharla’s Video (**Visual Whiteboard** Explanation):  
 <https://www.youtube.com/watch?v=JLPFkZinz30>

**Solution C** :

Amell Peralta’s Video :  
 <https://www.youtube.com/watch?v=6Ygq6wXM4-I>

Coding With Jaz’s Video :

<https://www.youtube.com/watch?v=s69Hc_MszQ8>

Phani Thaticharla’s Video (**Visual Whiteboard** Explanation):  
 <https://www.youtube.com/watch?v=JLPFkZinz30>

#### (2) Minimum number of platforms required for a railway

**Question** :

<https://practice.geeksforgeeks.org/problems/minimum-platforms-1587115620/1>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=dxVcMDI7vyI>

GFG Editorial (Greedy Approach):

<https://www.geeksforgeeks.org/minimum-number-platforms-required-railwaybus-station/>

GFG Editorial (Map Based Approach):

<https://www.geeksforgeeks.org/minimum-number-platforms-required-railwaybus-station-set-2-map-based-approach/>

#### (3) Job sequencing Problem (Maximum Profit in Job Scheduling)

**Question** :

<https://practice.geeksforgeeks.org/problems/job-sequencing-problem-1587115620/1>   
 OR  
 <https://leetcode.com/problems/maximum-profit-in-job-scheduling/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=LjPx4wQaRIs>

Abdul Bari’s Video :

<https://www.youtube.com/watch?v=zPtI8q9gvX8>   
  
 GFG Editorial :   
 <https://www.geeksforgeeks.org/job-sequencing-problem/>   
  
 Medium Article :  
 <https://medium.com/@withsnowy2009/job-sequencing-with-deadlines-c3e996df0928>

#### (4) Fractional Knapsack Problem

**Question** :

<https://practice.geeksforgeeks.org/problems/fractional-knapsack-1587115620/1>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=F_DDzYnxO14>

Abdul Bari’s Video :

<https://www.youtube.com/watch?v=oTTzNMHM05I>

Yusuf Shakeel’s Video :

<https://www.youtube.com/watch?v=_08myilrxq8>

GFG Editorial :

<https://www.geeksforgeeks.org/fractional-knapsack-problem/>

AfterAcademy Editorial :

<https://afteracademy.com/blog/fractional-knapsack-problem>

#### (5) Greedy algorithm to find minimum number of coins

**Question** :

<https://www.geeksforgeeks.org/greedy-algorithm-to-find-minimum-number-of-coins/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=mVg9CfJvayM>

Progressive Coder Editorial :  
 <http://progressivecoder.com/coin-change-problem-using-greedy-algorithm/>

Codeforces Comment (**Required Condition for Greedy Approach to work**) :

<https://codeforces.com/blog/entry/60565?#comment-725434>

CodesDope Editorial :  
 <https://www.codesdope.com/course/algorithms-greedy-algorithm/>

GFG Editorial :

<https://www.geeksforgeeks.org/greedy-algorithm-to-find-minimum-number-of-coins/>

#### (6) Activity Selection [SEARCH FOR MORE EXPLANATION]

**Question** :

<https://afteracademy.com/problems/activity-selection-problem>

**Solutions** :

CodesDope Editorial :

<https://www.codesdope.com/course/algorithms-activity-selection/>

## Practice (Week 8)

#### Q1. Non-overlapping Intervals [SEARCH FOR MORE EXPLANATION]

**Problem** :

<https://leetcode.com/problems/non-overlapping-intervals/>

**Solutions** :

Algorythm’s Video :

<https://www.youtube.com/watch?v=hyQZCTfQDxo>

Anish Malla’s Video :

<https://www.youtube.com/watch?v=3oDvuHCTFmY>

#### Q2. Remove Covered Intervals [SEARCH FOR MORE EXPLANATION]

**Problem** :

<https://leetcode.com/problems/remove-covered-intervals/>

**Solutions** :

Anish Malla’s Video :

<https://www.youtube.com/watch?v=emPnw5m2nN0>

Francesco Manicardi’s Video :

<https://www.youtube.com/watch?v=eSqgqWV5D50>

#### Q3. Chef and Bitwise Product [SEARCH FOR MORE EXPLANATION]

**Problem** :

<https://www.codechef.com/problems/CHANDF>

**Solutions** :

Rachit Jain’s Video :

<https://www.youtube.com/watch?v=-F7cHQ-gWS4>

#### Q4. Gas Station

**Problem** :

<https://leetcode.com/problems/gas-station/>

**Solutions** :

Knapsak’s Video :

<https://www.youtube.com/watch?v=wDgKaNrSOEI>

Joy Liu - Computer Psyc’s Video (With Proof) :

<https://www.youtube.com/watch?v=rf66wlb9aNQ>

Anish Malla’s Video :

<https://www.youtube.com/watch?v=lWl-HbIC70g>

Applied AI Course’s Video :

<https://www.youtube.com/watch?v=nTKdYm_5-ZY>

Medium Article :

<https://medium.com/@rohitsharmacr/gas-station-leetcode-134-d9693244d18c>

GFG Editorial :

<https://www.geeksforgeeks.org/find-a-tour-that-visits-all-stations/>

GFG Editorial (DP Approach):

<https://www.geeksforgeeks.org/number-of-circular-tours-that-visit-all-petrol-pumps/>

# Week 9 - [Recursion](https://vibgyor.hashnode.dev/beginners-guide-to-recursion)

## Editorial

#### (1) Combination sum-1

**Question** :

**Question A** (Combination Sum):

<https://leetcode.com/problems/combination-sum/>

OR  
<https://practice.geeksforgeeks.org/problems/combination-sum-1587115620/1>

**Question B** (Combination Sum IV):

<https://leetcode.com/problems/combination-sum-iv/>

**Solutions** :

**Solution A** :

take U forward’s Video :

<https://www.youtube.com/watch?v=OyZFFqQtu98>

Time Complexity Infinity’s Video :

<https://www.youtube.com/watch?v=MTI2wc8s0BY>

GFG Editorial :

<https://www.geeksforgeeks.org/combinational-sum/>

AfterAcademy Editorial :  
 <https://afteracademy.com/blog/combination-sum>

**Solution B** :

Knapsak’s Video :

<https://www.youtube.com/watch?v=VPdg1gPRe04>

#### (2) Combination sum-2

**Question** :

**Question A** (Combination Sum II):

<https://leetcode.com/problems/combination-sum-ii/>

**Question B** (Combination Sum III):

<https://leetcode.com/problems/combination-sum-iii/>

**Solutions** :

**Solution A** :

take U forward’s Video :

<https://www.youtube.com/watch?v=G1fRTGRxXU8>

Kevin Naughton Jr.’s Video :

<https://www.youtube.com/watch?v=IER1ducXujU>

GFG Editorial :

<https://www.geeksforgeeks.org/all-unique-combinations-whose-sum-equals-to-k/>

**Solution B** :

Anish Malla’s Video (Backtracking):

<https://www.youtube.com/watch?v=J2hcPZRpbMk>

thecodingworld’s Video (DFS Approach):

<https://www.youtube.com/watch?v=RdqpTEiR9ss>

#### (3) Palindrome Partioning

**Question** :

<https://leetcode.com/problems/palindrome-partitioning/>

OR

<https://www.interviewbit.com/problems/palindrome-partitioning/>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=WBgsABoClE0>

LeetCode Article (Backtracking + DP):  
 <https://leetcode.com/problems/palindrome-partitioning/solution/>

NeetCode’s Video (Visual Explanation + Python Code):

<https://www.youtube.com/watch?v=3jvWodd7ht0>

Algorithms Made Easy’s Video (Java Code):

<https://www.youtube.com/watch?v=uJeS6FmbSjM>

GFG Editorial :

<https://www.geeksforgeeks.org/print-palindromic-partitions-string/>

AND

<https://www.geeksforgeeks.org/given-a-string-print-all-possible-palindromic-partition/>

#### (4) Subset Sum-1 (Subset Sums) [Search For More Explanation]

**Question** :

<https://practice.geeksforgeeks.org/problems/subset-sums2234/1>

**Solutions** :

take U forward’s Video :

<https://www.youtube.com/watch?v=rYkfBRtMJr8>

#### (5) Subset Sum-2 [Search For More Explanation]

**Question** :

**Question A** (Subsets):

<https://leetcode.com/problems/subsets/>

**Question B** (Subsets II):

<https://leetcode.com/problems/subsets-ii/>

**Solutions** :

**Solution A** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solution B** :

take U forward’s Video :

<https://www.youtube.com/watch?v=RIn3gOkbhQE>

#### (6) K-th permutation Sequence (Permutation Sequence)

**Question** :

<https://leetcode.com/problems/permutation-sequence/>

**Solutions** :

Algorithms Made Easy’s Video (Visual Logic Explanation):

<https://www.youtube.com/watch?v=T_KP7GcsHVY>

Khushboo Goel’s Video (Code Explanation):

<https://www.youtube.com/watch?v=QNCuMUApTSc>

GFG Editorial :

<https://www.geeksforgeeks.org/find-the-k-th-permutation-sequence-of-first-n-natural-numbers/>

# Week 10 - Backtracking

## Editorial

#### (1) N queens Problem

**Question** :

**Question A** (N-Queens):

<https://leetcode.com/problems/n-queens/>

OR

<https://www.interviewbit.com/problems/nqueens/>

**Question B** (N-Queens II):

<https://leetcode.com/problems/n-queens-ii/>

**Solutions** :

**Solution A** :

Back To Back SWE’s Video (WhiteBoard Explanation):

<https://www.youtube.com/watch?v=wGbuCyNpxIg>

CSBreakdown’s Video (Visual Code Walkthrough):

<https://www.youtube.com/watch?v=kX5frmc6B7c>

GeeksforGeeks’s Video (Animation):

<https://www.youtube.com/watch?v=0DeznFqrgAI>

CodesDope Editorial (with **Complexity Analysis**):

<https://www.codesdope.com/course/algorithms-backtracking/>

GFG Editorial (All possible output):  
 <https://www.geeksforgeeks.org/printing-solutions-n-queen-problem/>

Medium Article (Optimization Intuition):

<https://towardsdatascience.com/data-manipulation-with-n-queens-640d37e3c774>

**Solution B** :  
 Programming Live with Larry’s Video :

<https://www.youtube.com/watch?v=1x6DraMw35c>

Medium Article (Induition):

<https://medium.com/swlh/how-many-solutions-does-the-n-queens-problem-have-e8da5d45a34c>

#### (2) Sudoko

**Question** (Sudoku Solver):

<https://leetcode.com/problems/sudoku-solver/>

OR

<https://www.interviewbit.com/problems/sudoku/>

**Solutions** :

GeeksforGeeks’s Video (Visual Explanation) :

<https://www.youtube.com/watch?v=_vWRZiDUGHU>

Back To Back SWE’s Video (Only Theory):

<https://www.youtube.com/watch?v=JzONv5kaPJM>

happygirlzt’s Video (with Code):

<https://www.youtube.com/watch?v=wWUDo2FkdMc>

AfterAcademy Editorial :

<https://afteracademy.com/blog/sudoku-solver>

GFG Editorial :  
 <https://www.geeksforgeeks.org/sudoku-backtracking-7/>

CodesDope Blog :

<https://www.codesdope.com/blog/article/solving-sudoku-with-backtracking-c-java-and-python/>

#### (3) M coloring Problem (Graph prob)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) Rat in a Maze

**Question** :

**Question A** (Rat in a Maze Problem):

<https://practice.geeksforgeeks.org/problems/rat-in-a-maze-problem/1>

**Question B** (Rat Maze):

<https://www.codechef.com/problems/BPHC03>

**Question C** (Jumping Rat in a Maze):

<https://www.geeksforgeeks.org/rat-in-a-maze-with-multiple-steps-jump-allowed/>

**Solutions** :

**Solution A** :

GeeksforGeeks’s Video (Visual Explanation):

<https://www.youtube.com/watch?v=PwxGTHraMNg>

GFG Editorial :  
 <https://www.geeksforgeeks.org/rat-in-a-maze-backtracking-2/>

CodesDope Editorial :

<https://www.codesdope.com/blog/article/backtracking-to-solve-a-rat-in-a-maze-c-java-pytho/>

**Solution B** :

GFG Editorial :

<https://www.geeksforgeeks.org/rat-in-a-maze-problem-when-movement-in-all-possible-directions-is-allowed/>

GFG Editorial (Using **STACK**):

<https://www.geeksforgeeks.org/rat-in-a-maze-backtracking-using-stack/>

CodeChef Accepted Solution (Code with **Comments**):  
 <https://www.codechef.com/viewsolution/40295633>

**Solution C** :

GFG Editorial (Backtracking Approach):

<https://www.geeksforgeeks.org/rat-in-a-maze-with-multiple-steps-jump-allowed/>

GFG Editorial (DP Approach):

<https://www.geeksforgeeks.org/a-variation-of-rat-in-a-maze-multiple-steps-or-jumps-allowed/>

#### (5) Print all Permutations of a string/array

**Question** :

**Question A** (Permutations):

<https://leetcode.com/problems/permutations/>

OR  
<https://practice.geeksforgeeks.org/problems/permutations-of-a-given-string/0>

**Question B** (Permutations II):

<https://leetcode.com/problems/permutations-ii/>

**Solutions** :

**Solution A** :

Back To Back SWE’s Video (Logic Explanation):

<https://www.youtube.com/watch?v=GCm7m5671Ps>

computer’s Video (Logic + Code):

<https://www.youtube.com/watch?v=Nabbpl7y4Lo>

Amell Peralta’s Video (Code Explanation):

<https://www.youtube.com/watch?v=idmgLLNIC2U>

GFG Editorial :

<https://www.geeksforgeeks.org/write-a-c-program-to-print-all-permutations-of-a-given-string/>

**Solution B** :

Amell Peralta’s Video :

<https://www.youtube.com/watch?v=A3ge2mdQi4g>

Programming Live with Larry’s Video :

<https://www.youtube.com/watch?v=PRUXZW-mVLQ>

GFG Editorial :  
 <https://www.geeksforgeeks.org/distinct-permutations-string-set-2/>

#### (6) Word Break (print all ways) (Word Break II)

**Question** :

<https://leetcode.com/problems/word-break-ii/>

OR

<https://practice.geeksforgeeks.org/problems/word-break-part-2/0>

**Solutions** :

another digital nomad’s Video (Code Explanation):

<https://www.youtube.com/watch?v=uR3RElKnrkU>

Algorithms Made Easy’s Video (Visual Explanation):

<https://www.youtube.com/watch?v=PdaXY6GOL2U>

Medium Article (with Time & Space Complexity Analysis):  
 <https://salonikaurone.medium.com/leetcode-word-break-ii-explained-d41ecfbe8fc5>

GFG Editorial (Backtracking):

<https://www.geeksforgeeks.org/word-break-problem-using-backtracking/>

GFG Editorial (Optimized using **DP**):  
 <https://www.geeksforgeeks.org/word-break-problem-dp-32-set-2/>

# Week 11 - Divide and Conquer

## Editorial

#### (1) 1/N-th root of an integer (use binary search) (square root, cube root, ..)

**Question** :

**Question A** (Calculating n-th real root using binary search):

<https://practice.geeksforgeeks.org/problems/find-nth-root-of-m5843/1>

**Question B** (Floor value Kth root of a number using Recursive Binary Search):

<https://www.geeksforgeeks.org/floor-value-kth-root-of-a-number-using-recursive-binary-search/>

**Solutions** :

**Solution A** :

GFG Editorial :  
 <https://www.geeksforgeeks.org/calculating-n-th-real-root-using-binary-search/>   
  
 **Solution B** :   
 GFG Editorial :

<https://www.geeksforgeeks.org/floor-value-kth-root-of-a-number-using-recursive-binary-search/>

#### (2) Matrix Median

**Question** :

**Question A** (Median of K Sorted Arrays of Same Size):

<https://www.interviewbit.com/problems/matrix-median/>

OR

<https://afteracademy.com/problems/median-in-row-wise-sorted-matrix>

OR

<https://practice.geeksforgeeks.org/problems/median-in-a-row-wise-sorted-matrix1527/1>

**Question B** (Median of K Sorted Arrays of Different Size):

<https://www.lintcode.com/problem/median-of-k-sorted-arrays/description>

**Solutions** :

**Solution A** :

AfterAcademy Editorial :

<https://afteracademy.com/blog/median-in-a-row-wise-sorted-matrix>

GFG Editorial :  
 <https://www.geeksforgeeks.org/find-median-row-wise-sorted-matrix/>

LeetCode Discussion :

<https://leetcode.com/discuss/interview-question/904537/Amazon-or-Onsite-or-Median-of-K-sorted-arrays-of-size-N-each>   
  
**Solution B** :   
 StackOverflow Discussion :  
 <https://stackoverflow.com/questions/6182488/median-of-5-sorted-arrays>

#### (3) Find the element that appears once in sorted array, and rest element appears twice (Binary search)

**Question** (Single Element in a Sorted Array):

<https://leetcode.com/problems/single-element-in-a-sorted-array/>

OR

<https://practice.geeksforgeeks.org/problems/find-the-element-that-appears-once-in-sorted-array0624/1>

**Solutions** :

daose’s Video (Visual & Intuitive):

<https://www.youtube.com/watch?v=aFXhs190zeg>

Michael Muinos’s Video (Visual) :

<https://www.youtube.com/watch?v=4Gi8uAz666s>

Algorithms Made Easy’s Video (Visual) :

<https://www.youtube.com/watch?v=4iMnnMcEDpQ>

GFG Editorial :

<https://www.geeksforgeeks.org/find-the-element-that-appears-once-in-a-sorted-array/>

LeetCode Comment :

<https://leetcode.com/problems/single-element-in-a-sorted-array/discuss/628111/C%2B%2B-Solution-O(logn)-with-detailed-explanation>

#### (4) Search element in a sorted and rotated array/ find pivot where it is rotated

**Question** (Search in Rotated Sorted Array):

<https://leetcode.com/problems/search-in-rotated-sorted-array/>

OR

<https://www.interviewbit.com/problems/rotated-sorted-array-search/>

OR

<https://practice.geeksforgeeks.org/problems/search-in-a-rotated-array/0>

**Solutions** :

NeetCode’s Video (Visual Python O(logn)) :

<https://www.youtube.com/watch?v=U8XENwh8Oy8>

Coding Blocks’s Video (**Hindi** Visual Recursive O(logn)) :

<https://www.youtube.com/watch?v=ctW9Q6Y_Z8k>

daose’s Video (Visual Java O(logn)) :

<https://www.youtube.com/watch?v=82VgjkIdzFQ>

Time Complexity Infinity’s Video (Finding Pivot Visual O(2\*logn)) :

<https://www.youtube.com/watch?v=VoP4woobBns>

Medium Article (Detailed Explanation With Diagrams) :

<https://medium.com/spotthedifference/search-in-a-rotated-sorted-array-72c12bcb212>

LeetCode Comment (Detail Case Wise Explanation):

<https://leetcode.com/problems/search-in-rotated-sorted-array/discuss/14547/Java-Super-Clear-Solution-with-Super-Detailed-Explanation-(Took-me-2-hours-to-write)>

GFG Editorial :

<https://www.geeksforgeeks.org/search-an-element-in-a-sorted-and-pivoted-array/>

#### (5) Median of 2 sorted arrays

**Question** :

**Question A** (Median of Two Sorted Arrays of Different Size):

<https://leetcode.com/problems/median-of-two-sorted-arrays/>

OR

<https://www.interviewbit.com/problems/median-of-array/>

**Question B** (Median of Two Sorted Array of Same Size):

<https://afteracademy.com/problems/median-of-two-sorted-array-of-same-size>

**Solutions** :

**Solution A** :

Medium Article (Intuitive Explanation) :

<https://medium.com/@hazemu/finding-the-median-of-2-sorted-arrays-in-logarithmic-time-1d3f2ecbeb46>

Keerti Purswani’s Video (WhiteBoard Explanation) :

<https://www.youtube.com/watch?v=yD7wV8SyPrc>

NeetCode’s Video (Visual + Python Code) :

<https://www.youtube.com/watch?v=q6IEA26hvXc>

Krishna Teaches’s Video (Using Heap | Visual Code Walkthrough):

<https://www.youtube.com/watch?v=juzJ6SmxovA>

ForAllEpsilon’s Video (Part 1) :

<https://www.youtube.com/watch?v=CMjAo8_8JYM>

GFG Editorial (Different Size Array | 2 Approaches) :

<https://www.geeksforgeeks.org/median-of-two-sorted-arrays-of-different-sizes/>

GFG Editorial (Different Size Array | 1 Optimized Approach) :

<https://www.geeksforgeeks.org/median-two-sorted-arrays-different-sizes-ologminn-m/>

**Solution B** :

AfterAcademy Editorial (Same Size Array) :  
 <https://afteracademy.com/blog/median-of-the-two-sorted-array-of-same-size>

GFG Editorial (Same Size Array | 2 Approaches) :

<https://www.geeksforgeeks.org/median-of-two-sorted-arrays/>

#### (6) K-th element of two sorted arrays

**Question** :

<https://practice.geeksforgeeks.org/problems/k-th-element-of-two-sorted-array1317/1>

OR

<https://leetcode.com/discuss/interview-question/351782/Google-or-Phone-Screen-or-Kth-Largest-Element-of-Two-Sorted-Arrays>

**Solutions** :

AlgorithmsAndMe Editorial :

<http://www.algorithmsandme.com/find-kth-smallest-element-in-two-sorted-arrays/>

GFG Editorial :

<https://www.geeksforgeeks.org/k-th-element-two-sorted-arrays/>

Gaurav Sen’v Video (Generalize + Concept Explanation):  
 <https://www.youtube.com/watch?v=Q3JUfHpffIg>

# Week 12 - Bits / Bit Manipulation

## Editorial

#### (1) Check if a number if a power of 2 or not in O(1)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) Count total set bits

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) Divide Integers without / operator

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) Power Set (this is very important)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) Find MSB in o(1)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Find square of a number without using multiplication or division operators.

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

## Practice (Week 12)

#### Q1. Find elements that occurs once while other elements occur M times

**Problem** :

**Problem A** (Single Number II):

<https://leetcode.com/problems/single-number-ii/>   
 OR

<https://www.interviewbit.com/problems/single-number-ii/>

**Problem B** (Single Number III):

<https://leetcode.com/problems/single-number-iii/>

**Solutions** :

**Solution A** :

IDeserve’s Video (Visual) :  
 <https://www.youtube.com/watch?v=mHfvInveXDQ>

Scaler Academy’s Video :

<https://www.youtube.com/watch?v=jO7uGdvGGC4>

IDeserve Article :   
 <https://www.ideserve.co.in/learn/find-the-element-that-appears-once-in-an-array>

CareerCup Discussion :

<https://www.careercup.com/question?id=7902674>

GFG Editorial :

<https://www.geeksforgeeks.org/find-the-element-that-appears-once/>

LeetCode Comment (Summarised Explanation):

<https://leetcode.com/problems/single-number-ii/discuss/326621/All-In-One-Summary-(Single-Number-I-II-III)>

**Solution B** :   
 Sheep’s Video :  
 <https://www.youtube.com/watch?v=3TSC0nlur58>

LeetCode Comment (Summarised Explanation):

<https://leetcode.com/problems/single-number-iii/discuss/326622/All-In-One-Summary-(Single-Number-I-II-III)>   
  
LeetCode Comment (Generalization of Bitwise Operation for Single Numbers) :

<https://leetcode.com/problems/single-number-ii/discuss/43295/Detailed-explanation-and-generalization-of-the-bitwise-operation-method-for-single-numbers>

# Week 13 - Stack and Queue

## Editorial

#### (1) Implement Stack / Implement Queue [In Progress]

**Question** :

**Question A** (Implement stack using array):

<https://practice.geeksforgeeks.org/problems/implement-stack-using-array/1>

**Question B** (Implement Stack using Linked List):

<https://practice.geeksforgeeks.org/problems/implement-stack-using-linked-list/1>

**Question C** (Implement Queue using array):

<https://practice.geeksforgeeks.org/problems/implement-queue-using-array/1>

**Question D** (Implement Queue using Linked List):

<https://practice.geeksforgeeks.org/problems/implement-queue-using-linked-list/1>

**Question E** (Design Circular Queue):

<https://leetcode.com/problems/design-circular-queue/>

**Solutions** :

**Solution A** :

CodesDope Editorial :  
 <https://www.codesdope.com/blog/article/making-a-stack-using-an-array-in-c/>

**Solution B** :   
 CodesDope Editorial :  
 <https://www.codesdope.com/blog/article/making-a-stack-using-linked-list-in-c/>

**Solution C** :   
 CodesDope Editorial :  
 <https://www.codesdope.com/blog/article/making-a-queue-using-an-array-in-c/>

**Solution D** :   
 CodesDope Editorial :  
 <https://www.codesdope.com/blog/article/making-a-queue-using-linked-list-in-c/>

**Solution E** :   
 GFG Editorial (Set 1 - Array Implementation) :

<https://www.geeksforgeeks.org/circular-queue-set-1-introduction-array-implementation/>

GFG Editorial (Set 2 - Linked List Implementation) :

<https://www.geeksforgeeks.org/circular-queue-set-2-circular-linked-list-implementation/>

#### (2) BFS

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) Implement Stack using Queue [In Progress]

**Question (Implement Stack using Queues)** :

<https://leetcode.com/problems/implement-stack-using-queues/>   
 OR  
 <https://practice.geeksforgeeks.org/problems/stack-using-two-queues/1>

**Solutions** :

Terrible Whiteboard’s Video (Using **One** Queue) :

<https://www.youtube.com/watch?v=t3OvIcsaXjk&ab_channel=TerribleWhiteboard>

CodeWhoop’s Video (Using **Two** Queues) :

<https://www.youtube.com/watch?v=kKjYSBeDpFA&ab_channel=CodeWhoop>

#### (4) Implement Queue using Stack

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) Check for balanced parentheses

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Next Greater Element

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 14 - Advance Stack & Queue

## Editorial

#### (1) Next Smaller Element

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) LRU cache (vvvv. imp)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) Largest rectangle in histogram

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) Sliding Window maximum

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) Implement Min Stack [IN PROGRESS]

**Question** :

**Question A** (Min Stack):

<https://leetcode.com/problems/min-stack/>

**Question B** (Max Stack):

<https://www.lintcode.com/problem/max-stack/description>

**Solutions** :

**Solution A** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solution B** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Rotten Orange (Using BFS)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 15 - String

## Editorial

#### (1) Reverse Words in a String

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) Longest Palindrome in a string

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) Roman Number to Integer and vice versa

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) Implement ATOI/STRSTR

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) Longest Common Prefix

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Rabin Karp (Longest Duplicate Substring)

**Question** :

<https://leetcode.com/problems/longest-duplicate-substring/>

**Solutions** :

Aalekh Jain’s Video (Full Explanation):

<https://www.youtube.com/watch?v=bmEHg_yUr0k>   
   
 Stable Sort’s Video (Rolling Hash Function Tutorial):

<https://www.youtube.com/watch?v=BfUejqd07yo>

Abdul Bari’s Video (Rabin-Karp String Matching Algorithm Tutorial):

<https://www.youtube.com/watch?v=qQ8vS2btsxI>   
  
 Medium Article (Rabin-Karp Algorithm Tutorial):

<https://medium.com/@darshanrathod4400/rabin-karp-algorithm-50bf47265b29>

## Practice (Week 15)

#### Q1. Ways to split string into two palindromes (Rabin Karp Application)

**Problem** :

<https://www.geeksforgeeks.org/count-of-ways-to-split-given-string-into-two-non-empty-palindromes/>

**Solutions** :

GFG Editorial :

<https://www.geeksforgeeks.org/count-of-ways-to-split-given-string-into-two-non-empty-palindromes/>

# Week 16 - Advance String

## Editorial

#### (1) Prefix Function/Z-Function

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) KMP algo

**Question** :  
 **Question A** (Longest Prefix Suffix ):   
 <https://practice.geeksforgeeks.org/problems/longest-prefix-suffix2527/1>   
   
 **Question B** (Implement strStr()):   
 <https://leetcode.com/problems/implement-strstr/>

**Solutions** :

**Solution A** :  
 Stable Sort’s Video (Visual Animation + Code Tutorial):

<https://www.youtube.com/watch?v=EL4ZbRF587g>  
   
 Lucian Bicsi’s Video (Intuitive Explanation):  
 <https://www.youtube.com/watch?v=7V-Nt-TA3m0>

Back To Back SWE’s Video (Whiteboard Explanation):

<https://www.youtube.com/watch?v=BXCEFAzhxGY>

Abdul Bari’s Video (Whiteboard Explanation):

<https://www.youtube.com/watch?v=V5-7GzOfADQ>   
  
 Logic First’s Video (Visual + Code Explanation in Python):  
 <https://www.youtube.com/watch?v=4jY57Ehc14Y>

GFG Editorial :

<https://www.geeksforgeeks.org/kmp-algorithm-for-pattern-searching/>   
   
 Explanation of **computeLPSArray()** on GFG Editorial :

<https://leetcode.com/problems/implement-strstr/discuss/13160/detailed-explanation-on-building-up-lps-for-kmp-algorithm>   
  
**Solution B** :  
 Amell Peralta’s Video (Clean Implementation in Java):  
 <https://www.youtube.com/watch?v=TsxFvVy_5m0>

Basheer Ahmad’s Video (Hindi Explanation):

<https://www.youtube.com/watch?v=4yT8mpJxHks>   
  
 LeetCode Comment (Code in C with comments):

<https://leetcode.com/problems/implement-strstr/discuss/797907/Simple-C-solution-with-detailed-explanation.-faster-than-100>

#### (3) Minimum characters needed to be inserted in the beginning to make it palindromic.

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) Check for Anagrams [IN PROGRESS]

**Question** :

**Question A** (Valid Anagram):

<https://leetcode.com/problems/valid-anagram/>

**Question B** (Find All Anagrams in a String):

<https://leetcode.com/problems/find-all-anagrams-in-a-string/>

**Question C** (Group Anagrams):

<https://leetcode.com/problems/group-anagrams/>   
 OR  
<https://www.interviewbit.com/problems/anagrams/>

**Solutions** :

**Solution A** :

Terrible Whiteboard’s Video :

<https://www.youtube.com/watch?v=FMkueJAQ2pE>   
   
 GFG Editorial :

<https://www.geeksforgeeks.org/check-whether-two-strings-are-anagram-of-each-other/>   
 AfterAcademy’s Blog :

<https://afteracademy.com/blog/valid-anagram>

**Solution B** :

Inside code’s Video :

<https://www.youtube.com/watch?v=Y6DLFLceX7Q>

**Solution C** :

Scaler Academy’s Video :

<https://www.youtube.com/watch?v=psDooQ8Dwdo>

LeetCode Solution :  
 <https://leetcode.com/problems/group-anagrams/solution/>

#### (5) Count and Say

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Compare version numbers

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 17 - Binary Tree (Easy)

## Editorial

#### (1) Inorder Traversal (with recursion and without recursion)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) Preorder Traversal (with recursion and without recursion)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) Postorder Traversal (with recursion and without recursion)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) LeftView Of Binary Tree

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) Bottom View of Binary Tree

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Top View of Binary Tree

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 18 - Binary Tree (Medium)

## Editorial

#### (1) Level order Traversal / Level order traversal in spiral form

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) Height of a Binary Tree

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) Diameter of Binary Tree

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) Check if Binary tree is height balanced or not

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) LCA in Binary Tree

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Check if two trees are identical or not

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 19 - Binary Tree (Advance)

## Editorial

#### (1) Maximum path sum

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) Construct Binary Tree from inorder and preorder

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) Construct Binary Tree from Inorder and Postorder

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) Symmetric Binary Tree

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) Flatten Binary Tree to LinkedList

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Check if Binary Tree is mirror of itself or not

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 20 - Binary Search Tree

## Editorial

#### (1) Populate Next Right pointers of Tree

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) Search given Key in BST

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) Construct BST from given keys.

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) Check is a BT is BST or not

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) Find LCA of two nodes in BST

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Find the inorder predecessor/successor of a given Key in BST.

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 21 - Advance Binary Search Tree

## Editorial

#### (1) Floor and Ceil in a BST

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) Find K-th smallest and K-th largest element in BST (2 different Questions)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) Find a pair with a given sum in BST

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) BST iterator

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) Size of the largest BST in a Binary Tree

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Serialize and deserialize Binary Tree

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 22 - Mixed Questions

## Editorial

#### (1) Binary Tree to Double Linked List

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) Find median in a stream of running integers.

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) K-th largest element in a stream.

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) Distinct numbers in Window.

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) K-th largest element in an unsorted array.

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Flood-fill Algorithm

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

## Practice (Week 22)

#### Q1. Brick Wall

**Problem** :

<https://leetcode.com/problems/brick-wall/>

**Solutions** :

LeetCode Comment (**Intuition + Pictures**) :

<https://leetcode.com/problems/brick-wall/discuss/888577/IntuitionC++With-PicturesHashMapDetailed-ExplanationCommentsSolutionCode>

#### Q2. Minimum Area Rectangle

**Problem** :

<https://leetcode.com/problems/minimum-area-rectangle/>

**Solutions** :

LeetCode Comment (**Intuition + Pictures**) :

<https://leetcode.com/problems/minimum-area-rectangle/discuss/900264/IntuitiveWith-PicturesC++JavaExplanation>

Sonu Raj’s Video :

<https://www.youtube.com/watch?v=vvRY7bS4OMI>

#### Q3. Maximum Equal Frequency

**Problem** :

<https://leetcode.com/problems/maximum-equal-frequency/>

**Solutions** :

Dream FAANG’s Video (Visual + Code Walkthrough):  
 <https://www.youtube.com/watch?v=h27CAXRgd94>

LeetCode Comment (**Intuition** @deleted\_user):

<https://leetcode.com/problems/maximum-equal-frequency/discuss/404144/Python-long-explanation-with-lots-of-comments-in-the-code>.

happygirlzt’s Video :

<https://www.youtube.com/watch?v=nbw-jm4S1bc>

Errichto’s Video :

<https://www.youtube.com/watch?v=6kaRxT7pI4I&feature=youtu.be&t=636>

# Week 23 - Graph

## Editorial

#### (1) Clone a graph (Not that easy as it looks)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) DFS

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) BFS

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) Detect A cycle in Undirected Graph/Directed Graph

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) Topo Sort

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Number of islands (Do in Grid and Graph both)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (7) Bipartite Check

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 24 - Advance Graph

## Editorial

#### (1) SCC(using KosaRaju’s algo)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) Djisktra’s Algorithm

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) Bellman Ford Algo

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) Floyd Warshall Algorithm

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) MST using Prim’s Algo

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) MST using Kruskal’s Algo

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 25 - Dynamic Programming

## Editorial

#### (1) Max Product Subarray

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) Longest Increasing Subsequence

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (3) Longest Common Subsequence

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (4) 0-1 Knapsack

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

CodesDope Editorial :

<https://www.codesdope.com/course/algorithms-knapsack-problem/>

WilliamFiset’s Video :

<https://www.youtube.com/watch?v=cJ21moQpofY>

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) Edit Distance

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Maximum sum increasing subsequence

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (7) Matrix Chain Multiplication

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 26 - Advance Dynamic Programming

## Editorial

#### (1) Maximum sum path in matrix, (count paths, and similar type do, also backtrack to find the maximum path)

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (2) Coin change [IN PROGRESS]

**Question** :

**Question A** (Coin Change):

<https://leetcode.com/problems/coin-change/>

**Question B** (Coin Change 2):

<https://leetcode.com/problems/coin-change-2/>

**Solutions** :

**Solution A** :

Back To Back SWE’s Video :

<https://www.youtube.com/watch?v=jgiZlGzXMBw>

GFG Editorial :

<https://www.geeksforgeeks.org/find-minimum-number-of-coins-that-make-a-change/>

CodesDope Editorial :

<https://www.codesdope.com/course/algorithms-coin-change/>

**Solution B** :

Back To Back SWE’s Video :

<https://www.youtube.com/watch?v=DJ4a7cmjZY0>

GFG Editorial :

<https://www.geeksforgeeks.org/coin-change-dp-7/>

#### (3) Subset Sum [IN PROGRESS]

**Question** :

**Question A** (Perfect Sum Problem ):

<https://practice.geeksforgeeks.org/problems/perfect-sum-problem5633/1>

**Question B** (Subset Sum Problem!):

<https://www.interviewbit.com/problems/subset-sum-problem/>

**Solutions** :

**Solution A** :

GFG Editorial (Backtracking Approach):  
 <https://www.geeksforgeeks.org/subset-sum-backtracking-4/>

GFG Editorial (Optimized DP Approach):  
 <https://www.geeksforgeeks.org/perfect-sum-problem-print-subsets-given-sum/>

**Solution B** :

GFG Editorial (DP Approach):

<https://www.geeksforgeeks.org/subset-sum-problem-dp-25/>

GFG Editorial (Space Optimal Approach):

<https://www.geeksforgeeks.org/subset-sum-problem-osum-space/>

#### (4) Rod Cutting

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (5) Egg Dropping

**Question** :

[https](https://en.wikipedia.org/wiki/HTTPS)

**Solutions** :

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

#### (6) Word Break

**Question** :

**Question A** (Word Break): [SEARCH FOR MORE EXPLANATION]

<https://leetcode.com/problems/word-break/>

OR

<https://afteracademy.com/problems/word-break-problem>

OR

<https://practice.geeksforgeeks.org/problems/word-break1352/1>

**Question B** (Word Break III):

<https://leetcode.com/discuss/interview-question/385870/Google-or-Onsite-or-Word-Break-III/347600>

**Solutions** :

**Solution A** :

Knapsak’s Video :

<https://www.youtube.com/watch?v=1U4jQusbeJc>

GFG Editorial (DP Approach):

<https://www.geeksforgeeks.org/word-break-problem-dp-32/>

GFG Editorial (Using **Trie**):

<https://www.geeksforgeeks.org/word-break-problem-trie-solution/>

**Solution B** :

GFG Editorial :  
 <https://www.geeksforgeeks.org/minimum-word-break/>

#### (7) Palindrome Partitioning (MCM Variation)

**Question** :

**Question A** (Palindrome Partitioning II) :

<https://leetcode.com/problems/palindrome-partitioning-ii/>

OR

<https://www.interviewbit.com/problems/palindrome-partitioning-ii/>

OR

<https://afteracademy.com/problems/palindrome-partitioning>

OR

<https://practice.geeksforgeeks.org/problems/palindromic-patitioning4845/1>

**Question B** (Palindrome Partitioning III) :

<https://leetcode.com/problems/palindrome-partitioning-iii/>

**Solutions** :

**Solution A** :

IDeserve’s Video :

<https://www.youtube.com/watch?v=WPr1jDh3bUQ>

Arun Goel’s Video :

<https://www.youtube.com/watch?v=rmNK2awrhkU>

GFG Editorial :

<https://www.geeksforgeeks.org/palindrome-partitioning-dp-17/>

**Solution B** :

Arun Goel’s Video :  
 <https://www.youtube.com/watch?v=I4jp7c53LGI>

## Practice (Week 26)

#### Q1. Subset Sum Partition

**Problem** :

**Problem A** (Partition Equal Subset Sum):

<https://leetcode.com/problems/partition-equal-subset-sum/>

OR

<https://practice.geeksforgeeks.org/problems/subset-sum-problem2014/1>

OR

<https://afteracademy.com/problems/partition-equal-subset-sum>

**Problem B** (Partition to K Equal Sum Subsets):

<https://leetcode.com/problems/partition-to-k-equal-sum-subsets/>

OR

<https://practice.geeksforgeeks.org/problems/partition-array-to-k-subsets/1>

**Problem C** (Minimum sum partition ):

<https://practice.geeksforgeeks.org/problems/minimum-sum-partition3317/1>

**Solutions** :

**Solution A** :

GFG Editorial :

<https://www.geeksforgeeks.org/partition-problem-dp-18/>

AfterAcademy Editorial :

<https://afteracademy.com/blog/partition-equal-subset-sum>

**Solution B** :

Stable Sort’s Video :

<https://www.youtube.com/watch?v=DB-9JlnbBpM>

GFG Editorial (Recursive Approach):

<https://www.geeksforgeeks.org/partition-set-k-subsets-equal-sum/>

GFG Editorial (Using **BitMask** and DP):

<https://www.geeksforgeeks.org/partition-of-a-set-into-k-subsets-with-equal-sum-using-bitmask-and-dp/>

**Solution C** :

GFG Editorial :

<https://www.geeksforgeeks.org/partition-a-set-into-two-subsets-such-that-the-difference-of-subset-sums-is-minimum/>

# Week 27 - Heap

## Practice (Week 27)

#### Q1. Top K Frequent Words/Elements

**Problem** :

**Problem A** (Top K Frequent Words): [SEARCH FOR MORE EXPLANATION]

<https://leetcode.com/problems/top-k-frequent-words/>

**Problem B** (Top K Frequent Elements):

<https://leetcode.com/problems/top-k-frequent-elements/>

**Solutions** :

**Solution A** :

Michael Muinos’s Video (Using Priority Queue) :

<https://www.youtube.com/watch?v=cupg2TGIkyM>

**Solution B** :

LeetCode Article :

<https://leetcode.com/problems/top-k-frequent-elements/solution/>

GFG Editorial :

<https://www.geeksforgeeks.org/find-k-numbers-occurrences-given-array/>

# Week 28 - OS (Operating System)

#### (1) Love Babbar Cheat Sheet

**Cheat Sheet Link :**

<https://whimsical.com/operating-system-cheatsheet-by-love-babbar-S9tuWBCSQfzoBRF5EDNinQ>   
  
**Original Video Link :**<https://www.youtube.com/watch?v=SWBjv-GU3VQ>

## Sub-Section Name

#### (1) Resource name is written over here

**Resource Link :**

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 29 - DBMS (Database Management System)

#### (1) Love Babbar RoadMap

**Cheat Sheet Link :**

<https://whimsical.com/dbms-roadmap-by-love-babbar-FmUi8ffVop33t3MmpVxPCo>   
  
**Original Video Link :**<https://www.youtube.com/watch?v=BQBGORBPytw>

## Sub-Section Name

#### (1) Resource name is written over here

**Resource Link :**

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 30 - CN (Computer Networking)

#### (1) Resource name is written over here

**Resource Link :**

[https](https://en.wikipedia.org/wiki/HTTPS)

## Sub-Section Name

#### (1) Resource name is written over here

**Resource Link :**

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 31 - OOP (Object Oriented Programming)

#### (1) Resource name is written over here

**Resource Link :**

[https](https://en.wikipedia.org/wiki/HTTPS)

## Sub-Section Name

#### (1) Resource name is written over here

**Resource Link :**

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)

# Week 32 - System Design

## Sub-Section Name

#### (1) Resource name is written over here

**Resource Link :**

Channel Name’s Video :

[https](https://en.wikipedia.org/wiki/HTTPS)