A BETTER VERSION ->>	https://neetcode.jo/			
A DETTER VERSION ->>				
Video Solution	Category	Name	Link Notes	
https://youtu.be/KLIXCFG5TnA	Arrays	Two Sum	https://leetcode_use hash map to instantly check for difference value, map will add index of last occurrence of a num, don't use same element twice;	
https://youtu.be/1pkOgXD63yU	Arrays	Best Time to Buy and Sell Stock	https://leetcode_find local min and search for local max, sliding window;	
	Arrays	Contains Duplicate	https://leetcode hashset to get unique values in array, to check for duplicates easily	
https://youtu.be/bNvlQl2wAjk	Arrays	Product of Array Except Self	https://leetcode_make two passes, first in-order, second in-reverse, to compute products	
	Arrays		https://leetcode.pattern: prev subarray cant be negative, dynamic programming: compute max sum for each prefix	
https://youtu.be/IXVy6YWFcRM	Arrays	Maximum Product Subarray	https://leetcode_dp: compute max and max-abs-val for each prefix subarr;	
https://youtu.be/nIVW4P8b1VA	Arrays	Find Minimum in Rotated Sorted Arra	https://leetcode_check if half of array is sorted in order to find pivot, arr is guaranteed to be in at most two sorted subarrays	
https://youtu.be/U8XENwh8Oy8	Arrays	Search in Rotated Sorted Array	https://leetcode_ at most two sorted halfs, mid will be apart of left sorted or right sorted, if target is in range of sorted portion then search it, otherwise search other half	
https://youtu.be/jzZsG8n2R9A	Arrays	3Sum	https://leetcode; sort input, for each first element, find next two where -a = b+c, if a = prevA, skip a, if b = prevA skip b to elim duplicates; to find b,c use two pointers, left/right on remaining list;	
https://youtu.be/UuiTKBwPgAo	Arrays	Container With Most Water	https://leetcode/shrinking window, left/right initially at endpoints, shift the pointer with min height;	
https://youtu.be/gVUrDV4tZfY	Binary	Sum of Two Integers	https://leetcode add bit by bit, be mindful of carry, after adding, if carry is still 1, then add it as well;	
https://youtu.be/5Km3utixwZs	Binary	Number of 1 Bits	https://leetcode/modulo, and dividing n; mod and div are expensive, to divide use bit shift, instead of mod to get 1's place use bitwise & 1;	
	Binary	Counting Bits	https://leetcode write out result for num=16 to figure out pattern; res[i] = res[i - offset], where offset is the biggest power of 2 <= 1;	
	Binary	Missing Number	https://leetcode_compute expected sum - real sum; xor n with each index and value;	
	Binary	Reverse Bits	https://leetcode_reverse each of 32 bits:	
https://youtu.be/Y0IT9Fck7ql	Dynamic Programming	Climbing Stairs	https://leetcode_subproblem find (n-1) and (n-2), sum = n;	
	Dynamic Programming	Coin Change	https://leetcode_top-down: recursive dfs, for amount, branch for each coin, cache to store prev coin_count for each amount; bottom-up: compute coins for amount = 1, up until n, using for each coin (amount - coin), cache prev values	
	Dynamic Programming		https://leatcade/recursive: foreach num, get subseq with num and without num, only include num if prev was less, cache solution of each; dopsubseq length which must end with each num, curr num must be after a prev do or by itself;	
	Dynamic Programming	Longest Common Subsequence	INDEX_INSECSED_TEXT CONTINUES TO A STATE OF THE PROPERTY OF TH	
	Dynamic Programming	Word Break Problem	INLEST PRECEDED. PRECEDED. PRECEDED TO PRECEDE THE SECRET PRESEDENCE PRECEDED TO PRECEDE THE SECRET PRECEDE THE SECRET PRECEDED TO PRECEDE THE SECRET PRECEDED THE SECRET PRECEDED TO PRECEDE THE SECRET PRECEDE THE SECRET PRECEDED TO PRECEDE THE SECR	
		Combination Sum		
	Dynamic Programming		https://leetcode/usualize the decision tree, base case is curSum = or > target, each candidate can have children of itself or elements to right of it incorder to elim duplicate solutions;	
	Dynamic Programming	House Robber	https://leetcode/ for each num, get max of prev subarr, or num + prev subarr not including last element, store results of prev, and prev not including last element	
	Dynamic Programming	House Robber II	https://leetcode_subarr = arr without first & last, get max of subarr, then pick which of first/last should be added to it	
	Dynamic Programming	Decode Ways	https://leetcode_can cur char be decoded in one or two ways? Recursion -> cache -> Iterative dp solution, a lot of edge cases to determine, 52, 31, 29, 10, 20 only decoded one way, 11, 26 decoded two ways	
	Dynamic Programming	Unique Paths	https://leetcode_work backwards from solution, store paths for each position in grid, to further optimize, we don't store whole grid, only need to store prev row;	
	Dynamic Programming	Jump Game	httos://leetcode_visualize the recursive tree, cache solution for O(n) time/mem complexity, iterative is O(1) mem, just iterate backwards to see if element can reach goal node, if yes, then set it equal to goal node, continue;	
https://youtu.be/mQeF6bN8hMk	Graph	Clone Graph	https://leetcode_recursive dfs, hashmap for visited nodes	
https://youtu.be/EgI5nU9etnU	Graph	Course Schedule	https://leetcode_build adjacentcy_list with edges, run dfs on each V, if while dfs on V we see V again, then loop exists, otherwise V isnt in a loop, 3 states not visited, visited, still visiting	
https://youtu.be/s-VkcjHqkGl	Graph	Pacific Atlantic Water Flow	https://leetcode. dfs each cell, keep track of visited, and track which reach pac, atl; dfs on cells adjacent to pac, atl, find overlap of cells that are visited by both pac and atl cells;	
https://youtu.be/pV2kpPD66nE	Graph	Number of Islands	https://leetcode, foreach cell, if cell is 1 and unvisited run dfs, increment cound and marking each contigous 1 as visited	
https://youtu.be/P6RZZMu_maU	Graph	Longest Consecutive Sequence	https://leetcode_use bruteforce and try to optimize, consider the max subseq containing each num; add each num to hashset, for each num if num-1 doesn't exist, count the consecutive nums after num, ie num+1; there is also a union-find solution;	
https://voutu.be/6kTZYvNNvps	Graph	Alien Dictionary (Leetcode Premium)	https://leetcode/ chars of a word not in order, the words are in order, find adjacency list of each unique char by iterating through adjacent words and finding first chars that are different, run topsort on graph and do loop detection;	
https://youtu.be/bXsUuownnoO	Granh		https://leetcode/ union find, if union return false, loop exists, at end size must equal n, or its not connected; dfs to get size and check for loop, since each edge is double, before dfs on neighbor of N, remove N from neighbor list of neighbor;	
https://youtu.be/8f1XPm4WOUc	Granh		https://leetcode/ dfs on each node that hasn't been visited, increment component count, adjacency list; bfs and union find are possible;	
https://youtu.be/A8NUOmlwOIM	Interval		https://lectode insert new interval in order, then merge intervals; newinterval could only merge with one interval that come before it, then add remaining intervals;	
https://youtu.be/44H3cEC2fFM	Interval	Merge Intervals	https://lectocale.isort each interval, overlapping interests should be adjacent, iterate and build solution; also graph method, less efficient, more complicated	
	Interval	Non-overlanning Intervals	INDEX_PRECIOUS_SOFT each innerval, overlapping intervals should be able to interval, but each interval; Soft each interval Soft each interval Soft each	
		Non-overlapping intervals		
		Advantion December (Legitarda December)		
https://youtu.be/PaJxqZVPhbg			https://eecode/sort intervals by start time, if second interval doesn't overlap with first, then third def wont overlap with first;	
https://youtu.be/FdzJmTCVyJU	Interval	Meeting Rooms II (Leetcode Premium	https://leetcode_we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using	g min heap;
https://youtu.be/FdzJmTCVyJU https://youtu.be/GO_I-ZF0S38	Linked List	Meeting Rooms II (Leetcode Premium Reverse a Linked List	https://leetcode_we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode_iterate through maintaining cur and prev; recursively reverse, return new head of list	g min heap;
https://youtu.be/FdzJmTCVyJU https://youtu.be/GO_I-ZF0S38 https://youtu.be/g8Te7IFR3vc	Linked List Linked List	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List	http://keetcode_we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode/ terrate through maintaining cur and prev; recursively reverse, return new head of list https://leetcode/ cit to remember visited nodes; two pointers at different speeds, if they meet there is loop	g min heap;
https://youtu.be/FdzImTCVyJU https://youtu.be/GO_L-ZF0538 https://youtu.be/gBTe7IFR3vc https://youtu.be/Xldigk956u0	Interval Linked List Linked List Linked List	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists	https://leetcode_ we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode iterate through maintaining cur and prev; recursively reverse, return new head of list https://leetcode it to remember visited nodes; two pointers at different speeds, if they meet there is loop https://leetcode insert each node from one list into the other	g min heap;
https://voutu.be/FdzJmTCVyJU https://voutu.be/GD_LZF0538 https://voutu.be/gBTzJFR3vc https://voutu.be/gBTzJFR3vc https://voutu.be/gSdS5u0 https://voutu.be/gSa5OiGbT6Q	Linked List Linked List Linked List Linked List Linked List	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge K Sorted Lists	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode. Iterate through maintaining cur and prev, recursively reverse, return new head of list https://leetcode in the return of the r	g min heap;
https://youtu.be/fdzimTCVyIU https://youtu.be/60_L-ZF0S38 https://youtu.be/g8fe7lFR3vx https://youtu.be/g8fe956u0 https://youtu.be/g5a50(65f5Q https://youtu.be/g5a50(65f5Q https://youtu.be/gVuQxVej6y8	Interval Linked List Linked List Linked List Linked List Linked List	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge K Sorted Lists Remove Nth Node From End Of List	https://beetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://beetcode insert each number of counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://beetcode insert each number of counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://beetcode insert each number of the counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://beetcode insert each number of the counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://beetcode.jeu during num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://beetcode.jeu during num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://beetcode.jeu during num of meetings going at these points in time; for each meeting check if a prev meeting check if a pre	g min heap;
https://youtu.be/f4zlmTCVylU https://youtu.be/f6D.1zF0538 https://youtu.be/g8tE7JF83v https://youtu.be/g8tE7JF83v https://youtu.be/g8s5050F15Q https://youtu.be/g8s5050F15Q https://youtu.be/g8s5050F15Q https://youtu.be/g8s50fUTRKM	Interval Linked List	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge K Sorted Lists Remove Nth Node From End Of List Reorder List	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode.given traverse through maintaining cur and prev; recursively reverse, return new head of list https://leetcode.given to remember with endowed stop points and different speecks, if they meet there is loop https://leetcode.given teach node from one list into the other https://leetcode divided and conqueer, merge lists, N totalnodes, kists, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode divided and conqueer, merge lists, N totalnodes, kists, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode used unmay node at head of list, compute len of list; two pointerse, second has offset of n from first; https://leetcode/gereverse second half of list, then easily reorder it; non-optimal way is to store list in array.	g min heap;
https://youtu.be/f4zim/tCv/UU https://youtu.be/f6D.12F0538 https://youtu.be/f6D.12F0538 https://youtu.be/f6D.12F0538 https://youtu.be/Xdigle/95600 https://youtu.be/Xdigle/95600 https://youtu.be/Xdigle/9560 https://youtu.be/Xdigle/958 https://youtu.be/Xdi	Interval Linked List Matrix	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge K Sorted Lists Remove Nth Node From End Of List	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode.iterate through maintaining cur and prev; recursively reverse, return new head of list https://leetcode insert each node from one its into the other https://leetcode insert each node from one list into the other https://leetcode.	g min heap;
https://youtu.be/f4zlmTCVylU https://youtu.be/f6D.1zF0538 https://youtu.be/g8tE7JF83v https://youtu.be/g8tE7JF83v https://youtu.be/g8s5050F15Q https://youtu.be/g8s5050F15Q https://youtu.be/g8s5050F15Q https://youtu.be/g8s50fUTRKM	Interval Linked List	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge K Sorted Lists Remove Nth Node From End Of List Reorder List	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode.given traverse through maintaining cur and prev; recursively reverse, return new head of list https://leetcode.given to remember with endowed stop points and different speecks, if they meet there is loop https://leetcode.given teach node from one list into the other https://leetcode divided and conqueer, merge lists, N totalnodes, kists, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode divided and conqueer, merge lists, N totalnodes, kists, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode used unmay node at head of list, compute len of list; two pointerse, second has offset of n from first; https://leetcode/gereverse second half of list, then easily reorder it; non-optimal way is to store list in array.	g min heap;
https://woutu.be/f6zlmtCVyUU https://woutu.be/f6D-12F0538 https://woutu.be/f6D-12F0538 https://woutu.be/f6D-12F0538 https://woutu.be/Ndigle95500 https://woutu.be/Ndigle95500 https://woutu.be/Ndigle95500 https://woutu.be/Ndigle958 https://woutu.be/Ndigle958 https://woutu.be/SbidfUTKIM https://woutu.be/f3linfUSPnw https://woutu.be/f3li	Interval Linked List Matrix	Meeting Rooms II (Leetzode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge K Sorted Lists Remove Nith Node From End Of List Recorder List Set Matrix Zeroes Spiral Matrix Rotate Image	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode. International Continue of the previous of the points of the previous	g min heap;
https://youtu.be/f4tJmTCVylU https://youtu.be/f6b1.E7E538 https://youtu.be/f5b1E7E7E8xc https://youtu.be/f5s5G0f5h5Q https://youtu.be/f5s5G0f5h5Q https://youtu.be/f5s5G0ffhfkM https://youtu.be/f5s5G0ffhfkM https://youtu.be/f5sh6dUffkM https://youtu.be/f5sh6dUffkM https://youtu.be/f5sh6dUffkM https://youtu.be/f3hfAESxcOtw https://youtu.be/f3hfAESxcOtw https://youtu.be/f3hfAESxcOtw	Interval Linked List Matrix Matrix Matrix Matrix Matrix Matrix	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in Linked List Detect Cycle in Linked List Merge I'vo Sorted Lists Merge K Sorted Lists Remove Nth Node From End Of List Recorder List Sex Matrix Zeroes Spiral Matrix Rotate Image Word Search	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode divide and conquer, merge lists, N totalmodes, k-lists, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode insert each node from one list into the other https://leetcode divide and conquer, merge lists, N totalmodes, k-lists, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode is use dummy node at head of list, compute len of list; two pointers, second has offset of a from first; https://leetcode is used storm when the easily reporder it, non-optimal way is to store list in arrayy. https://leetcode is used storm keep track of visited cells; and endowed revisited cells in arrayy of the leeps of one ach cell, for each search remember visited cells, and remove cur visited cell right before you return from dist;	g min heap;
https://youtu.be/f4tJmTCVylU https://youtu.be/f6b1.E7E538 https://youtu.be/f5b1E7E7E8xc https://youtu.be/f5s5G0f5h5Q https://youtu.be/f5s5G0f5h5Q https://youtu.be/f5s5G0ffhfkM https://youtu.be/f5s5G0ffhfkM https://youtu.be/f5sh6dUffkM https://youtu.be/f5sh6dUffkM https://youtu.be/f5sh6dUffkM https://youtu.be/f3hfAESxcOtw https://youtu.be/f3hfAESxcOtw https://youtu.be/f3hfAESxcOtw	Interval Linked List Matrix Matrix Matrix	Meeting Rooms II (Leetzode Premium Reverse a United List Detect Cycle in Linide List Merge Two Sorted Lists Merge K Sorted Lists Merge K Sorted Lists Remove Nth Node From End Of List Recorder List Set Matrix Zeroes Spiral Matrix Rotate Image Word Searth Longest Substring Without Repeating Longest Substring Without Repeating Longest Substring Without Repeating	https://leetcode, we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode, terrate through maintaining cur and prev; recursively reverse, return new head of list. this children code is to remember visited endoes; two pointers at different speecks, if they meet there is loop https://leetcode, divide and conquer, merge lists, N totalnodes, kilsts, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode; divided and conquer, merge lists, N totalnodes, kilsts, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode; vewers escond half of list, then easily reorder it; non-optimal way is to store list in array; https://leetcode; use sets to keep track of visited cells; keep track of visited cells, and remove cur visited cell right before you return from dfs; https://leetcode; diding window, if we see same chart retwice within curr window, shift start position;	g min heap;
https://youtu.be/fdi.mTCVylU https://youtu.be/fdi.mTCVylU https://youtu.be/fdi.mTCVylU https://youtu.be/fdi.mTCPi https://youtu.be/siaSiGiGiFIG https://youtu.be/siaSiGiGiFIG https://youtu.be/siaSiGiGiFIG https://youtu.be/SibfUTKUM https://youtu.be/SibfUTKUM https://youtu.be/SibfUTKUM https://youtu.be/sibfUTKUM https://youtu.be/sibfUTKUM https://youtu.be/sibfUTKUM https://youtu.be/sibfGiG_PSIgE https://youtu.be/sibfGiG_PSIgE https://youtu.be/sibfGiG_PSIgE	Interval Linked List Matrix Matrix Matrix Matrix Matrix Matrix	Meeting Rooms II (Leetzode Premium Reverse a United List Detect Cycle in Linide List Merge Two Sorted Lists Merge K Sorted Lists Merge K Sorted Lists Remove Nth Node From End Of List Recorder List Set Matrix Zeroes Spiral Matrix Rotate Image Word Searth Longest Substring Without Repeating Longest Substring Without Repeating Longest Substring Without Repeating	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode divide and conquer, merge lists, N totalmodes, k-lists, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode insert each node from one list into the other https://leetcode divide and conquer, merge lists, N totalmodes, k-lists, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode is use dummy node at head of list, compute len of list; two pointers, second has offset of a from first; https://leetcode is used storm when the easily reporder it, non-optimal way is to store list in arrayy. https://leetcode is used storm keep track of visited cells; and endowed revisited cells in arrayy of the leeps of one ach cell, for each search remember visited cells, and remove cur visited cell right before you return from dist;	g min heap;
https://youtu.be/f4timTCVylU https://youtu.be/f6b1.PE0538 https://youtu.be/f6b1.PE0538 https://youtu.be/f3timSystem https://youtu.be/Sid5500 https://youtu.be/Sid5500 https://youtu.be/Sid5500 https://youtu.be/Sid55001F6Q https://youtu.be/Sid601F6Q https://youtu.be/Sid601F6Q https://youtu.be/Sid601F6Q https://youtu.be/Sid601F6Q https://youtu.be/BimXZNvuliIM https://youtu.be/BimXZNvuliIM https://youtu.be/BimXZNvuliIM https://youtu.be/BimSid552C01W https://youtu.be/wiGpQevYHdEQ https://youtu.be/wiGpQevYHdEQ https://youtu.be/wiGpQevYHdEQ https://youtu.be/graXUIUAABsh	Interval Linked List Matrix Matrix Matrix Matrix String String String	Neeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge Two Sorted Lists Remove Nith Node From End Of List Remove Nith Node From End Of List Rest Native Spring Martix Roote India Spring Martix	https://leetcode, we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode, terrate through maintaining cur and prev; recursively reverse, return new head of list. this children code is to remember visited endoes; two pointers at different speecks, if they meet there is loop https://leetcode, divide and conquer, merge lists, N totalnodes, kilsts, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode; divided and conquer, merge lists, N totalnodes, kilsts, O(N*logk). For each list, find min val, insert it into list, use priorityQ to optimize finding min O(N*logk) https://leetcode; vewers escond half of list, then easily reorder it; non-optimal way is to store list in array; https://leetcode; use sets to keep track of visited cells; keep track of visited cells, and remove cur visited cell right before you return from dfs; https://leetcode; diding window, if we see same chart retwice within curr window, shift start position;	g min heap;
https://youtu.be/f4timTCVylU https://youtu.be/f6b1.PE0538 https://youtu.be/f6b1.PE0538 https://youtu.be/f3timSystem https://youtu.be/Sid5500 https://youtu.be/Sid5500 https://youtu.be/Sid5500 https://youtu.be/Sid55001F6Q https://youtu.be/Sid601F6Q https://youtu.be/Sid601F6Q https://youtu.be/Sid601F6Q https://youtu.be/Sid601F6Q https://youtu.be/BimXZNvuliIM https://youtu.be/BimXZNvuliIM https://youtu.be/BimXZNvuliIM https://youtu.be/BimSid552C01W https://youtu.be/wiGpQevYHdEQ https://youtu.be/wiGpQevYHdEQ https://youtu.be/wiGpQevYHdEQ https://youtu.be/graXUIUAABsh	Interval Linked List Matrix Matrix Matrix Matrix String String	Neeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge Two Sorted Lists Remove Nith Node From End Of List Remove Nith Node From End Of List Rest Native Spring Martix Roote India Spring Martix	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode; therate through maintaining cur and prev; recursively reverse, return new head of list https://leetcode; diet to remember visited nodes; two pointers at different speeds, if they meet there is loop https://leetcode; insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode is used unmay node at head of list, tompute len of list; two pointers, second has offset of n from first; https://leetcode is used unmay node at head of list, tompute len of list; two pointers, second has offset of n from first; https://leetcode is used to keep track of all rows, cols to zero out, after, for each num if it is in a rarry. https://leetcode is keep track of visited cells; keep track of boundaries, layer-by-layer; that it is a square as advantage, rotate positions in reverse order, store a in temp, a = b, b = c, c = d, d = temp; https://leetcode is on each rel, for each search remember visited cells, and remove cur visited cell leight before you return from dfs; https://leetcode is on each rel, for each search remember visited cells, and remove cur visited cell leight before you return from dfs; https://leetcode is on each rel, for each search remember visited cells, and remove cur visited cell leight before you return from dfs; https://leetcode is on each rel, for each search remember visited cells, and remove cur visited cell leight before you return from dfs; https://leetcode is not controlled on the relation of the result of th	g min heap;
https://youtu.be/f4timTCVylU https://youtu.be/f6b1.PE0538 https://youtu.be/f6b1.PE0538 https://youtu.be/f3timSystem https://youtu.be/Sid5500 https://youtu.be/Sid5500 https://youtu.be/Sid5500 https://youtu.be/Sid55001F6Q https://youtu.be/Sid601F6Q https://youtu.be/Sid601F6Q https://youtu.be/Sid601F6Q https://youtu.be/Sid601F6Q https://youtu.be/BimXZNvuliIM https://youtu.be/BimXZNvuliIM https://youtu.be/BimXZNvuliIM https://youtu.be/BimSid552C01W https://youtu.be/wiGpQevYHdEQ https://youtu.be/wiGpQevYHdEQ https://youtu.be/wiGpQevYHdEQ https://youtu.be/graXUIUAABsh	Interval Linked List Matrix Matrix Matrix Matrix String String String	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in Linked List Detect Cycle in Linked List Merge R Wo Sorred Lists Merge R Sorted Lists Remove Nth Node From End Of List Recorder List Set Matrix Zeroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Longest Repeating Character Replace Minimum Window Substring	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode in the previous first of the previous first	g min heap;
https://youtu.be/f4t/mTCVyIU https://youtu.be/f4t/mTCVyIU https://youtu.be/f6t-EF6538 https://youtu.be/f6t-EF67F83x https://youtu.be/f3t65506 https://youtu.be/f3t65060 https://youtu.be/f3t65506 https://youtu.be/f3t65060F50 https://youtu.be/f3t6106797 https://youtu.be/ShidnUrktM https://youtu.be/shidnOrCA h	Interval Unked List Linked List Linked List Linked List Linked List Linked List Linked List Matrix Matrix Matrix Matrix Matrix String String String String	Meeting Rooms II (Lectcode Premium Revees a Linked List Detect Cycle in a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge Was Serdel Lists Remove NIS Node From End Of List Recorder List Seconder List Linked Lin	https://lectcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://lectcode.pice.ic.ic.ic.ic.ic.ic.ic.ic.ic.ic.ic.ic.ic.	g min heap;
https://youtu.be/fstimTCVylU https://youtu.be/fstimTCVylU https://youtu.be/fstimTCVylU https://youtu.be/fstimTRBxs https://youtu.be/Sid=Sid=Sid=Sid=Sid=Sid=Sid=Sid=Sid=Sid=	Interval Linked List Linked List Linked List Linked List Linked List Linked List Matrix Matrix Matrix Matrix Matrix String String String String String String	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in Linked List Detect Cycle in Linked List Merge Two Sorted Lists Merge K Sorted Lists Remore Rith Node From End Of List Recorder List Set Matrix Zeroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Character Replace Minimum Window Substring Walld Anagram Group Anagrams	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode iterate through maintaining cur and prev; recursively reverse, return mew head of list https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode is used unmay node at head of list, compute len of list; two pointers, second has offset of n from first; https://leetcode is used unmay node at head of list, compute len of list; two pointers, second has offset of n from first; https://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of n from first; https://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of n from first; https://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of n from first; https://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of n from first; https://leetcode is used to list of	g min heap;
https://youtu.be/f4zim/tCv/UU https://youtu.be/f4zim/tCv/UU https://youtu.be/f6D_L2E0538 https://youtu.be/f6D_L2E0538 https://youtu.be/f6D_L2E0538 https://youtu.be/f6Ig0187580c https://youtu.be/f6Ig0187580c https://youtu.be/f6Ig0187580c https://youtu.be/f6Ig01879m https://youtu.be/f6Ig01879m https://youtu.be/f6Ig01879m https://youtu.be/f6Ig01879m https://youtu.be/f6IG0_P5Le8E	Interval Linked List Matrix Matrix Matrix Matrix Matrix String String String String String String	Neeting Rooms II (Leetcode Premium Reverse a Linked List Merge Two Sorted Lists Merge Two Sorted Lists Nerge K Sorted Lists Remove Nth Node From End Of List Remove Nth Node From End Of List Remove List Set Matrix Remove Set Matrix Remove Rosted Remove Rosted Remove Rosted Remove Rosted Remove List Set Matrix Remove List Set Matrix Remove List Remove Rosted Remove	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode in this point in the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode in the previous points and ifferens used. If the previous points are different with the previous cit. to remain where the different with the previous cit. to remain which is not one to the https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other hittps://leetcode insert each node from one list into the other hittps://leetcode insert each node from one list into the other hittps://leetcode insert each node from one list into the other hittps://leetcode insert each node from one list into the other hittps://leetcode insert each node from one list into the other hittps://leetcode insert each node from one list into the other hittps://leetcode insert each node from one list into the other hittps://leetcode insert each node from one list into the other hittps://leetcode insert each node from one list into the needs in start of unique the visit of each sart into the needs to be zeroed; https://leetcode into visited cells, increment len, if in ot, shift window right; https://leetcode into visited cells, into an end call or each search remember visited cells, and remove cur visited cell light before your return from dfs; https://leetcode into visited cells, and remove cu	g min heap;
https://youtu.be/f4timTCVyIU https://youtu.be/f6timTCVyIU https://youtu.be/f6timTCVyIU https://youtu.be/f6timTCPiBSS https://youtu.be/SidSidSidSidSidSidSidSidSidSidSidSidSidS	Interval Linked List String	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in Linked List Merge It was Serviced Lists Merge Was Serviced Lists Remove Nth Node From End Of List Reorder List Sex Matrix Zeroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Longest Substring Without Repeating Longest Repeating Character Replace Minimum Window Substring Valid Anageram Group Anageram Valid Parentheses Valid Parentheses Valid Parentheses	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode international curry and prev; recursively reverse, return new head of list https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode use dummy node at head of list, compute len of list; two pointers, second has offset of a from first; the county of the each of list, compute len of list; two pointers, second has offset of a from first; this offset of lists; in a range; use set to keep track of all rows, color to care on us, after, for each num if it is in a zero row or col then change it to 0; flag first cell in row, and col to mark row/col that needs to be zeroed; https://leetcode use of the start is a square as advantage, rotate positions in reverse order, store a in temps, a = b, b = c, c = d, d = temp; https://leetcode listing window, if we see same char twice within curr window, shift start position; https://leetcode padding window, if we see same char twice within curr window, shift start position; https://leetcode padding window, if we see same char twice within curr window, shift start position; https://leetcode padding window, if we see same char twice within curr window, shift start position; https://leetcode padding window, if we see same char twice within curr window, shift start position; https://leetcode padding window, if we see same char twice within curr window, shift window right; https://leetcode padding window, if we see same char twice within curr window, shift window window; padding window, window, if we see same char twice within curr window, shift window window; padding window window is a statisfied; listing window, window is a statisfied; listing window, window is a statisfied; listing window window i	g min heap;
https://youtu.be/f6zim/tCV/UI https://youtu.be/G0_L2E0538 https://youtu.be/G0_L2E0538 https://youtu.be/G0_L2E0538 https://youtu.be/S16E07ER3xx https://youtu.be/S16E07ER3xx https://youtu.be/S16E07ER5x https://youtu.be/S16E07ER5	Interval Int	Meeting Rooms II (Lectcode Premium Revene a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge Row Sorted Lists Merge Row Sorted Lists Remove NISh Node From End Of List Node Indiana Nish Nish Nish Nish Nish Nish Nish Nish	https://lectode. we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://lectode.in.com/previous/ending-current-previous-pre	g min heap;
https://youtu.be/f4t/mTCV/UU https://youtu.be/f4t/mTCV/UU https://youtu.be/f6t/EE/EE538 https://youtu.be/f8t/EE/EE538 https://youtu.be/f8t/EE/EE538 https://youtu.be/f8t/EE/EE5488 https://youtu.be/f8t/EE5696	Interval Linked List String	Meeting Rooms II (Leetcode Premium Reverse a United List Detect Cycle in Linked List Detect Cycle in Linked List Merge Two Sorted Lists Merge Roomed Lists Remove Nth Node From End Of List Recorder List Set Matrix Zeroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Character Replace Minimum Window Substring Valid Anagram Group Anagrams Valid Parentheses Valid Paindrome Longest Palindromic Substring	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode iterate through maintaining cur and prev; recursively reverse, return mew head of list https://leetcode iterate through maintaining cur and prev; recursively reverse, return mew head of list https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode issert each node from one list into the other https://leetcode issert each node from one list into the other https://leetcode issert each node from one list into the other https://leetcode issert each node from one list into the other https://leetcode issert each node from one list into the other https://leetcode issert each node from one list into the other https://leetcode issert each node from one list into the other https://leetcode issert each node from one list into the other https://leetcode issert each node from one list into the other https://leetcode issert each node from one list into the other https://leetcode issert each node from one list into the other https://leetcode issert each node from one list into the neasily reorder (t; non-optimal way) is to store list in array. https://leetcode issert each node from one list into the neasily reorder (t; non-optimal way) is to store list in array. https://leetcode issert layer-by-layer, use that it's a square as advantage, rotate positions in reverse order, store a in temp, a = b, b < c, c < d, d = temp; https://leetcode issert layer-by-layer, use that it's a square as advantage, rotate positions in reverse order, store a in temp, a = b, b < c, c < d, d = temp; https://leetcode issert layer-by-layer, use that it's a square as advantage, rotate positions in reverse order, store a in temp or variety in the co	g min heap;
https://woutu.be/fsilmft/cVyUl https://woutu.be/fsilmft/cVyUl https://woutu.be/GD_12F0538 https://woutu.be/Jsilff2FIPF83vc https://woutu.be/Jsilff2FIPF84vc https://woutu.be/Jsilff2FIRF84vc https://w	Interval Linked List Matrix Matrix Matrix Matrix String	Meeting Rooms II (Lectcode Premium Rovens a Linked List Detect Cycle in a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge Was Sorted Lists Remove NIS Node From End Of List Remove NIS Node From End Of List Recorder List Set Matrix Zeroes Sprint Matrix Router Linguist Substring Without Repeating Longest Substring Without Repeating Character Replace Minimum Window Substring Walid Anagram Valid Anagrams Valid Parentheses Valid Palindromic Substring Palindromic	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode. Iterate through maintaining cur and prev, recursively reverse, return new head of list. White profession of the county of the	g min heap;
https://youtu.be/f4zim/tCv/UJ https://youtu.be/f4zim/tCv/UJ https://youtu.be/f6D_L2E0538 https://youtu.be/f6D_L2E0538 https://youtu.be/f6D_L2E0538 https://youtu.be/f3d[s6]55000 https://youtu.be/f3d[s6]5500 https://youtu.be/f3d[s6]5500 https://youtu.be/f3d[s6]5500 https://youtu.be/f3d[s6]5500 https://youtu.be/f3d[s6]55000 https://youtu.be/f3d[s6]5500 https://youtu.be/f3d[s6]5500 https://youtu.be/f3d[s6]5000 https://youtu.be/f3d[s6]50000 https://youtu.be/f3d[s6]50000 https://youtu.be/f3d[s6]5000000000000000000000000000000000000	Interval Linked List Matrix Matrix Matrix String	Neeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge Two Sorted Lists Remove Nith Node From End Of List Recorder List See Matrix Zeroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Longest Repeating Character Replace Minimum Window Substring Walled Anagram Group Anagrams Valid Parentheses Valid Parindrome Longest Repeating Character Replace Minimum Window Substring Valid Parentheses Valid Parindrome Longest Repeating Character Replace Minimum Uniform Substring Palindromic Substring Palindromic Substring Encode and Decode Strings (Leetcode Maximum Depth of Binary Tree Same Tree	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode intros/fleetcode intros/fleetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode insert each node from one list into the other https://leetcode is used unmay note at head of list, compute len of list; two pointers, second has offset of n from first; https://leetcode is used unmay note at head of list, compute len of list; two pointers, second has offset of n from first; https://leetcode is used to the measily reorder it; non-optimal way is to store list in array; https://leetcode is used to keep track of boundaries, layer-by-layer; https://leetcode is keep track of visited cells; keep track of boundaries, layer-by-layer; https://leetcode is on each cell, for each search remember visited cells, and remove cur visited cell right before you return from dfs; https://leetcode is on each cell, for each search remember visited cells, and remove cur visited cell right before you return from dfs; https://leetcode is diding window, if we see same char twice within curr window, shift start position; https://leetcode intos./leetcode peed is num of unique char in f, HAVE is num of char we have valid count for, sliding window, move right until valid, increment left until invalid, to check validity keep track if the count of each char in start, deptile in the count of each char in start, deptile in the count of each char in start, deptile and phanium, companied that of an apphanium to count each char in start, and phanium to companie left and right, continue until left >= right, don't distinguish between upper/lowercase;	g min heap;
https://youtu.be/f4t/mTCVylU https://youtu.be/f6t/mTCVylU https://youtu.be/f6t/mTCVylU https://youtu.be/f6t/mTCVylU https://youtu.be/f5t/mTCVylU https://youtu.be/f5t/mTCVylU https://youtu.be/f5t/mTCVylU https://youtu.be/f5t/mTCVylU https://youtu.be/f5t/mTCVylU https://youtu.be/f5t/mTCV https://youtu.be/f5t/mTCV https://youtu.be/fflQ https://youtu.be/flQ https://youtu.be/fflQ https://youtu.be/fflQ https://youtu.be/f	Interval Int	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in Linked List Merge I Wo Sorted Lists Merge I Wo Sorted Lists Remove Nth Node From End Of List Recorder List Sex Matrix Acroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Longest Substring Without Repeating Longest Repeating Character Replace Minimum Window Substring Valid Anagarm Group Anagarms Valid Parentheses Valid Parentheses Valid Parentheses Longest Repeating Character Replace Minimum Window Substring Palindromic Substring Palindromic Substring Palindromic Substring Palindromic Substring Sencode and Decede Strings (Leetcode Maximum Depth of Binary Tree	Nttos://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode instruction does two points are different separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode insert each node from one list into the other Nttos://leetcode insert each node from one list into the other Nttos://leetcode use dummy node at head of list, compute len of list; two pointers, second has offset of a from first; Nttos://leetcode; use dummy node at head of list, compute len of list; two pointers, second has offset of a from first; Nttos://leetcode; use dummy node at head of list, compute len of list; two pointers, second has offset of a from first; Nttos://leetcode; use dummy node at head of list, compute len of list; two pointers, second has offset of a from first; Nttos://leetcode; use dummy node at head of list, compute len of list; two pointers, second has offset of a from first; Nttos://leetcode; use dummy node at head of list, compute len of list; two pointers, second has offset of a from first; Nttos://leetcode; level reverse second half of list, then easily reporder; it, non-optimal way is to store list in array; Nttos://leetcode; level reverse second half of list, then easily reporder; it, non-optimal way is to store list in array; Nttos://leetcode; level prack of visited cells; keep track of boundaries, layer-by-layer; Nttos://leetcode; level prack-by-layer, use that it's a square as advantage, rotate positions in reverse order, store a in temps, a = b, b = c, c = d, d = temp; Nttos://leetcode; liding window, if we see same char twice within curr window, shift start position; Nttos://leetcode; level prack-by-layer, use that it's a square as advant	g min heap;
https://youtu.be/f6zim/tCv/UI https://youtu.be/G0_L2F0538 https://youtu.be/G0_L2F0538 https://youtu.be/G0_L2F0538 https://youtu.be/Midis/95800 https://youtu.be/Midis/95800 https://youtu.be/S560051500 https://youtu.be/S560051500 https://youtu.be/S56001700 https://youtu.be/S56001700 https://youtu.be/S56001700 https://youtu.be/S60001700 https://youtu.be/S600001700 https://youtu.be/S600001700 https://youtu.be/S6000000000000000000000000000000000000	Interval Linked List Matrix Matrix Matrix Matrix String	Neeting Rooms II (Lectcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge Row Sorted Lists Remove NISh Node From End Of List Node Indiance NISh Node Indiana Nish Nish Nish Nish Nish Nish Nish Nish	https://leetcode. we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode. iterate through maintaining cur and prev; recursively reverse, return new head of list. Interest through maintaining cur and prev; recursively reverse, return new head of list. which is the count of contract the previous of the county	g min heap;
https://youtu.be/f4t/mTCV/UU https://youtu.be/f6t/mTCV/UU https://youtu.be/f6t/EE/EE/SS8 https://youtu.be/SEE/EI/ERSxc https://youtu.be/SEE/EI/ERSxc https://youtu.be/SSEE/EI/ERSxc https://youtu.be/SSEE/EI/ERSxc https://youtu.be/SSEE/EI/ERSyc https://youtu.be/SSEE/EI/EI/ERSyc https://youtu.be/SSEE/EI/EI/EI/EI/EI/EI/EI/EI/EI/EI/EI/EI/E	Interval Linked List Matrix Matrix Matrix Matrix String	Meeting Rooms II (Leetcode Premium Reverse a United List Detect Cycle in Linked List Merge Two Sorted Lists Merge Two Sorted Lists Remove Nth Node From End Of List Remove Nth Node From End Of List Resource List Set Matrix Zeroes Spiral Matrix Resource List Set Matrix Zeroes Spiral Matrix Resource List Set Matrix Remove Repeting Character Replace World Search Longest Substring Without Repeating Character Replace Minimum Window Substring Valid Anagram Group Anagram Valid Parentheses Minimum Depth of Binary Tree Samer Tree Incode and Decode Strings (Leetcode Maximum Depth of Binary Tree Samer Tree Maximum Path Sum Binary Tree Less Tourney Teves Assimum Path Sum	https://leetcode we care about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using https://leetcode insert each node from one list into the other Intos://leetcode insert each node from one list into the other Intos://leetcode insert each node from one list into the other Intos://leetcode insert each node from one list into the other Intos://leetcode insert each node from one list into the other Intos://leetcode is used dummy node at head of list, compute len of list; two pointers, second has offset of n from first; Intos://leetcode is used dummy node at head of list, compute len of list; two pointers, second has offset of n from first; Intos://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of n from first; Intos://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of n from first; Intos://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of n from first; Intos://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of n from first; Intos://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of n from first; Intos://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of n from first; Intos://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of non-first; Intos://leetcode is used tummy node at head of list, compute len of list; two pointers, second has offset of list in array; Intos://leetcode is leep track of boundaries, layer-by-layer, use can have a subject of list of list of list layer by-layer, use that it's a square as advantage, orate positions in reverse order,	g min heap;
https://voutu.be/fsilmtCVpUI https://voutu.be/fsilmtAppI https://voutu.be/f	Interval Linked List Matrix Matrix Matrix Matrix String Tring String	Meeting Rooms II (Lectcode Premium Rooms II (Lectcode Premium Rooms III (Lectcode Premium Rooms III (Lectcode Premium Rooms III (Lectcode III III III III III III III III III I	https://lectoods/ intos/inectoods/ intos	g min heap;
https://youtu.be/fst.im/t/vylU https://youtu.be/fst.im/t/vylU https://youtu.be/fst.im/t/vylU https://youtu.be/fst.im/t/ssx https://youtu.be/rst.im/t/ssx https://youtu.be/rst.im/t/ssx https://youtu.be/fst.im/t/ssx https://youtu.be/fst.im/t/ssx https://youtu.be/fst.im/t/ssx https://youtu.be/fst.im/t/ssx https://youtu.be/fst.im/ssx https:/	Interval Linked List Matrix Matrix Matrix String Tring String	Neeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge Two Sorted Lists Remove Nith Node From End Of List Recorder List Semove Nith Node From End Of List Recorder List Semove Nith Node From End Of List Recorder List Semove Nith Node From End Of List Recorder List Semove Nith Node From End Of List Recorder List Compact Repeating Character Replace Word Search Longest Separing Character Replace Minimum Window Substring Valid Anagering Croup Anagerins Valid Parentheses Valid Palmdrome Longest Repeating Character Replace Longest Palmdromic Substring Palmdromic Substring Endorder List Substring Same Tree Simmar Tree Maximum Depth of Binary Tree Level Order Traversal Senialize and Describit Summy Tree List Substrie of Another Tree Subtree of Another Tree	https://lectoods with an analysis missing	g min heap;
https://youtu.be/f4t/mTCV/JU https://youtu.be/f4t/mTCV/JU https://youtu.be/f6t/E56328 https://youtu.be/f6t/E56328 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f3660 https://youtu.be/f36600 https://youtu.be/f36600 https://youtu.be/f36600 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f3660000 https://youtu.be/f3660000 https://youtu.be/f366000000 https://youtu.be/f36600000000000000000000000000000000000	Interval Uniked List Matrix Matrix Matrix Matrix String Tree Tree Tree Tree Tree Tree Tree Tre	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in Linked List Detect Cycle in Linked List Merge Two Sorted Lists Merge Row Sorted Lists Remore Rith More From End Of List Recorder List Set Matrix Zeroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Longest Substring Without Repeating Character Replace Minimum Window Substring Valid Anagram Group Anagrams Valid Parentheses Valid Parentheses Valid Parentheses Valid Parentheses Longest Repeating Character Replace Minimum Window Substring Palindromic Palin	Into://lectode local transports in time where we are starting-fending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time, for each meeting check if a prev meeting has finished before curr started, using https://lectode/into.//	g min heap;
https://youtu.be/f6zim/tCv/UU https://youtu.be/G0_L2E0538 https://youtu.be/G0_L2E0538 https://youtu.be/G0_L2E0538 https://youtu.be/f8ibE7JFR83x https://youtu.be/f8ibE7JFR83x https://youtu.be/f8ibE7JFR83x https://youtu.be/f8ibE7JFR93x https://yout	Interval Unked List Watrix Matrix Matrix Matrix String Tring Tree Tree Tree Tree Tree Tree Tree Tre	Neeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge Two Sorted Lists Remove Nith Node From End Of List Recorder List Semove Nith Node From End Of List Recorder List Semove Nith Node From End Of List Recorder List Semove Nith Node From End Of List Recorder List Semove Nith Node From End Of List Recorder List Compact Repeating Character Replace Word Search Longest Separing Character Replace Minimum Window Substring Valid Anagering Croup Anagerins Valid Parentheses Valid Palmdrome Longest Repeating Character Replace Longest Palmdromic Substring Palmdromic Substring Endorder List Substring Same Tree Simmar Tree Maximum Depth of Binary Tree Level Order Traversal Senialize and Describit Summy Tree List Substrie of Another Tree Subtree of Another Tree	attos://testcode into://testcode into://testco	g min heap;
https://youtu.be/f4t/mTCV/JU https://youtu.be/f4t/mTCV/JU https://youtu.be/f6t/E56328 https://youtu.be/f6t/E56328 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f365960 https://youtu.be/f3660 https://youtu.be/f36600 https://youtu.be/f36600 https://youtu.be/f36600 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f366000 https://youtu.be/f3660000 https://youtu.be/f3660000 https://youtu.be/f366000000 https://youtu.be/f36600000000000000000000000000000000000	Interval Uniked List Matrix Matrix Matrix Matrix String Tree Tree Tree Tree Tree Tree Tree Tre	Meeting Rooms II (Leetcode Premium Reverse a Linked List Detect Cycle in Linked List Detect Cycle in Linked List Merge Two Sorted Lists Merge Row Sorted Lists Remore Rith More From End Of List Recorder List Set Matrix Zeroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Longest Substring Without Repeating Character Replace Minimum Window Substring Valid Anagram Group Anagrams Valid Parentheses Valid Parentheses Valid Parentheses Valid Parentheses Longest Repeating Character Replace Minimum Window Substring Palindromic Palin	Into://lectode local transports in time where we are starting-fending a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time, for each meeting check if a prev meeting has finished before curr started, using https://lectode/into.//	g min heap;
https://youtu.be/f4t/mTCV/UU https://youtu.be/f4t/mTCV/UU https://youtu.be/f6t-EFE538 https://youtu.be/f8t-EFE7BF8xx https://youtu.be/f8t-EFE7BF8xx https://youtu.be/f8t/EFE7BF8xx https://youtu.be/f8t/EFE7BF8xx https://youtu.be/f8t/EFE7BF8xx https://youtu.be/f8t/EFEFBFXx https://youtu.be/f8t/EFEFSFXX https	Interval Unked List Watrix Matrix Matrix Matrix String Tring Tree Tree Tree Tree Tree Tree Tree Tre	Neeting Rooms II (Leetcode Premium Reverse a Linked List Merge Woo Sorted Lists Merge Woo Sorted Lists Nerge K Sorted Lists Remove Nth Node From End Of List Recorder List Set Matrix Zeroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Character Replace Minimum Window Substring Walled Amagerm Group Anagrams Valid Parentheses Valid Parentheses Valid Painforme Longest Repeating Character Replace Minimum Window Substring Valid Anagram Group Anagrams Valid Parentheses Valid Painforme Longest Repeating Character Replace Minimum Window Substring Pailindromic Substring Pailindromic Substring Pailindromic Substring Pailindromic Substring Pailindromic Substring Ferode and Decode Strings (Leetcode Maximum Depth of Binary Tree Same Tree Invert/Fiji Binary Tree Maximum Path Sum Binary Tree Maximum Path Sum Binary Tree Foot Proverties Binary Fee Foot Proverder Validate Binary Search Tree Kth Smallest Element in BST	attos://testcode into://testcode into://testco	g min heap;
https://voutu.be/f6zlmtCv/UU https://voutu.be/f6D-12F0538 https://voutu.be/f6D-12F0538 https://voutu.be/f6D-12F0538 https://voutu.be/f5ibf07F83x https://voutu.be/f5ibf07F85x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f8ibf07F81x https://voutu.be	Interval Int	Neeting Rooms II (Leetcode Premium Reverse a Linked List Merge Woo Sorted Lists Merge Woo Sorted Lists Remove Nth Node From End Of List Remove Nth Node From End Of List Remove List Set Matrix Zeroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Character Replace Minimum Window Substring Valid Anageram Group Anagrams Valid Parentheses Valid Paintonne Substring Valid Paintonne Substring Fercode and Decode Strings (Leetcode Maximum Depth of Binary Tree Same Tree Invert/Fije Binary Tree Binary Tree Maximum Path Sum Binary Tree Level Order Traversal Serialize and Descriative Binary Tree Construct Binary Tree From Percorder Validate Binary Feer Form Percorder Validate Binary Feer Form Percorder Validate Binary Search Tree Kth Smallest Element in BST	https://betcode/ into://betcod	g min heap;
https://voutu.be/f6zlmtCv/UU https://voutu.be/f6D-12F0538 https://voutu.be/f6D-12F0538 https://voutu.be/f6D-12F0538 https://voutu.be/f5ibf07F83x https://voutu.be/f5ibf07F85x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f5ibf07F81x https://voutu.be/f8ibf07F81x https://voutu.be	Interval Linked List Matrix Matrix Matrix Matrix String Tring Tring Tree Tree Tree Tree Tree Tree Tree Tre	Meeting Rooms II (Lectcode Premium Rooms II (Lectcode Premium Rooms II (Lectcode Premium Rooms III) (Lect Cycle in a Linked List Merge Row Sorted Lists Merge Row Sorted Lists Remove Nth Node From End Of List Remove Nth Node From End Of List Recorder List Set Matrix Zeroes Sprint Matrix Rotate Image Word Search Longest Substring Without Repeating Longest Repeating Character Replace Minimum Window Substring Valid Anagaram Valid Anagaram Valid Anagaram Valid Parentheses Valid Palindrome Longest Fability Company Palindrome Longest Control England Longest Lo	table. Placetode, we care about the points in time where we are starting/ferding a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time, for each meeting check if a prev meeting has finished before curr started, using black and the points in time, for each meeting check if a prev meeting has finished before curr started, using black and the points in time, for each most from one bits not be other with the points of the points of the points of the point of the points of the points of the point of the points of	g min heap;
https://youtu.be/fst.im/t/vylU https://youtu.be/fst.im/t/vylU https://youtu.be/fs0_L2E0538 https://youtu.be/fs0_L2E0538 https://youtu.be/fs0_L2E0538 https://youtu.be/fs1@fs1PiR8xc https://youtu.be/fs1@fs1PiR8xc https://youtu.be/fs1@fs1PiR9xc https://youtu.be/fs1@fs1PiR9xw https://youtu.be/fs1@fs1PiR9xw https://youtu.be/fs1@fs1PiR9xw https://youtu.be/fs1@fs1PiR9xw https://youtu.be/fs1@fs1PiR9xw https://youtu.be/fs1@fs1PiR9xw https://youtu.be/fs1@fs1PiR9xw https://youtu.be/fs1@fs1@fs1PiR9xw https://youtu.be/fs1@fs1@fs1PiR9xw https://youtu.be/fs1@fs1@fs1PiR9xw https://youtu.be/fs1@fs1@fs1PiR9xw https://youtu.be/fs1@fs1@fs1PiR9xw https://youtu.be/fs1@fs1@fs1PiR9xw https://youtu.be/fs1@fs1@fs1PiR9xw https://youtu.be/fs1@fs1@fs1PiR9xw https://youtu.be/fs1@fs1@fs1PiR9xw https://youtu.be/fs1@fs2@fs1 https://youtu.be/fs1@fs1 https://youtu.be/fs1@fs2@fs1 https://youtu.be/fs1@fs2@fs2 https://youtu.be/fs1@fs2@fs2 https://youtu.be/fs1@fs2 https://youtu.be/fs2 https://y	Interval Unked List String	Neeting Rooms II (Leetcode Premium Rovers a Linked List Detect Cycle in a Linked List Merge Wo Sorted Lists Merge Wo Sorted Lists Remove Nth Node From End Of List Recorder List Set Matrix Zeroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Longest Repeating Character Replace Minimum Window Substring Walid Anagram Group Anagrams Valid Parentheses Valid Paintforme Longest Repeating Character Replace Minimum Window Substring Valid Anagram Group Anagrams Valid Parentheses Valid Parentheses Valid Paintforme Longest Repeating Character Replace Minimum Depth of Binary Tree Same Tree Invert/Filip Binary Tree Same Tree Invert/Filip Binary Tree Substring Paintforme Paintforme Tree Construct Binary Tree Substree of Another Tree Construct Binary Tree From Peroder: Validates Binary Earch Tree Rth Smallest Element in a 85 Lowest Common Ancestro of 85T Implement Trie (Prefix Tee)	https://betcode. we care about the points in time where we are starting-inding a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using this control of the count	g min heap;
https://youtu.be/f6zim/tCv/UU https://youtu.be/G0_L2F0538 https://youtu.be/G0_L2F0538 https://youtu.be/G0_L2F0538 https://youtu.be/Midich956u0 https://youtu.be/Midich956u0 https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv6fu0https://youtu.be/Sv6fu0https://youtu.be/Sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/Sv6fu0https://you	Interval Uniked List Matrix Matrix Matrix Matrix String Tree Tree Tree Tree Tree Tree Tree Tre	Neeting Rooms II (Leetcode Premium Rovers a Linked List Detect Cycle in a Linked List Merge Two Sorted Lists Merge Row Sorted Lists Remove Nish Mode From End Of List Remove List Martin Remove List Martin Remove Linguist Repeating Longest Repeating Character Replace Minimum Window Substring Valid Anagram Group Anagrams Valid Palendrom Congest Palindromic Substring Palindromic Substring Palindromic Substring Longest Palindromic Substring Endogram Decode Strings (Leetcode National Palindromic Substring Palindromic Substring Endogram Decode Strings (Leetcode National Palindromic Substring Palindromic Substring Palindromic Substring Palindromic Substring Sinary Tree Level Order Endogram Tree Substring Palindromic Substring Sinary Tree Level Order Endogram Tree Substree of Another Tree Construct Binary Tree Level Order Endogram Tree Substree of Another Tree Construct Binary Tree Home Tree Substree of Another Tree Construct Binary Tree Home Tree Substree of Another Tree Construct Binary Tree Home Tree Substree of Another Tree Construct Binary Tree Home Tree Substree of Another Tree Construct Binary Tree Home Tree Substree of Another Tree Construct Binary Tree Home Tree Substree of Another Tree Construct Binary Tree Home Tree Substree of Another Tree Construct Binary Tree Home Tree Substree of Another Tree Construct Binary Tree Home Tree Substree of Another Tree Construct Binary Tree Home Tree Substree of Another Tree Construct Binary Tree Home Tree Substree of Another Tree Construct Binary Tree Home Tree Substree Office Tree Substree Off	https://lectocols. we care about the points in time where we are starting-inding a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using this country of the state of the country of the c	g min heap;
https://youtu.be/f4t/mTCV/UU https://youtu.be/f4t/mTCV/UU https://youtu.be/f6t-EFE538 https://youtu.be/f8t-EFE7B538 https://youtu.be/f5t0-EFE5038 https://youtu.be/f5t0-EFE5058 https://youtu.be/f5t0-EFE5058 https://youtu.be/f5t0-EFE5058 https://youtu.be/f5t0-EFE5058 https://youtu.be/f5t0-EFE5058 https://youtu.be/f5t0-EFE5058 https://youtu.be/f5t0-EFE0058 https://youtu.be/f5t0-EFE0068 https://yo	Interval Uniked List Watrix Matrix Matrix Matrix String Tring Tring Tree Tree Tree Tree Tree Tree Tree Tre	Needing Rooms II (Leetcode Premium Reverse a Linked List Merge Woo Sorted Lists Merge Woo Sorted Lists Merge Woo Sorted Lists Remove Nth Node From End Of List Renove Nth Node From End Of List Renove List Set Matrix Zeroes Spiral Matrix Rotate Image Word Search Longest Substring Without Repeating Character Replace Minimum Window Substring Walled Anageram Group Anagrams Valid Parentheses Valid Parentheses Valid Parentheses Valid Parentheses Valid Paintonne Substring Pallindromic Substring Free Maximum Depth of Binary Tree Same Tree Invert/Fiji Binary Tree Maximum Path Sum Binary Tree Maximum Path Sum Binary Tree Maximum Path Sum Binary Tree Foom Preorder Validate Binary Search Tree Construct Binary Tree Foom Preorder Validate Binary Search Tree Construct Binary Tree Foom Preorder Validate Binary Search Tree Construct Binary Tree Foom Preorder Validate Binary Search Tree Construct Binary Tree Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Search Tree Construct Binary Free Foom Preorder Validate Binary Searc	Into Jicketoods. Vector about the points in time where we are starting/ending a meeting, we already are given those, just separate start/end and traverse counting run of meetings going at these points in time, for each meeting check if a perv meeting has finished before curr started, using blood of the country of the property vectors. A property vectors are considered to the country of the start of the country vectors. A property vectors are considered to the country of the start of the country of the	g min heap;
https://youtu.be/f6zim/tCv/UU https://youtu.be/G0_L2F0538 https://youtu.be/G0_L2F0538 https://youtu.be/G0_L2F0538 https://youtu.be/Midich956u0 https://youtu.be/Midich956u0 https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv56u0https://youtu.be/Sv6fu0https://youtu.be/Sv6fu0https://youtu.be/Sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/sv6fu0https://youtu.be/Sv6fu0https://you	Interval Uniked List Matrix Matrix Matrix Matrix String String String String String String String String String Tring Tring Tree Tree Tree Tree Tree Tree Tree Tre	Neeting Rooms II (Lectcode Premium Rooms in (Lectcode Premium Rooms in Lined List Marge Two Sorted Lists Marge Two Sorted Lists Remove Nith Node From End Of List Node Indiana Room End List Nith Node Indiana Room End	https://lectocols. we care about the points in time where we are starting-inding a meeting, we already are given those, just separate start/end and traverse counting num of meetings going at these points in time; for each meeting check if a prev meeting has finished before curr started, using this country of the state of the country of the c	g min heap;