2. Add Two Numbers

dont forget to set prev pointer->next to currN dont forget to reset carry to 0

---

4. Median of Two Sorted Arrays

lo<=hi

i is the element to the right of the division halfLen-i since i+j=halfLen. the +1 in numerator so odd total sizes get rounded correctly make checks that i is not out of bounds when doing . comparisons, not . j

---

5. Longest Palindromic Substring

edge cases are length 1, length 2, and remember to only change max checking both len>maxL and dp[start][end]==true

---

6. ZigZag Conversion

When you have a vector of vectors, with specific row/column access needed, initialize sizes as liberally as needed instead of getting stuck, figure out space optimization later if asked.
Do no struggle with complicated min of for loop limits, just set them so that relevant/changing variables don't go over their acceptable limits. You can use "&&" within the for loop terminating condition.

---

10. Regular Expression Matching

if there is a * , check two columns to the left and see if a true has occured in certain amount of vertical levels based on the char before the * and how many consecutive letters there are

---

15. 3Sum

i moves regularly, but with an . inner while loop which uses a two pointer approach. each iteration variable makes sure to make its specific val change on successful triplet find

---

16. 3Sum Closest

initialize closest to be the sum of the first 3 numbers, not just the first number itself

---

18. 4Sum

Make sure not to repeat index names in nested lips.

Make sure to use the correct index letter in nested loops.

You can sort numbers and skip consecutive same element for a given index, so as to avoid any repeat combos, instead of having to generate all string combos for any given set of indices. I.e. use sorting to prevent a repeat set of nums in the first place.

---

19. Remove Nth Node From End of List

safer to iterate until pointer is not null, than until when it's next is not null, unless you explictly need the last element

---

22. Generate Parentheses

just use regular char by char backtrack, keeping track of how many open there are to know if you can use a closed.

---

23. Merge k Sorted Lists

Always think about final answer when doing complexity analysis. In this case, N final nodes, each time there is a log k insertion/comparison into the heap. Similar to how we think of the memoization table for complexity analysis in dfs.

---

24. Swap Nodes in Pairs

remember not just to deal with anchor/doublePrev but also to make prev->next=next

---

25. Reverse Nodes in k-Group

create a dummyhead that points to head. when you get to a count%k==0 node, reverse exclusively between the next node and begin. remember to include first in the reverse function (which will point to end at the end), it will be the next begin. Also set the previous begin to point the the last prev. after the reverse, set curr to point to begins next

29. Divide Two Integers

similar to least operators expression q, bit shift as much as possible until can't, call again using remainder but same divisor don't bitshift the result of the recursion, it's already . in the correct form by definition

30. Substring with Concatenation of All Words

If needing to start at each index, maybe only need to start at first word.size() if each word same length, due to multiples aspect.

33. Search in Rotated Sorted Array

remember to not only consider where the pivot is, but also where we should search for the target given whether the pivot is in the right side or not, 2 situations. Don't forget to do the <= for the side that will be searched.

34. Find First and Last Position of Element in Sorted Array

for highest Point, make hi = nums.size(), and final answer should be lo-1. Otherwise same as lo Binary with conditions flipped but same internals within conditional blocks. It's just an arrow flip from .

37. Sudoku Solver

just use maps keeping track of whats available, forget "todo" don't forget to skip the cells that already have numbers (from the original config)

43. Multiply Strings

double for loop doing manual multiplication, i+j+1 is place in result, i+j is where carry goes

---

45. Jump Game II

bfs keeping track of furthest reachable node and not even considering neighbors that are below or equal to maxReachable. I.e. memoizing not just what's been visited, but how far has been visited to prevent checking anything under the limit.
Frontier set style bfs

---

52. N-Queens II

offset by highest negative amount to make col-row diag accesses positive

---

54. Spiral Matrix

easier to use a seen matrix and do a clockwise turn when something is out of bounds or has been seen before

---

56. Merge Intervals

remember to take the max of the ending of the last interval in the result with the ending of the current interval in the input

---

59. Spiral Matrix II

Don't forget to start 1 further away for last 3 directions

---

60. Permutation Sequence

Subtract 1 from nth permutation to get how many is left, just start with input k and do -1. That's how many you need to skip. Then don't have to worry about anything.
just test different indices as needed, one of them will work. Don't waste time trying to understand on some theoretical level.

---

66. Plus One

only change carry if sum is LESS THAN 10

---

67. Add Binary

sum%2 since its base 2

---

68. Text Justification

dealing with the last sentence is the biggest corner case, remember to differentiate the three main cases: 1 word, then last sentence, then regular

---

72. Edit Distance

remember to include replace+1 if replace is a permitted op

---

75. Sort Colors

remember that when encountering a middle value, only thing to do is move idx go up to <=rightDiv

---

78. Subsets

don't mess up base case idx condition

don't forget when inserting into a vec within a vec of vecs, need to do an access within the insert

---

81. Search in Rotated Sorted Array II

rotated array search: two branches: mid smaller than hi, mid bigger than hi third branch if duplicates: mid==hi first two branches: see if target can goto the obvious side, if not it goes in the other third: hi--

82. Remove Duplicates from Sorted List II

If you are stuck on a linked list problem, think about how to use more pointers as necessary, even if they won't be used. They are like spare variables.

84. Largest Rectangle in Histogram

If you want a stack to be evaluated for a max entry at the end of a loop, no need to recode the while loop, just iterate one past end of list and do a (i==heights.size() ||) condition so it's emptied out on the last iteration. Conditions are a good tool for avoiding duplicate code.

85. Maximal Rectangle

don't mess up () for containers vs pairs

when using monoq or any container, better to keep track of initial positions of data then running lengths (in this case the horizontal aspect), as running lengths need to be continously updated while positions stay true

87. Scramble String

Use indices and length(s) when doing recursive functions that involve passing in strings. They are easier to memoize, easier to calculate time complexity with, and much easier to transfer to dp with.
Only focus attention on most complicated parts when it comes to bugs, as in complex index calculations.

89. Gray Code

to generate sequence of 4:

gen sequence of 3 as left copy for right, but reversed append bit of 1 to the left end of right concat right to left and return

90. Subsets II

remember to add curr at beginning of subsets function

91. Decode Ways

instead of many "0" checks, only checks should be: include dp[i-1] if current is not 0 include dp[i-2 if current twoDig is between 10 and 26

96. Unique Binary Search Trees

don't need to do palindrome dp, can just do n^2 single array do because only the specific numbers used don't matter, only the AMOUNT of them used to the left and right of each potential root

99. Recover Binary Search Tree

only change prev in morris traversal after visiting.

in the decision of setting first and second, set second whenever there is a lack of correct ordering detected and first is already set
algo: 12345 swap: 15342

notice the first swapped element is previous to curr when the first lack of order detection occurs (prev->val>curr->val) and the second swapped is curr when it occurs. This is due to the nature of swapping a larger element with a smaller in a sorted array

113. Path Sum II

when leaf nodes are a pivotal part of the question, you must consider a termination check when reaching a leaf node

114. Flatten Binary Tree to Linked List

if doint it this way, pay special attention to tail handling, and remember to set root->left=NULL before returning

Alternatively, don't return a pair, and simply iterate to end of left side if left side exists, to get its tail

---

115. Distinct Subsequences

If you do a dp for how to reach a state, remember that all dimensions need to be not included, as in pre-reaching that state.

But really, for string dp and dp in general, don't do reaching state, do solving entire solution for that state as dp is supposed to represent

use an initialization L when any [i-1][j-1] is required

don't just fall back on the recursion formula when doing dp, just look at the dp table, and use logic about what happens in the different instances of that state (as in, how the positions in the inputs at those indices relate to each other), how that affects would it could arrise from.

when doing the initialization, just literally think about what it means. E.g. empty subsequence can be generated in one way by any usable input. But empty string input can not generate any target. Practice this, not too hard.

---

116. Populating Next Right Pointers in Each Node

make sure not to confuse next with right

---

117. Populating Next Right Pointers in Each Node II

bfs ending condition is empty group (necessary).

---

124. Binary Tree Maximum Path Sum

remember for the maxSumCandidate to only add the maxpath from the left and/or right if they are greater than 0. Also, max the maxPath with the current root's val for a similar prevention of less than optimal result

---

126. Word Ladder II

use bfs to make the graph: when exploring, if a child already was found, always connect it to current word, but if its not found, additionally put it in map and queue

---

129. Sum Root to Leaf Numbers

this is a rare case where it's better to check if the left and right child exist, and if neither do, add to total, since you need root to leaf sums.

---

132. Palindrome Partitioning II

2d dp to determine which substrings are pals 1d dp to determine minimum cuts up to i for everything to be a pal, then LIS like
always consider 1d before 2d

---

133. Clone Graph

use a map

---

134. Gas Station

avoiding a duplication of a curcular array is easy, just create a virtual index for when accessing is needed, which is just idx%sz

---

140. Word Break II

use memo and bubble up

can't really do memo with backtracking

---

143. Reorder List

reverse second half of the list (first half bigger if anything). Keep one pointer to the tail of the building list, one pointer to the smallSide of the rest of the list, and one pointer to the largeSide of the list. Make the current tail point to alternating sides, incrementing those sides and the tail accordingly

---

144. Binary Tree Preorder Traversal

push Null on to stack in iterative tree traversal, then just pop and continue when its at the top.

make sure to put actions in correct if/else branch

---

146. LRU Cache

make sure list Nodes are pairs of key and value, for erasing when capacity is reached

to get iterator to last element in list:

auto it = lruLst.end(); it--;

because lruLst.end()-1 is just a position in memory to be used in certain vector ops

---

148. Sort List

slow fast pointer trick is actually pointless time complexity wise here since you have to go through to break the list anyways.

---

149. Max Points on a Line

use a map for each point specifically, what slope occurs most frequently for a given point remember that you need a currMax because of potentially overlapping points

---

150. Evaluate Reverse Polish Notation

can only use switch for ints remember to call stack and list containers by their declared names, not "stack" or "list"

---

152. Maximum Product Subarray

dp maintaining high and low ending at a current number

155. Min Stack

just need one stack, no need to pop min elem as in max stack

156. Binary Tree Upside Down

must set new Node to be filled in later to something other than NULL, otherwise need a reference to a pointer

158. Read N Characters Given Read4 II - Call multiple times

don't overcomplicate the amt read form the read4 api. That is the prime source of error.

162. Find Peak Element

use a line chart when dealing with peaks/valleys or anything with a sequence of numbers where ups and downs might matter

164. Maximum Gap

the max gap will never be smaller than if the difference between max and min were evenly divided amongst all the gaps (max-min/(sz-1)); So if we make our buckets that size, we can be certain that the max gap will occur between buckets and not within.

166. Fraction to Recurring Decimal

map records index where answer to rem(r) starts. (result.size() means an insert at that will start after the end of the string at that point).

174. Dungeon Game

reason we start from bottom right is we know what it must end with, but not what it must start with. Always consider doing things in reverse
dp table is minimum health to reach end if starting from certain start point

end has a base case, beginning doesn't. dp must always start from valid base case, make sure to question your base cases
earlier limit can override later limit, but later limit cannot override earlier becuz you wouldn't get there to begin with

179. Largest Number

trick is in comparison, put number before and after the other

186. Reverse Words in a String II

reverse entire string, then reverse each word individually

188. Best Time to Buy and Sell Stock IV

since a transaction involves both a buy and a sell, make the [k-1] index assymmetric. On buying we want [k-1] since it's a new transaction, on selling we come from [k] since its the same transaction.
initialize holding at INT_MIN and not holding at 0

new transaction occurs when buying

200. Number of Islands

watch out for confusion of x and y, call it row and col instead

201. Bitwise AND of Numbers Range

be careful with taking substrings or subvecs of binary, convert to string instead

std::bitset<8>(128).to_string(); int ans = stoi(res,nullptr,2); //for binary strings

---

204. Count Primes

when doing the sieve and going up from a certain prime to cross out all its multiples, we ADD, not multiply, by i

---

207. Course Schedule

good example of directed cycle detection

---

210. Course Schedule II

you can do the top ordering from either the earliest prereq, or from the leaf. Prob easier to start from root and do outgoing map + indegree vec

---

212. Word Search II

don't forget to init isWord to false in Trie constructor

---

213. House Robber II

vector constructor from iterators has an exclusive end

---

214. Shortest Palindrome

we want the longest prefix palindrome, so we can just reverse the stuff to the right of our longest prefix palindrome and prepend it to the original.
the way to do this is to find the longest prefix of the original string which equals the suffix of the reverse, because that means that substring is equal to its reverse (and hence a palindrome), as the suffix of the reverse is equal to a certain length prefix, and it being equal means that prefix is equal to its reverse

to do this, append a hash followed by the reversed string to the original, and do kmp table gen on it, use the last value in the table as this represents the longest prefix equalling the suffix of the reverse, and then anything to the right of this prefix is just reversed and prepended to the original for the final result

215. Kth Largest Element in an Array

remember that you can use a heap of size k for a runtime of n log k, just keep the heap at that size

218. The Skyline Problem

use an ordered map of int:vector<pair<bool,int>> the int is either a starting or ending index, the vector lists all the starting and ending indices at that loc Process all of them, then pop from the heap while the top is invalid

221. Maximal Square

dp is to get the max side len, square it at the end

224. Basic Calculator

use a pair for parenidx and ops, second term being the idx in the string, so that you can determine if last op should be performed or if last paren came after it

need to check if last op was - before combining numbers

you deal with the - issue when you do an op, via applying the negative to the first number in the op, popping the -, and pushing a +

even when immediately performing a calc (as in not after encountering a ')'), use compressStacks

227. Basic Calculator II

just use currNum and push when you get to an op

229. Majority Element II

remember that in majority element problems, the "if cand1 Count == 0" condition should be in an else if, everything should be in an else if, with deceremented both being in the final else.

230. Kth Smallest Element in a BST

233. Number of Digit One

whenever you do a % and / series to get digits, DO NOT mutate original input

241. Different Ways to Add Parentheses

Don't forget the other main way to do recursion other than going from left to right and shortening the input that way, is to break it up in the middle at various parts and calc left and right results. This latter way is typically what ends up being dp solved with the dp span technique, and often the dp span with cuts technique (also used in min cost binary tree)

244. Shortest Word Distance II

When doing a traverse two lists/merge situation, increment the smaller pointer. Calculate min difference each iteration in while loop. No need to have post while loop checking, because the smaller one will have iterated to the end of its list while it was still smaller than the still ongoing list.

245. Shortest Word Distance III

transfer indices if word1 and word2 are equal

248. Strobogrammatic Number III

must have extra function to see if we can go up to and including the middle digit in odd or 1st left of mid when even

250. Count Univalue Subtrees

modify a becomesFalse variable, easier than maintaining truthness of a true variable, because its more active as opposed to preventative

---

253. Meeting Rooms II

heap contains rooms required, by current end time of room. sort the intervals. If an interval starts after the top of the heap, we can replace its time with current interval's end time, otherwise need to push on new end time as new elem on heap.
It's a min heap because the earliest ending time means the room is the one most likely to be free

---

254. Factor Combinations

use bt when it comes to avoiding combos

You can fill result during bt in a loop, doesn't have to be base case

don't always need base case in bt

at any point in the loop, if we have a factor, we can either add that factor and it's complement to the path and then push the path to the result, or we can recurse for that complement without including the complement in the path of course.
the bt avoid combos trick is that future nodes are always increasing, so each . combo path will only be increasing, eg: 1,2,4 2,4,6 4,6,7
notice that even though certain numbers will appear more than once obviously, each list is structly non-decreasing, and due to the logic of the for loop, they will all start with different values (increasing for loop). This is why we needed the [i-1]!=[i] trick in combination sum 2. Here we need the start trick to endure increasing nodes.
draw out what a path looks like, and what it should be. Easiest thing, ensure the path is accurate at beginning of function. Also, if you think something should be added to the path, add it. Just make sure it is as it should be for the next call.

---

255. Verify Preorder Sequence in Binary Search Tree

at each point, we want to find what is the parent of the current node. so we pop all values smaller than it. last value popped is parent, which becomes lower bound

260. Single Number III

think about bit being different between different numbers, and the xor between them will reflect that.

Remember that shift operators need an = to make them mutate

xor with 0 is identity

remember to do &, not &&

---

264. Ugly Number II

optimal approach: i,j,k point to the min index such that when 2,3,5 respectively are multiplied to val at that index, will produce num greater than nums.back(). if 2 * nums[i] is greater than last num, then surely 2 * nums[i+1] will be.

---

267. Palindrome Permutation II

do not iterate through a set or map using (auto elem:set) while simultaneously altering the container, copy it first.
Write full "else if" conditions, skimping only leads to bugs

---

269. Alien Dictionary

undirected cycle = visited set, if child is visited and !parent = cycle

directed cycle = both visited and rec set, rec set popped at end of func, if child is in rec set it's a cycle.
Visited is just used to visit all components e.g. if they are disjoint
top sort: visited set, add element to stack at end

combo of top and directed: add element to stack plush pop it from rec stack at end before returning false.

---

272. Closest Binary Search Tree Value II

dont assume stacks will magically finish processing, be methodical and maintain last value popped for each stack, and pop each stack while its not going in the right direction at the beginning of the while loop

---

273. Integer to English Words

under twenty, under hundred, and over hundred, for the 3 dig group part to get last two elements, do %100, to access first two, /10.

---

275. H-Index II

want to find left most point which statisfies citations[mid]>=sz-mid because number of papers is limited to how many papers exist at/to the right of an index, and we want the furthest left index as that will yield the highest amount of papers just set ans to what its supposed to be within the search

---

276. Paint Fence

for each color, we can put it at the end of all previous (dp[i-1]) combos, except for the ones that end with two of the same color. How many end that way? number of combos 3 ago (dp[i-3]) * (k-1/k) (all colors other than say red 3 ago, * 1 for only possib of last two being two reds). same logic for every color, and since they are evenly probable, we just mult by k

---

277. Find the Celebrity

every call to "knows" eliminates one person. First pass switches to non-eliminated person

alternative: push all to stack keep comparing last two on stack, then second pass is the same as first sol

---

279. Perfect Squares

like coin change make sure for loop conditions are in right order (first should be j<size

---

281. Zigzag Iterator

use iterators and a queue for iterator/traversal problems.

---

282. Expression Add Operators

keep prev num and curr num args for dealing with running totals in dfs problems involving all */+-

286. Walls and Gates

put all 0 seeds/buildings in q at the beginning

287. Find the Duplicate Number

think of the list of nums as a linked list with a cycle, each node corresponds to one val, so one node may correspond to multiple elements.
the entrance point of the cycle is the only node which has two other nodes point to it. Just as the duplicated number is the only type of element which is reached to by more than one other element (since there is more than one version of the duplicated element).

291. Word Pattern II

Use two maps, one for pattern char, and one for str substr, map to each other for loop through every substring starting from initial l, if neither pattern char nor substring mapped, map them and recurse.

295. Find Median from Data Stream

follow up, keep track of the frequencies, in an array or bit

296. Best Meeting Point

just collect the row elems and col elems in sorted fashion during traversal (traverse twice)

299. Bulls and Cows

keep a map of char frequencies in secret first go over for bulls first and decrement map then go over and count cows

300. Longest Increasing Subsequence

binary search solution:

building sequence essentially, don't focus on tails[i] bs, push_back instead to build the sequence

iterate through nums, then binary search lower_bound for where you can place nums[i]. If nothing is bigger than it, push_back: you increase size of list. Otherwise, replace iter somewhere in list, thereby increasing chances later number will pass through unhindered to end of list

302. Smallest Rectangle Enclosing Black Pixels

instead of binary searching for the max and min for each column by searching by row (which we can't do anyways as there might be a split), we do a binary search for the max and min of ANY col, by doing an iterative search through an entire column at a time. Same for rows.
So it's essentially a 2d binary search for the bounds of the rectangle, which actually works.

307. Range Sum Query - Mutable

to get sum between ranges, must subtract up to hi minus up to (but not including) lo.

308. Range Sum Query 2D - Mutable

make sure all accesses have a +1 to account for the bit needing to start at this

size of bit is row+1, col+1

only set the grid in update, don't init it to matrix, will screw up deltas during all the updates in constructor

309. Best Time to Buy and Sell Stock with Cooldown

no need to have buying come from same day as no stock, just have everything come from i-1

none is the 2nd day of cooldown. Cooldown is the day of selling.

310. Minimum Height Trees

need to do 3 dfs's One from leaf to find furthest Leaf one the from furthestLeaf One to find furthest leaf Two then from furthest Leaf two to get distances from it
simpler way: bfs the graph putting the leaves on layer by layer, processing a leaf by removing it from its neighbor's adj list, whatever's left at the end with 0 neighbors is a root

311. Sparse Matrix Multiplication

loop through elements of A, and then the 3rd inner loop is a corresponding row of b, adding to result at indices of row of A and col of B

312. Burst Balloons

the cut in the dp with cuts is if that is the last baloon to be popped in the group

consider every baloon outside current span to still be existing

each span is like: all baloons in span (but nothing outside of it) have been popped except for one, which would make a good last baloon to exist in the span

313. Super Ugly Number

there might be multiple primes that produce an equal next value, all of them must have their positions in lst iterated

315. Count of Smaller Numbers After Self

when doing index calc for inplace_merge: pass in first index and last of array make halfL = lo+(hi-lo)/2

remember if its the answer is in the form of an array, to add to the slots in the result rather than =

318. Maximum Product of Word Lengths

hash the bit representation of each word in a map, maintaining the max word length for any given hash. max of 2^26 possibs. Then compare each elem in the map to any other with an &.

319. Bulb Switcher

When the relevant issue is even vs odd toggling/flipping/switching of a given element, consider cancellations of pairs of factors.
write out at least 15-20 element examples or pattern won't be noticed.

Don't just look at amounts when writing out example, look at the relevant issue, as in which are odd vs even

321. Create Maximum Number

loop through each len for nums2, finding maxNum with that len (using stack), then maxNum for (k-len) for nums2, merge both, compare to max so far
when merging, when we get to a point where there are equal numbers, keep iterating . until one array ends or we get to a non-equal number, the one with the higher num is the one we want to take from

323. Number of Connected Components in an Undirected Graph

cycle is irrelevant, we only visit each node once anyways

324. Wiggle Sort II

we want to find the median, so we can partition nums greater than it on one side, and nums smaller on the other
dutch flag: swap with far: far++ swap with near: near++,j++ median elem: j++ one iterator starting at end, one at mid, both working their way down alternating

327. Count of Range Sum

for the merge calc step:

for each left, we will record its leftMost prefSum doable to be higher than lower, and rightmost prefSum doable for upper. We do the upper by iterating j in reverse for the right side

329. Longest Increasing Path in a Matrix

Remember that you don't need to record visited nodes if the sequence has some logic that prevents revisiting previous nodes in the sequence (as in its strictly increasing).
Also remember to use plenty of parens when doing bitwise comparisons

---

332. Reconstruct Itinerary

you can use array or map for visited instead of ints to prevent overflow.

If you need the sol in certian lexicographical order, than order the children in the map/graph in that same lexicographical manner, and exit the function as soon as a result is found as it will by definition be the best due to its ordering. Requires returning right after checking result of dfs if that dfs completed (return true).

---

335. Self Crossing

1st situation: 4th crossing 1st 2nd: 5th meeting 1st last: 6th crossing 1st

any other situation is extraneous as these are the only unique differences between curr and prev line.

Notice the last situation has some extra conditions beyond the intersection establishment to establish that it is in fact the situation we're describing. Another way to look at it is we establish floors on the elements we've already established ceilings for in the first two bools in the if statement.

---

336. Palindrome Pairs

You can iterate through indices, OR through size of words, sometimes the later is more convenient. Try it if get stuck.
for each word, we want to see if left part is palindrome, and there exists a match for the right part, and vice versa. we do this by iterating through lengths
every word is considered as being the longer or equal word as both prefix and suffix

---

341. Flatten Nested List Iterator

Use iterators for nested Lists. Make sure to advance iterators in stack. Try to use more if else instead of ifs with no elses. The if else helps prevent excessive edge case checks.

---

351. Android Unlock Patterns

Remember when doing the bitset place calculation, to have the place generator do/return the leftwise bit shift.
You can add any additional directional moves by extending dir1 and dir2 arrays with whatever movements are desired, e.g. L movements.

---

352. Data Stream as Disjoint Intervals

the vector insert function inserts right BEFORE the iterator passed to it.

don't forget if you're using an enum type flag to use int instead of bool

set hi = vect.size()-1

---

354. Russian Doll Envelopes

https://leetcode.com/problems/russian-doll-envelopes/discuss/82802/C%2B%2B-46ms-binary-search (https://leetcode.com/problems/russian-doll-envelopes/discuss/82802/C%2B%2B-46ms-binary-search)
sort by increasing height, decreasing width, we want to find the furthest place such that our envelop is wider than the prev but less wide than the one we're replacing

---

355. Design Twitter

bulk of complexity in generating feed. Use a heap to collect from all the users the target user is following (including itself), comparison function sorts them based on their counts (which you keep track of). They are dumped into a result most recent one first

---

356. Line Reflection

hash all the points, find the min and max by x-coordinate, median must be evenly between them, find if each point has a pairing based on this median

---

358. Rearrange String k Distance Apart

break when top.second==0 or pq is empty. then check if string size is correct don't forget to decrement count use a deque to keep hibernating nodes/pairs, no need to search them don't forget that for comparisons with container.size(), the container.size() MUST be cast to int

---

360. Sort Transformed Array

horizontal center: -b/2a

if struggling with comparison with center, just use solved equation outputs in the comparison

when can't think of concise coding style, just brute force the code

---

363. Max Sum of Rectangle No Larger Than K

for the bin seach on sums, insert 0 first (helps prevent edge case of same number being good), and search as the sums are being inserted (search before you insert current sum)

---

364. Nested List Weight Sum II

use dfs for nestedIntegerSum, don't use class with iterators the dfs iterates over the whole list each frame

---

369. Plus One Linked List

lines 29-32 missing is most common bug

---

371. Sum of Two Integers

two's complement addition inherently takes care of the negative bit so dont worry about it (think of what happens to the signed bit when the neg number is bigger vs the positive bigger vs both negative..there will either be a carry and it disappears or carry and it doesnt etc)

---

372. Super Pow

to find say a^5 mod base (a^3) mod base = x (a^2)(a^3) mod base = (a^2)x then (a^2)x mod base

377. Combination Sum IV

the opposite loop order of coin change 2

382. Linked List Random Node

resevoir sampling:

pick k numbers that you need then start from k+1, gen rand each time, see if rand%currIdx is <k, if so replace that pos

385. Mini Parser

when you reach a ']' pop the back and add it to the nestedInteger at the top behind it. Otherwise duplicate objects that have no relation to each other (when pushing to stack and adding to nestedInteger separate copies are created)

386. Lexicographical Numbers

use dfs with a currNum as input, simple for loop iterating through 10 numbers, and dfs into a 10 of currNum if possible

388. Longest Absolute File Path

/t and /n are each one character and can be checked with =='\n'

390. Elimination Game

keep track of lo and move it over based on whether going forwards or back and whether numsLeft is odd or even, delta will be pow(2,level)

391. Perfect Rectangle

there must be even number of overlapping corners for everything but the 4 far corners

392. Is Subsequence

for follow up, record all occurences of char then iterate through s, doing binary search in corresponding vector in map, finding minimal index greater than . low bound (which keeps increasing based on where we are in t)

393. UTF-8 Validation

don't do anything weird when converting from number to a vector of binary. Just put them in the exact places they need to be in from the get go using reverse indexing.

394. Decode String

when getting to an open bracket, just push the num onto the stack whenever pushing an alpha string onto the stack, use a stackPush helper that combines it with the top if possible

395. Longest Substring with At Least K Repeating Characters

Make sure you insert and delete the right things from/to containers, especially when the containers are for chars, as inserting an int will not be caught.
enforce a limit of unique characters when sliding window alone is not robust enough to deal with given problem, and iterate over 26

396. Rotate Function

move the highest coeff over to the left, just calc what its move does (it subtracts coeff*A[i] and adds sum-A[i])

399. Evaluate Division

when dealing with doubles, make sure you don't downcast to int accidentally anywhere in the program

do a control f for int

400. Nth Digit

figure how many numbers can be skipped at each digit range. if cant skip, figure which number to skip to, and what index within the number

402. Remove K Digits

don't overcomplicate positional logic when its not needed. Don't need to store positions here, next character is always assured to be one away from prev.

403. Frog Jump

forward dp, use an int,set map, as you can reach the same place in multiple ways from the same stone (if jumps are 1,2..).
don't ever make a set have type bool

405. Convert a Number to Hexadecimal

signed int already has the correct 2's complement bits, so just convert it to unsigned

406. Queue Reconstruction by Height

sort in ascending order of height, descending of pos, and keep track of available slots

407. Trapping Rain Water II

we keep the max boundary popped so far, because it represents the limiting boundary which surrounds any given remaining lower cell. Analogous to min of two maxes we do in trw1. I.e. it was at some point the minimum boundary around a cell, so it is a limiter, but less strict a limiter than potentially lower future boundaries.

We want to traverse in order of lowest boundaries so that when we get to a certain cell, the current maxH represents its lowest boundary since we would not have gotten to this cell if we didn't first start at the lowest boundary which then discovered it

---

410. Split Array Largest Sum

don't forget one less than partitions, and also 0 partitions is ok, just not less than 0.

---

411. Minimum Unique Word Abbreviation

just generate all target possibs, place them in pq (along with their lengths), and compare each abbrev to each word in the dictionary, returning the first that does not overlap with any of the words in the dict

---

417. Pacific Atlantic Water Flow

remember not to include corners twice, check for visited even during initial population of q

---

418. Sentence Screen Fitting

make sure not to increment inputs as opposed to variables

make sure to increment variables in the right order in while loops. particularly index in list, typically shoudl be last.

---

420. Strong Password Checker

use pq, del lowest mod of 3 first, so as to minimize number of replacements needed after deletions are performed (lower mod of 3 means after a deletion, replacements are more effective, e.g. going from 6 to 5, means 2 replacement vs 1)

---

421. Maximum XOR of Two Numbers in an Array

build trie, each num with 32 places. then for each num, iterate through trie finding its closest complement. see which has the highest xor amongs all nums and their complemets

432. All O`one Data Structure

use Nodes such that each node is a frequency group with a set of keys. map of keys to or list node iterator

433. Minimum Genetic Mutation

https://leetcode.com/problems/minimum-genetic-mutation/discuss/91604/C%2B%2B-two-end-BFS-solution-exactly-same-as-127.Word-Ladder (https://leetcode.com/problems/minimum-genetic-mutation/discuss/91604/C%2B%2B-two-end-BFS-solution-exactly-same-as-127.Word-Ladder)

435. Non-overlapping Intervals

can use a prev pointer instead of deleting from orig array

don't forget the case of previous interval having further endpoint than curr, must set prev to curr to get smaller last endpoint

437. Path Sum III

its like a prefix sum prob where you want to check if any contiguous section up to current node equals val

439. Ternary Expression Parser

use general indices when using stack and string parsing, as in index within container, as opposed to location in original string.
make sure to use while loop when crunching backwards, not just an "if"

don't forget to manage all your containers, for example container that includes relevant positions

442. Find All Duplicates in an Array

must push back during the traversal

443. String Compression

dont fast-forward to where the number should be placed, just start at the currently ending char and write its char and then the number

444. Sequence Reconstruction

in top sort using degrees, the graph is to parents, and the degree is for how many children. I.e. they have different types and opposite directions.

446. Arithmetic Slices II - Subsequence

vector of maps maintaining amount of 2 and over sequences of different diffs remember to add one to a map's diff for each j element visited (the 2 seq between any two numbers adds one to a map)

450. Delete Node in a BST

either put left subtree of target under the leftMost node of its right subtree, or vice versa

remember that hardcore in-place tree transformation should be done from the parent

453. Minimum Moves to Equal Array Elements

adding 1 to all elements but 1 is similar to subbing from just 1 element at a time

456. 132 Pattern

iterate from back to front, keeping a stack of s3 candidates. We will keep s2 in a variable. Remove elements from stack smaller than currNum, constantly replacing s2 with removed element

You don't have to think about what s3 is, we just want the best s2 possible. So pop off elements replacing s2, then put curr num on stack to maximize chance of high s2.

457. Circular Array Loop

Use a color code for each cycle keep several variables indicating goodness of loop just use trial/error to figure out negative number indexing: has to do with modding remember to convert size to int, better yet, just do it at the beginning typically

458. Poor Pigs

one pig can test one dimension, where a dimension is (minutesToTest/minutesToDie)+1. (it can test dimSz-1 of the items and then for last we know is the poison if it didnt die). So we need a n-dimensional cube of size ((minutesToTest/minutesToDie)+1)^p

459. Repeated Substring Pattern

repeated substring is s.substr(0,sz-table[sz-1]) as this is the part that is not matched within itself, it is matched by the kmp suffix so we just need to check that it is evenly divisible by the whole string length

460. LFU Cache

don't increment amount in a put function if you're only adjusting a value;

make sure you don't do certain operations before a quick return. For example, incrementing amount when a value is simply being changed in put.
just use a freq map of freq to list of key,val pairs , and a map of key to iterator of list of pairs

remember that keeping track of minFreq is easy given we know that put always sets the minFreq to 1

465. Optimal Account Balancing

must use backtrack, use two groups, borrowers and lenders

at each recursive frame, deal with current borrower, by iterating through lenders from current lender index to end, swapping ith lender with current lender index before and after dfs (backtrack part)

466. Count The Repetitions

we pretend the repetion block starts at the beginning of an s1 block for convenience in calculating after (if it truly started in the middle, it would still end up having the same at the end of the first rep block as at the end of the block where we noticed a repeat).

467. Unique Substrings in Wraparound String

keep track of longest substring that ends at each letter, then add up results for all 26 letters

468. Validate IP Address

remember the pattern of always starting the overall while loop after the last divider

don't forget to increment at end of while loop

in the inner while loop which accumulates a current string, you can place individual checks on illegal characters

469. Convex Polygon

Don't forget the sign must not change from one to the other, can be either depending on direction

471. Encode String with Shortest Length

palindromic dp with cuts(substrings here).

After checking the minimum for two sides with any given cut, collapse the substring(i,j) by finding if a prefix repeats itself throughout the string. Remember to pass dp into this function as it is the core string which will be wrapped.

474. Ones and Zeroes

don't forget to use knapsack for problems that involve selecting items from list. The alternative of set memoization should only be used when it has to.

480. Sliding Window Median

set comparison functions must not take equal elements lightly, make sure to have explicit tiebreakers

483. Smallest Good Base

unsigned long long is biggest

remember to write down confusing search/base statements

don't forget to consider the 2^0 + 2^1 ..etc. notation for base relationships. Helps with understanding how to convert to decimal and also what the limits are for what bases can be (if all 1's, just do the nth root)
use variables to solve an equation for a limit that is hard to intuitively figure out

484. Find Permutation

start the stack empty, but go 1 past the end of stack and place last number at stack at that point too

unload stack when you reach an I or end of string

can also just reverse all the numbers under D's, when string is placed correctly under and in between ascending sequence

488. Zuma Game

preOrder keeping track of best in reference, aborting certain paths when they exceed best (though best starts at -1, so that's an edge case). Use run length encoding for the shrink op. (pop the last pair in the pre string and then do stack op with the pre-encoding as stack and adding post encoding to it).

493. Reverse Pairs

can't get index of iterator of set in log n time

494. Target Sum

for target dfs problems, always go past end of array, with target equalling 0 being a good result

no map dp method: vector of vectors with colums going up to 2* sum, each el can come from bal below or bal above (i.e. treat it like a neg number below sum if you need to, only diff is the array index, not the context, so the targ then becomes sum instead of 0)

---

498. Diagonal Traverse

do an upwards traversal, storing each travel in a vector, and then do the downwards traversals. (going across column and then row or vice versa). Then alternatively concat these vectors from their 2 vector of vectors to the result

---

499. The Maze III

djikstra is bfs with weights

you can use memo in backtrack by checking and setting it before anything later occurs

---

502. IPO

sort projects by capital required

while current total>capital required for a project, add it to the heap pop max profit and add to total

---

505. The Maze II

should ideally use djikstras here

1059. All Paths from Source Lead to Destination

always dfs here, even if place has been visited

---

514. Freedom Trail

for getting distance between two indices both ways, either try each character being before the other (ends up being 3-way min, or do string.size()-(abs(diff)).

517. Super Washing Machines

only highest number is used as limiter because lower numbers are accounted for by the throughput calc. A higher number needs to unload one at a time, whereas a low number will be filled multiple times per step

525. Contiguous Array

hashmap keeping track of a certain property (mod, differential between 1's and 0's, sums encountered) is a valid addition to the window, two pointer, stack, heap, bin Search paradigm.
put -1 as index for before first index

don't need to add one to diff, remember that the count we find will be before the subarray starts

increment, check counts and update max if found, otherwise place

527. Word Abbreviation

group all the words by their max abbreviation, for each group make a trie, then query each word in the group to find the first place its unique and make the corresponding abbreviation for it
don't forget to check numWords at end of loop instead of beginning (root is always 1)

542. 01 Matrix

1. don't use condensed if statement when more than one thing is being done, to be safe.
2. always check for bounds in matrix dir problems
3. always check for correct <,> in bfs condition, otherwise infinite loop

545. Boundary of Binary Tree

the bottom level is the leaves go down left side as much as you can (sometimes will need to go right); same for right side (but put in deque); concatenate root to left side to middle to right side special case where only one node, as leaf is root

549. Binary Tree Longest Consecutive Sequence II

use bools such as bool leftInc = false; bool leftDecr = false;
bool rightInc = false; bool rightDecr = false;
also dont forget to max with maxIncr and maxDecr

---

552. Student Attendance Record II

Don't forget that O(n) dp can come from several levels behind.

If you have trouble seeing how a cell comes from a previous one that you know it derives from, see which previous ones that cell came from, and add up the appropriate ones. Essentially you can go an extra level back in dp during deduction using logic.
Simplest way to do table is just do every possible state, don't get fancy with potentially non-independent states even if .it seems it might work as that can get confusing.

---

553. Optimal Division

try out different ways to divide denom using the case of 3 nums

---

555. Split Concatenated Strings

all strings should def be their maximum except for the one within which the answer might start, in which case we need to test both versions and all possib starting places.
Be conscious of the nesting of loops, can be confusing.

---

556. Next Greater Element III

when having issues with edge conditions in traversal, consider going in the reverse direction.

for next permutation, find first a[j]>a[j-1], then find where in a[j] to the end the min element greater than a[j-1] is, swap, then reverse everything after a[j].

---

564. Find the Closest Palindrome

one extra consideration for palindromes below: if num-1 is all 9's (i.e. str size reduces by 1), return that

567. Permutation in String

just keep count of how many are off in the comparison

do char-'a' for array indexing

587. Erect the Fence

use a used set and check that points to be inserted into the hull are not in it C is prospective new leftMost point in the cross product, counterclockwise is pos in z direction
insert collinear into hull (pending not in used) before doing the nextTarget=start break check

591. Tag Validator

only open tags matter, check if open tag is a cdata start, or an open or closed

593. Valid Square

sort based on a tilted version of the square where every point has a different horizontal position, figure out how to break tibreaks based on vertical positions by looking at example of axis-aligned square.

600. Non-negative Integers without Consecutive Ones

zero smaller can come from treading with 0 if the current is a one, remember that

611. Valid Triangle Number

i,j,k technique: start at low end, move k as far as you can, once you can't move j: n^2

616. Add Bold Tag in String

Notes on kmp: main trick used in both main kmp and table building is pattern index is reset to either pattTable[j-1], or result[j-1] ,respectively. In both cases this is done only if j!=0, otherwise we just increase pattIdx/i, and set result[i]=0 for table build just prior.
Think of table building as finding a prefix that matches a current suffix, as in the prefix is the pattern and i/suffix is our text
make sure to not set initial elem in result vec of merge intervals until after the sorting if you need back to back intervals to be made into one, simply add an extra condition in your merging if branches

---

625. Minimum Factorization

generate factors from 9 downwards, reverse list, create number with high numbers at most significant places using powers and places

---

629. K Inverse Pairs Array

for dp matrix with confusing indices, just add 1 to the row and column sizes so you don't have to worry about off by 1 calculations, makes it easier
A good way to figure out complex sequence dp formulas is to write out the table, and experiment with what sequences will produce answers for different cells, and think about what would be added to those sequences in what manner, so as to produce the next sequence. typically just look one row above as that is one item less in the sequence
basically done

---

630. Course Schedule III

sort by end day, try taking a course, if can't take then replace it with longest duration taken so far which is stored in heap.
What if replacing earlier day makes it so we can take courses we dismissed? not possible:

(a,b) (e,f) (g,h) (c,d)

if we can't take e or g, and we get to c and can't take it but it's shorter than a, it will still have meant that a+c>d. so a+c will def be longer than b, which means we must replace it, there is no way to take both c and a. So we def can't take c with e or g and finish before d, as they are even longer than a

---

631. Design Excel Sum Formula

convert from any letter format to pairs and just work with those

toSums is from a cell to all the cumulative cells it contributes to sumCellTo is from the cumulative cell to its contributors, map because of overlapping ranges+weights

st.pop_back() in set because top element should not be summed from its elements but just set to value stack is not reversed because its already in order from end to beginning, no need to unload it into vector and revere, though itd be the same thing when traversing through stack, make sure to calculate each cell based on what leads to it, do not just use diff produced by first change, that is factored in by the actual cell change and percolates down

---

634. Find the Derangement of An Array

we can either place 1 (the first element in general in whatever we have left) at position x and have n-1 left, or swap 1 with what is at position x and we will have n-2 left. There are n-1 ways to pick position x given an array of n

---

635. Design Log Storage System

use a multimap instead of multiset+map

use the compare function when instantiating a set or multiset, and it will come into play during upper and lower bound.

---

640. Solve the Equation

always put parentheses after exclamation point comparisons

---

642. Design Search Autocomplete System

use a templated arg to pass in pq to a function

---

644. Maximum Average Subarray II

use binary search when answer is supposed to be a double

for the window part, keep two sums, a right sum which is just the sum up to the current index from the beg of array (but -targ each time we add el), and min leftSum, where left sum is up to just before our window, and also has -m for each el that is added to it. Then if the diff between right and minLeft>0, the differential of that span is positive relative to the average and it works

---

651. 4 Keys Keyboard

don't need to worry about buffer fullness (paste right after add), its always either add or multiply some number of times

---

659. Split Array into Consecutive Subsequences

1. Don't get stuck on greedy thought process, DP can be greedy and is easier to understand.
2. make array for use with map if map iteration with ordering is necessary.

Edit: may as well concatenate to older chains as they will def be long enough, whereas shorter chains may not become long enough. Keep track of count of elements, and decide if element can be appended to older sequence or start new ones, and change elements accordingly.
Keep track of how many sequences end right before certain elem so as to do this efficiently. if an elem can't be added to a prev chain, or start a new one, return false. if the count of a current elem is 0 (since we reduce counts ahead of currNum when we start a new seq), then just continue.

---

660. Remove 9

https://leetcode.com/problems/remove-9/discuss/176621/remove-0-9-solution (https://leetcode.com/problems/remove-9/discuss/176621/remove-0-9-solution)

---

665. Non-decreasing Array

you can either change the current element to be smaller or the next element to be larger. current element is limited by its prev element if i!=0, if that limit will not fix the non-decrease, must increase the next element as minimally as possible (to current element size). If we see another non-decrease then return false;

---

666. Path Sum IV

keep track of number of paths, and total up to it;

DON'T: do bfs/frontier set two group algo for creating tree

b * 2-1 to go from parent's horizontal pos to child's

---

668. Kth Smallest Number in Multiplication Table

bSearch is ALWAYS lo = lowest possible number in container, and hi= highest in container. Then find if the number is sufficiently large. Focus all energy on this issue, finding how many elements are below a certain number within the container, as opposed to complicated traversal thoughts.
for bSearch, lo<+hi, hi must = mid-1 or it will infinite loop

if getting which element in a grid or container of some sort is kth smallest using bSearch, use an enough function, checking if there are k elements less than or equal to mid. Lowest number that satisfies this. Conceptually, it will pick a number in the container, as only when mid is equal to a number that is kth smallest/biggest, is it the smallest solution that will have k below it.

---

670. Maximum Swap

use prev to get element before last

---

672. Bulb Switcher II

why n%2==0:7? For 0 flips we have 1 possibs, for 1 flip, 3 possibs, 2 flips we have 3

Can also do bfs to handle edge cases

---

673. Number of Longest Increasing Subsequence

easy

---

675. Cut Off Trees for Golf Event

make sure not to confuse curr with seed

---

676. Implement Magic Dictionary

you can directly change a letter in a string to another char via idx access

---

679. 24 Game

The result of a dfs/backtracking step can either be separate from the input list, or be directly put inside it. Modify argument by placing result, or keep them separate, two options!

---

681. Next Closest Time

remember vector cannot contain repeated elements

find with a vector is std::find

don't forget to make everything after the replacement the min num

---

683. K Empty Slots

a set is implemented as a bst, so use it when you want to do insertions to the middle based on sorted principle

---

684. Redundant Connection

---

685. Redundant Connection II

loops can be checked for by seeing if same nodes in edge are visited for second time, while iterating through edges
don't just rush to dfs, evaluate possibs. Either a node has two parents, with or without loop, or no two parents and a loop somewhere.
don't mess up a break in a dfs if statement

---

688. Knight Probability in Chessboard

divide by an eigth at each recursive step at the end

if doing dp, we want probability of knight being at each square at kth step, and so by adding them and /8, we know prob of being at another square that comes from several others

691. Stickers to Spell Word

use a vector of chars instead of a map for keeping track of letters in an advanced string problem, easier and consistent indexing (instead of having to compare keys between maps

692. Top K Frequent Words

nlogk for priority_queue: the log k is becuase there are at most k elements in the pq at a time. logk for inserting through k elements

694. Number of Distinct Islands

hashing via relative to top left works, just remember to use to_string

698. Partition to K Equal Sum Subsets

state top down dp, iterate through all nums each recursive call, don't forget to skip when current num puts currSum above limit

699. Falling Squares

just iterate over positions and check vector of overlaps for height of given position, dont need to use map

711. Number of Distinct Islands II

center a certain shape group on an anchor such as (minBottom,minLeft) make sure to sort each group after sorting to make it order-invariant use a regular set to store hashes of whatever type of container you want

712. Minimum ASCII Delete Sum for Two Strings

don't mess up indices ANYWHERE in dp calculations (including string indices)

713. Subarray Product Less Than K

the while loop in sliding window may require an (&& left<=right)

714. Best Time to Buy and Sell Stock with Transaction Fee

just remember that the selling is based off the current day, not previous

716. Max Stack

better to use ll and regular map with value being vec of iterators than a multimap

718. Maximum Length of Repeated Subarray

dp with two lists/strings = 2d DP

719. Find K-th Smallest Pair Distance

we don't want the numbers strictly below, we want the amount of numbers which covers our target difference, minimized.
in context: endpoint=either smallest possib distance(0), or largest (INT_MAX) so if each pair dist is post , if we want kth smallest, we want the 2nd away from the left endpoint, so need to get down towards it, and vice versa for largest: we could satsify the equality being too low and so need to search higher so we get snug with the 3rd highest lets say. I.e. you minimized dist from endpoint.

722. Remove Comments

use currLine, a type of comment state flag, and decide at end of line whether to push currLine to result depending on it current state not being block flag and currLine not being empty

723. Candy Crush

use read and write pointer column by column for gravity step. Read always moves (bottom to top), write moves when a number can be placed (is >=0).
pay special attention to how many zeros to place at the end (keep track of how many are placed/rowSz-negs, and then do rowSz-that for how many zeros to place)

725. Split Linked List in Parts

Do not get fancy with % operator in linked lists, unless you have certainty it works. Just keep counts, not that hard.
Really, just do two while loops. Inner while loop can increment just as well.

727. Minimum Window Subsequence

1. print matrix at end of every row/dp iteration
2. triple check dp statements, play it safe by checking row above, don't get cutesy with useless optimizations
3. use pairs for start and end instead of single indices if it helps
4. you only need dimensions number of checks to deal with dp bounds, i and j. May as well do them. (When filling in dp entry).
5. Initialize with -1 for indices type q's, 0 for max type q's

Edit: rows are T, columns are S, keep track of min window up to certain cell (ending at it), but also min result up to now, making sure to add 1 to the min each time before doing the comparison. Dp will just be dp[i][j]=INT_MAX if letter is not equal, otherwise 1+dp[i-1][j-1].second
Edit2: shortest span up to ith letter that includes up to jth subsequence, with the last letters not necessarily matching. Just have to add 1 if they dont match.

729. My Calendar I

https://leetcode.com/problems/my-calendar-i/discuss/173127/C%2B%2B-easy-to-understand-std%3A%3Alower_bound-solution (https://leetcode.com/problems/my-calendar-i/discuss/173127/C%2B%2B-easy-to-understand-std%3A%3Alower_bound-solution)

---

736. Parse Lisp Expression

central function in recursive parser is evaluate, which does a bit of parsing itself, but calls upon parse to know what next variable is or next balanced paren expression
remember to check expression[start] in parse, not expression[0], as it is possibly parsing somwhere within a string (whereas helper gets a properly beginning string always)

---

738. Monotone Increasing Digits

make sure to use proper < or > conditions, double check. make sure to use to_string on ints.

---

740. Delete and Earn

house robber, with thee extra check of whether prev element is one less than current, if it isnt, we can come from the max of the previous elem and use the current
make sure to use new array size as bounds not original

---

741. Cherry Pickup

dp is r1 c1 r2, two paths starting from beginning set default to INT_MIN, change result to 0 only when returning if needed

---

742. Closest Leaf in a Binary Tree

use bfs after graph

---

745. Prefix and Suffix Search

word = pop prefix=p suffix=pop

this is why we need suffix+"#"+full word

---

749. Contain Virus

for hardcore complicated bfs problems, there's not enough time for an elegant union-find, just do a brute force simulation with bfs's
a visited table can be useful in conjunction with coloring to differentiate different scenarios.

---

751. IP to CIDR

we use base 256, convert the input to a num. Count the trailing zeroes, and we can skip up to that many zeroes (by left shifting 1 or power of 2), removing the amount of skipped nums that equals, adding the string to result (with 32-count of zeroes used/1 shifted), and setting currNum to be the currNum+skippedNums.
We shift out num over to the right and mask it with 255 to get the string version for that block of 8 bits

---

754. Reach a Number

https://leetcode.com/problems/reach-a-number/discuss/112968/Short-JAVA-Solution-with-Explanation (https://leetcode.com/problems/reach-a-number/discuss/112968/Short-JAVA-Solution-with-Explanation)

---

756. Pyramid Transition Matrix

when making a new string out of characters, first declare and intialize string to "". Do not add to string during initialization, it'll mess it up.
Don't waste time with trying to set arguments to next DFS call in some clever generalizing way based on a given flag. Just write an if else and do the string copying. Make more succint later if it helps.

---

757. Set Intersection Size At Least Two

Sort by endpoints

Remember instead of keeping track of multiplicity of every interval, we can just keep a max1 and max2, and see if our current interval's start point is beyond either the max2 or max1 and act accordingly

---

759. Employee Free Time

Line Sweep:

NOT about sorting create a map Create event which is a time, and open or closing place event in map keep a count of how many open as you iterate through map.

---

761. Special Binary String

consider the different mountains, find them, sort them. Remember that each mountain is recursively optimized not including its endpoints, which we know will be '1' and '0' respectively.

---

763. Partition Labels

store position of last occurence of any given char

---

764. Largest Plus Sign

cannot consider result at tail ends of plus, all sorts of edge cases, must do so in the middle

can do either four vecs, or four different runs through the grid.

---

431. Encode N-ary Tree to Binary Tree

Don't pass in parents when dealing with anything other than graph cycle detection. You use current Node, and create new children nodes, make neew relationships accordingly, and call them with the original tree's node and the new node

---

427. Construct Quad Tree

don't skimp on index calculations, be very exact. must calc mid for row and col separately, and add half diff to start vals.

---

558. Quad Tree Intersection

if one of the nodes is a leaf, don't need to recurse, can definitely return either the leaf if its true, or the other node (a false leaf will have no effect on the other node)

---

590. N-ary Tree Postorder Traversal

for postorder stack trav, use deque for both pre and post, push all children on at once

---

770. Basic Calculator IV

Term class is mainly a list of variable strings (strings like "temperature" are why we need a list and cant just use one string, not every var is a char). Para is the coefficient.
A number will have an empty varLst, and only have a para.

Main terms functions are plus and times. plus simply adds paras, times multiplies them and inserts second varLst in first
Expression is mainly a vector of Terms. Main functions are mult,add,and shrink. Add appends second term list to first. Mult does foil on both lists. Shrink is done after both. Shrink sorts the lists(calls toString and then sorts after each terms currStr is filled), and places them onto a growing result (stackish), combining terms if appropriate.
Main string functionalities are Term's toString which fills currStr with lexSorted varString (no coeff), and Term's output which includes para, and Expression's final output which sorts by var length and lex sorts to break ties and discards terms with 0 para
Don't forget to use a while loop for the precedes part when an op is encountered (consider 10-0-10, must use while)

---

773. Sliding Puzzle

don't do pointless optimization like keeping track of how many are good. you'll find out that it is undone by other things. I.e. don't waste time on average case complexity. In this case, still have to memoize board which still makes obverall O(RC * (RC!))

---

774. Minimize Max Distance to Gas Station

when the problem states the answer needs to be within a certain precision range: that strongly suggest binary search, and the way you deal with it is (lo+diff<=hi), as in the range between lo and hi needs to be no larger than diff at the end.
when you don't know how to modify lo and hi, due to them not representing indices but just numbers in a range, you set them equal to mid instead of mid-1 or mid+1.

---

775. Global and Local Inversions

all global inversions must also be local inversion for them to be equal. So if we find a global inversion which is not a local one (2 away is bigger than min up to point) then return false;

---

779. K-th Symbol in Grammar

each successive state is the previous state plus the flipped prev state just track which index an index will be in N-1 (same index if less than sz/2, otherwise K-sz/2)

---

781. Rabbits in Forest

if doing a certain calculation, even a var+1 many times, just do it once and use it everywhere else so there is no forgetting of using it correctly

---

782. Transform to Chessboard

Two main observations: each row should be either equal to any another row, or inverse of it (and same for col). The other is that there are either equal number of 1's and 0's in row/col, if even board, or diff of at most 1.
In line with the first observation, every rectangle with 0,0 as the top left must have an even number of 1's for corners(so as for the rows/cols to be equal or inverse). Note that if every other col and row satisfy this property with respect to the first row/col, then they implictly satisfy it with respect to each other. Another way of looking at it is we are considering every bottom right point for an L embedded in the first ranks, as our guides. i.e. if we know the first row and col are valid (which we check), then if every bottom right can validly match, then every point works together to make the board valid.

we either want to move 1's away from the odd positions or towards them. We assume we want to move them away from the odds, on the assumption there are more 1's than 0's, and we want to move the more frequent element in the given row or col, to the evens (as there are more evens).
If the board size is odd, then there might be more 0's then 1's in the first row or column, in which case the assumption to move 1's away from odds was incorrect, and we do odd_positions-1's counted, as that is how many odd positions contain 0's (which should be moved).
If the board is even, we try both scenarios (moving 1's away from odds vs moving 0's away from odds) by taking the min of both. If we fix both the first row and column, we've constrained the board to its final position and thus all will be fixed (given its a doable board).
The reason for counting 0's in first row and col, is that if it is an odd board, we want to know if the assumption of more 1's (and therefore 1's should be moved from odds) is correct.

---

785. Is Graph Bipartite?

use same vec that holds colors to store visited

if the node has already been colored , return whether it equals its already color

color nodes one at a time

---

786. K-th Smallest Prime Fraction

don't need set to record visited, just make initial pq placements comprehensive in their ability to be futurely iteratable

---

788. Rotated Digits

we iterate through the digits, considering that we've firmly locked all prev digits up to current in place. Then, for any number under current value at digit, we do all possibs (7 * pow(sz-i-1) - bad possibs (3 * (sz-i-1)) , where bad possibs are the ones such that they only use 0,8,1. We also keep a flag of whether the number already contains one of the good vals, in which case we shouldn't subtract all bad possibs since there aren't any at this point

---

790. Domino and Tromino Tiling

reality based dp-initialization

---

792. Number of Matching Subsequences

If you are replacing a vector (value) in a map, make sure you store all new values in a "new Bucket" and set that key equal to the new Bucket, even if it's empty.

793. Preimage Size of Factorial Zeroes Function

5's are the limiting factor, we want number of 5's up to a certain factorial, divide by increasing number of 5 powers within a binary search

795. Number of Subarrays with Bounded Maximum

at each elem, add maxim number of starting spans that could be. we want number of elems since one that was too high (this sets bound on how many combos will be added). we want number of elems since an acceptably high elem (also acceptably low). do numsSince too high - numsSince high enough and add to total.

798. Smallest Rotation with Highest Score

use opening and closing events

799. Champagne Tower

you can add to the row ahead whilst iterating on the prev row just set the top cup to equal the amount of pours instead of simulating every pour

802. Find Eventual Safe States

the recStack hold all cyclic nodes, thats how it works (if it didn't, non-connected comps would be an issue)

803. Bricks Falling When Hit

1. To find size of component, keep track of it in the parent. when in makeUnion, just add size of both and set each parent's count equal to that
2. Don't waste time thinking about whether cell was included, just do a check befroe and after to see if it was anchored, and if there was a change from un to an, than know to subtract it (since it's not supposed to be include).
3. can also keep track of whether something is anchored by looking it up in the parent.
4. N*Q * A(Q,N)
5.

---

805. Split Array With Same Average

use equations to figure out rules (averages must be equal to average of whole array), do subset sum over different lengths to see if an array can achieve that avg

---

808. Soup Servings

there is unnevenness because a can serve more than b per serving

---

810. Chalkboard XOR Game

N=total xor of all elements remove one element: N= N^elemRemove only way for this new N to be 0 is if elemRemove = N so Alice just has to avoid picking that elem. so only if all numbers are the same, would she not be able to avoid this if all numbers are the same and the number of elements is even, the the total is 0 and she's already won if all numbers are same and odd number of elements, that player will lose as next removal sets total to 0

---

813. Largest Sum of Averages

painters partition algo with slightly different rec relation

---

815. Bus Routes

1. see if there are shortcuts to making graph (sortable lists anywhere?)

2. graph nodes should be created with new, added to a vector for easy access during children addition
3. just use delete on pointer, iterate through vector.

---

816. Ambiguous Coordinates

edge cases aren't that much, just return empty vec if first and last chars are 0

if first char in half is 0, only one result possible

---

817. Linked List Components

this looks like a union find, but can just be solved by keeping track of current group, since unions are only formed sequentially

---

708. Insert into a Cyclic Sorted List

1. You can always leverage the head node to determine if a cycle has been achieved yet, no need for two pointers or keeping an extra list (just check "==head"), keep a cycle variable
2. sometimes not only is null-noded list an edge case, but so is single-noded list.

---

825. Friends Of Appropriate Ages

use map to keep count of ages since it is a limited range, remember to subtract 1 from within product for same age

---

827. Making A Large Island

once you color code an island, it is essentially marked as visited. If a cell is 1, that means it is both unvisited and the next cell to visit

---

829. Consecutive Numbers Sum

don't forget to use arithmetic series everytime there is a consecutive number sum situation

don't forget to use a float for a sqrt limit

---

834. Sum of Distances in Tree

root tree at 0 do a regular bubbling up postorder traversal for getting below scores and below nodes do postorder traversal (calculating curr's score in result, and then doing same for its children which can check the result vector which contains its parents score); simply do N-curr's below to get nodes outside of N from its parent outside

---

835. Image Overlap

hor and vert shift is also n^4, but it is not as efficient for sparse matrices as hashing of deltas

---

489. Robot Room Cleaner

think about where the robot is when it ends its last recursion, make sure it can get back to present in an upright manner

---

837. New 21 Game

for sliding window, ALWAYS first do the removal of left end, then calc of current result, then any incrementing or addition operations.
when a dp result depends on previous results, consider whether sliding window can linearize solution.

when doing probability of reaching certain state(s) problems, dp is very useful, as dp[i] is the probability of reaching state i, and it can be produced by summing the probabilities of ways to get to dp[i] * the connections, as in (Prob of getting from i-1 to i)***dp[i-1] + (Prob of getting from i-2 to i)***dp[i-2] etc.
As in, do not overcomplicate by trying to figure out total number of goal possibilities /total number of possibilties. Unless perhaps it's a prob of who wins in a two player game problem.

---

839. Similar String Groups

stop messing up the lookup function in union find, make sure the recursive step is : n->parent = lookup(n->parent)
because if you do lookup(n) it will infinitely loop

841. Keys and Rooms

you don't need think of current path, becuase it's clear path doesn't matter much, it's about whether the rooms are one connected component. Use a single visited and keys set for all of them and just dfs what hasn't been visited.
don't overcomplicate dfs for connected component problems\

842. Split Array into Fibonacci Sequence

you can use a try catch with stoi

always enforce for loop limits when it makes sense. It usually prevents bugs

use backtrack with reference variable if it will lead to less comlication.

843. Guess the Word

best guess should be the one with least amount of 0 matches, since that is the worst case scenario (causes least amount of eliminations since 0 is most common

847. Shortest Path Visiting All Nodes

Don't forget to set the original bitwise cover for a node to include that starting node as having been visited.

Changing cover is an |, not an &, don't forget that

848. Shifting Letters

S[i] = ((S[i] + num) - 'a') % 26 + 'a';

do it this way, get the overall sum, then difference with a, then %26, then add back to a

850. Rectangle Area II

two Maps: one map for a currently active interval and its multiplicty one map for each height mapped to vector of events

delete from the multiplicty map when something reaches 0 use const after compare fucntion in struct passed to ordered map

at each height, convert multiplicity map to a merged span vec and return the length of all the spans added together

add at beginning of loop

---

851. Loud and Rich

1. Use less variables that need to be altered, even at the expense of verbosity!
2. Make sure all nodes are created, even if not present in one of the lists.
3. Make sure you thoroughly understand whether smaller or larger values are desired
4. Make sure the relationship between parent and children is not inverted. crucial.
5. use an id set to keep track of created nodes

---

853. Car Fleet

sort cars as pairs, iterate back to front putting them in stack

---

855. Exam Room

use a set to store intervals

how to deal with a seat being removed that is next to another seat? Keep in mind that l>r for (one or 2) of the intervals created within seat in which the seat is next to another. So all you must do is have getD return -1 when l>r so it gets sorted to the end in the set. no extra work needs to be done during leave, as the maps implictly will hold the correct info

---

857. Minimum Cost to Hire K Workers

wage/quality as the limiter.

sort elements by their wage/quality ratios, least to greatest iterate through these ratios, we know that every element before the one we are up to has a lower ratio and therefore the current element will be the limiter. Keep elements up to and including now in a heap, keeping track of the sum and popping . when heap size surpasses k (popping highest quality). multiply the sum by the current . element's ratio

861. Score After Flipping Matrix

1. make sure first column is all 1
2. for the rest columns, flip it if 0's is more than 1's.

---

863. All Nodes Distance K in Binary Tree

when converting from container A to a container of user-defined type, make sure you are iterating through container A, otherwise nothing will happen

---

710. Random Pick with Blacklist

purify the first N-len(B) by mapping to the last B as needed (if a num is bklacklisted in the last B nums just dont use it to map to one of the first N-len(B)

---

864. Shortest Path to Get All Keys

hash based on state (bit string of keys and current loc)

---

865. Smallest Subtree with all the Deepest Nodes

If getting compiler issue with NULL as pointer, cast it to pointer

can't end non-void function on "else if", just have some logical sureness and end on an "else"

don't doubt the effectiveness of NULL node as base case, return a result that will surely be disregarded, and you can add an else if condition to check if both results could only have come from null nodes

---

866. Prime Palindrome

next palindrome making: get the first half string to at least the midway iterate For i from (int) halfStr through (int) halfStr + 1 suffix = i if original len was odd, reduce suffix length by 1 cand = i+suffix.reverse
this will yield two candidates, take the smaller of the two which is still above or equal to original

must increase currNum 1 in while loop, as you may get same output for nextP forever for same number

870. Advantage Shuffle

map B's ranks to original indices, sort A and put it in deque, iterate through ranks placing in appropriate slot if largest in Deque is bigger otherwise use smallest

871. Minimum Number of Refueling Stops

drive along stations, if gas reaches negative, pop from maxheap until gas is above 0. Only add a station after the while loop of popping

470. Implement Rand10() Using Rand7()

think of a 7x7 grid, reject anything over num=40 (where num=7 * row + col)

872. Leaf-Similar Trees

can't use bfs because it does layer order, you need left right order

873. Length of Longest Fibonacci Subsequence

just see how far you can go from any two starting points, checking for existence of nextNum in set of all nums

878. Nth Magical Number

lo<hi, return lo

if its a min binary search then hi=mid on both >=, and for < do lo = mid+1

879. Profitable Schemes

feedforward dp is necessary here to deal with maxP

528. Random Pick with Weight

seperate out equals in this bin search don't put 0 at the beginning, just search for first val greater than pick pick is just rand()%sums.size()

519. Random Flip Matrix

must replace the val at the picked location in the map with the val at the bound loc if it was picked, or just total-1 if it wasnt. Direct virtual array corollary

497. Random Point in Non-overlapping Rectangles

we need to add +1 in area calc so # points are proportional to rects

880. Decoded String at Index

constantly modK with size to maintain the index in the current string, also maintain current size of current string (which is everything less than i)

881. Boats to Save People

if you can't sit the heaviest person with the lightest, the heaviest will definitely need his own boat. use two pointers to heavy and light

882. Reachable Nodes In Subdivided Graph

with djikstra you add to visited map upon visiting the node, not during adding. Also need to check when you pop node of pq, that you haven't been there yet. This is because it doesn't know when adding to the q if that is the best choice, but it knows when popping that is the min distance to that node, so only then do we mark it as visited.

885. Spiral Matrix III

when doing the spiral matrix problems, consider both the gap/dist on the outside and dir on the inside, and vice versa

In this case, it's a while loop, iterating through the current dirr, with increments on dir and halfStep and count

887. Super Egg Drop

the dp relation is a min of the max between two states depending on whether egg is broken when dropped from certain floor. Those two states have clear relation to growing or decreasing as floor increases, binary search based on which state is bigger

889. Construct Binary Tree from Preorder and Postorder Traversal

use the same number for both the pre and post partitioning (if you know num of elements you can segment post with that knowledge)

891. Sum of Subsequence Widths

for subsequences: to determine if order matters or not, check if its a max/min type, or one in which it must be ordered for some reason. If order doesn't matter, sorting helps.

when you need to add or subtract to a cumulative total during an iteration, and its adding or subtracting a formula at a time (Total + (max in seq-min in seq)), you can just contribute the individual elements in the formula, which can perhaps reduce time compelxity

2^i = how many sequences with curr elem as the max, 2^sz-i-1=with curr as the min

895. Maximum Frequency Stack

its ok to leave numbers at the end of the freqMp locations they were previously at, as they still exist at that frequency to be popped later.

897. Increasing Order Search Tree

remember to set root->left to NULL

just do inorder

---

898. Bitwise ORs of Subarrays

frontier set is like bfs in a way.

Make sure to have 3 sets: the result set, the prev set, and the curr set. Do not modify the curr set while iterating through it

to know what values can be created by ors with current val, we look at all vals that can be generated up to the preceding element (hence frontier set)

---

902. Numbers At Most N Given Digit Set

must iterate from left, how many can be generated from each index onwards if all to the left are the same. So how to deal with situations in which we can't keep all to the left the same? Simple, can no longer increase our total maintaining this property, and so simply exit loop/return answer

---

903. Valid Permutations for DI Sequence

For a D, it can only go from same rel rank to up to one lower than total amount of letters up to now

DIDD 103 2 when we have an extra number to use, we can increase all prev numbers so as to accomodate a lower number, but we cant decrease them, because we're coming from less numbers that were already small as possible note sequence prior to last must reflect numbers that were available, and THEN you consider boosting them given curr numbers https://leetcode.com/problems/valid-permutations-for-di-sequence/discuss/196939/Easy-to-understand-solution-with-detailed-explanation (https://leetcode.com/problems/valid-permutations-for-di-sequence/discuss/196939/Easy-to-understand-solution-with-detailed-explanation)

---

906. Super Palindromes

nextP involves a >=input

---

907. Sum of Subarray Minimums

figure out leftEnd and rightEnd for every element using stacks.

From there, the actual count is #(j) = (j - i + 1) * (k - j + 1), as there are j - i + 1 choices i, i+1, ..., j for the left index of the subarray, and k - j + 1 choices j, j+1, ..., k for the right index of the subarray.

---

909. Snakes and Ladders

just use num in your q and you only need one way conversion

use a map to keep track of lengths

---

910. Smallest Range II

after sorting, there is a point at which all elements to the left of it add K, and all to the right subtract K. keep track of the min and max element at any point in time given this info.

---

911. Online Election

don't forget that to know the max candidate after a certain period of time based on vote events, you don't need a heap, you just need a hashmap + basic max maintenance.

---

913. Cat and Mouse

50 50 3, the third param is whose turn, we use 3 for ease (1 and 2)

keep track of what states have been colored using a matrix, also degree of states (how many of a state's children have been colored differently than the state's turn and therefore the state can not be determined from that child)
populate queue with final states, color accordingly. for each front node on q that is processed, determine its parents, color them if their turn matches the color and place them on the queue, otherwise decrement their degree. If a parent's degree reaches 0, color it as a loss and place it on the queue.
In this way the queue only contains resolved colored states. Meanwhile, the degree matrix is the only data structure that maintains information for uncolored,unqueued states

---

916. Word Subsets

don't forget that when constructing the B map, to only update it with a val if the current string in B has character count greater than what's in the b map

918. Maximum Sum Circular Subarray

use a min kadane's algorthm, subtract from total sum as one candidate vs the other one being kadane's

use prefix sums and monotonic queue over double array, remember whenever sliding window seems bad you can try monotonic queue, as it adds smart pruning, in this case we keep increasing prefix sums, as we don't need earlier prefix sum with higher sum
can also use "next array" max of leftSum and rightSum

remember its max of min and max kadanes, and dont need to do circular, these make it so you dont have to do that

919. Complete Binary Tree Inserter

maintain nodes with no left or right in a q

920. Number of Music Playlists

we memoize number of ways to reach l songs with j different songs having been played

then, for dp[l][j] we can consider number of ways for last song to be unique (N-(j-1)), plus number of ways for it to not be unique (dp[i-1][j]max(j-k,0)), j songs to choose from, minus the last k
edit1: not enough focus on j

923. 3Sum With Multiplicity

distinct(n)

remember to deal with the cases of 3 diff nums, 2 diff nums with both dirs, and 1 num

927. Three Equal Parts

remember to compare the groups at the end to see that they have the same ordering of 0's and 1's

928. Minimize Malware Spread II

make the graph whilst not including the infected nodes make a map of infected node to components(parents), simultaneously keep track of how many times a component is infected
iterate through infected nodes adding the component sizes to its result only for the components which are infected once
return the node with the highest total uniquely infected component size total, or if there is a tie the lowest of the nodes.

---

930. Binary Subarrays With Sum

remember to only increase your pSum map after adding to result (because we only base result off number of previously seen sums, not including current)

---

932. Beautiful Array

First initialize an array to have all the numbers from 1 to N. Pass into recursive helper function. In the helper, un-enmesh the odd and even indices, and pass each into the helper, than concatenate.
Reason is, by separating the odd indexed elements from the even-indexed ones, none of the odd indexed ones which were equidistant between even-indexed ones are still equidistant between them, so we've reduced the equidistant situations by half, and we can recursively keep doing so until the base case of one.

---

936. Stamping The Sequence

when matching the stamp in reverse, a full match at any time is the best you can do. If you can't do a full match, it won't work. Make sure to skip by stampSz when you find a match at a given iteration through target. Also make sure to return no match if there are only 's over a stamp's location over target

---

940. Distinct Subsequences II

remember you need additional prefix sums array to keep track of sum up to a letter

---

943. Find the Shortest Superstring

use number (bit representation) to represent states.

dont forget this line (coming from not having traversed anything) if (prev==0) dp[i][j]=(int)A[j].size();

---

950. Reveal Cards In Increasing Order

simulating on indices works because it lets us know what position we'll be looking at/will be at the top of the deck after each iteration

---

952. Largest Component Size by Common Factor

Don't forget to go up to and equals the sqrt for factorization

don't forget to add val into the factor set after the loop, it may be prime. Also, if list is empty, add it in even if it's 1.
don't need to double loop for unioning components with union find: if all other nodes are parented to the first node, the component is already made correctly
Thus, can just keep the latest val for each factor in the map, and union curr number with that val, and then replace the val with curr Num (unioning things one after another)

---

954. Array of Doubled Pairs

sort use a map storing indices for each val pop from back of map pair's vector, record visited in an array skip element if visited

---

955. Delete Columns to Make Sorted II

keep a set of row indices which represent a "cut" right before that index (and after the previous), indicating that that index is already lexicographically greater than the previous one, and we don't need to compare it.
Keep a seperate set of new cuts to add, and only insert it into the overall divSet if delCol=false.

---

956. Tallest Billboard

record max height that can be reached at any give balance at a certain rod. Can either transfer from balance above, from above and to the right, or from above and to the left. only add to total if we are putting on the right side (and this coming to the left)
when using the 2* sum trick, all that matters is the initialization of dp[0][sum] to 0 or true as opposed to INT_MIN

---

958. Check Completeness of a Binary Tree

Do not access root in place of curr.

---

959. Regions Cut By Slashes

remember that the union find technique is to mate whatever cells can be mated at any given point in time, be liberal with it. For example here, split into 4 triangels per cell, and union whichever in current cell can be unioned, and then union to the right and below.

---

964. Least Operators to Express Number

base case is 1, remember that

---

968. Binary Tree Cameras

post-order: if a node isn't covered and has no parent, or it's children aren't

---

972. Equal Rational Numbers

geometric series sum for repeating part remember you have to move back the result for the repeating part by the repeating len, and then by the decimal size before it
geometric series is 66/100 + 66/10000 etc
repeating part: factor out the number. so : 66*(1/100+1/10000...), calc result of series and mult by the number and then mult/divide by how far it is move along from the decimal.

---

974. Subarray Sums Divisible by K

When dealing with number of contiguous spans, options are not just sliding window and dp, but also a map of prefix sums or prefix amounts of some kind.
Look at line 11 for dealing with neg rem

975. Odd Even Jump

consider using a map for searches

upper_bound-- to get highest node less than or equal to

982. Triples with Bitwise AND Equal To Zero

can't uniquely reverse & operation, many &'s can = 0, so must iterate through map

983. Minimum Cost For Tickets

You can insert 0 into the input vect to keep it in line with your dp vec

another dp pattern: iterate back as in LCS, but using dp val from one before curr j, as in the inner loop val is to check a stop condition in orig vec

984. String Without AAA or BBB

do not shy away from double coding symmetrical situations.

986. Interval List Intersections

for the mergeIntervals function, make sure Interval B is passed as having an equal or later start, so as to lessen number of conditions needed.

988. Smallest String Starting From Leaf

when something must start at a leaf, or a leaf plays central relevance, must use flag to determine if it was a leaf and do the base condition at the end

---

990. Satisfiability of Equality Equations

Can't have an object be a value in an unordered_map, must be a pointer

Use a map with pointer lookup when you need constant back and forth between a val and its node in a graph

---

991. Broken Calculator

keep dividing by 2 in reverse while its greater than add back the difference

---

992. Subarrays with K Different Integers

at most (k) - at most(k-1) . similar to product not more than

can also think of it as establishing left vs right bounds for each endpoint with two sliding windows

---

995. Minimum Number of K Consecutive Bit Flips

flip whenever we encounter a zero that needs flipping (and store in queue), which we determine based on the size of our flipped index queue

Note that this is only O(n), because the most we have to process an index is twice, once when we go over it, and once when we remove it from the queue when it expires

---

996. Number of Squareful Arrays

remember to divide by factorial of count freq for repeated numbers as they can be arranged in multiple ways

difference between no memo and memo impls: no memo uses a map which is by value, so no potential for duplicate combos, whereas memo uses a visited num which thus requires the graph to be by actual element index as opposed to by value

---

998. Maximum Binary Tree II

the construct function is basically in order traversal, so just keep going to the right until you see a spot where the node can be placed

1000. Minimum Cost to Merge Stones

algo is dp i j k where k is the number of piles to achieve for a span of i to j. We want one pile to the left and K-1 piles to the right.
remember that in dp with cuts, it's not always doing something to a middle entry, rather it can be separating the left from the right. these are the two possibs in dp with cuts
reason its one pile on the left is because say we wanted two piles on the left and one on the right, it would already be considered when our left was further back and we were considering two on the right

1001. Grid Illumination

diag trick: ascDiags go left to right ascending, are sums, descDiags are diffs

don't need to worry about what the diags look l ike, just remember hashing with no abs for sums and diffs, that's it. Don't need to know how it looks.

1007. Minimum Domino Rotations For Equal Row

a candidate must occur sz times in A and B combined (but only count it once if its in both A and B at the same slot). See whether A or B can be converted more easily.
It's true that you can have two candidates, but if you consider it, only one cand needs to . be evaluated to get the min as they'll both involve the same number of flips (consider one side starting all 2's, the other all 7's, any swap will create equal distance for either cand being present in all of one row)

1058. Minimize Rounding Error to Meet Target

get floor of all and ceil of all sums to check how many round ups and round downs we need. Use two pqs of mins of each type of rounding. remember to use integer var to keep track of how many 2.000 type cases there are and not put them in pq (they don't need rounding)

1010. Pairs of Songs With Total Durations Divisible by 60

sum of two numbers with mods a and b equals a number with mod a+b. So just do twoSum. edge case of a or b has 0 remainder, store it as 60 in the map

1012. Numbers With Repeated Digits

repeated number or entries in array, use factorial

1015. Smallest Integer Divisible by K

a remainder can be used in place of a value as the value, if our only interest is the future value of a remainder.

1016. Binary String With Substrings Representing 1 To N

just use the string itself instead of a q

1067. Digit Count in Range

number of times a digit appears in certain loc is different numbers that can be on left * diff numbers that can be on right.
for say 1274 number of times 3 appears in 10's place is: 12 * 10 + 1* 5 + 11 * 10 + 10 * 1 (0-11 on left with 0-9 on right, and 12 on left with 0-4 on right) (- : through)

for 0 we do pre-=1, since we can only start at 1 on the left side

1017. Convert to Base -2

decimal to octal: while !0 n%8 n/=8 res = string(rem) + res

octal to decimal: (just use standard coeffbase^0 + coeff2&base^1 as you would for base 2)

base = 1; while !0 n%10; total += (base***rem); base***=8

to negativeBase: if rem is neg, make it pos increase n by 1, to make up for the loss in remainder in the qoutient remainder equation

1022. Sum of Root To Leaf Binary Numbers

remember to update the value of currN right at the beginning if root exists, since once it is determined to be a leaf it will instantly be returned.

1023. Camelcase Matching

Remember there are two different character matching styles between a pattern and a larger word. One is the match must be contiguous, in which case either an n^2 or KMP approach is needed, other is that the match need not be contigous, in which case a two pointer (one in each word) approach is what is called for.

1031. Maximum Sum of Two Non-Overlapping Subarrays

for prefix sums make it 1-indexed init dp array to 0 and only update entries if they are reasonable two call structure for either 2 possibilities or overlapping/cancelling out possible algo
Edit: first go through left to right and then right to left with either L or M, keeping track of best contiguous sum up to that point. Then go through with the other length, checking left and right for what the best result of the other length was, doing several maxes

1035. Uncrossed Lines

Remember to consider all dp patterns for complicated search probs

1. LCS
2. knapsack (both varieties
3. spans and span cuts

1039. Minimum Score Triangulation of Polygon

you take the first and last element of a span as the base of a triangle, as they are connected by a side, and allow you to have all the spans that contribute to the sum contained within the current span (dp[i][k]+dp[j][k]

1038. Binary Search Tree to Greater Sum Tree

you can experiment with traversals as needed, i.e. right, left, curr in this case

1037. Valid Boomerang

remember equal slopes with one point in common means same line

1048. Longest String Chain

remember to sort the words first

1072. Flip Columns For Maximum Number of Equal Rows

rows that can be made equal by column flips are either complementary or equal. Hash by their state if they started with 1.

1073. Adding Two Negabinary Numbers

sum can be negative carry should be reversed each time as it means something different between consecutive indices so if sum<0, carry should be 1 (negative carry turns into positive for next index) etc.

1080. Insufficient Nodes in Root to Leaf Paths

you can do postorder and root to leaf path simply by passing in sum to function.

1081. Smallest Subsequence of Distinct Characters

same as 316. use a stack, pop letters off the back while they are larger than curr char and there are going to be more of them in the sequence. if the curr char is present (using set), continue. But first thing to do in loop is reduce count of curr char

---

1096. Brace Expansion II

how does handleBrace work? it's role is to parse into seperate exprs by comma, so only if bal is 0 when there is a comma should that be pushed into result

---