**379. Design Phone Directory**

Design a Phone Directory which supports the following operations:

get: Provide a number which is not assigned to anyone.

check: Check if a number is available or not.

release: Recycle or release a number.

**355. Design Twitter**

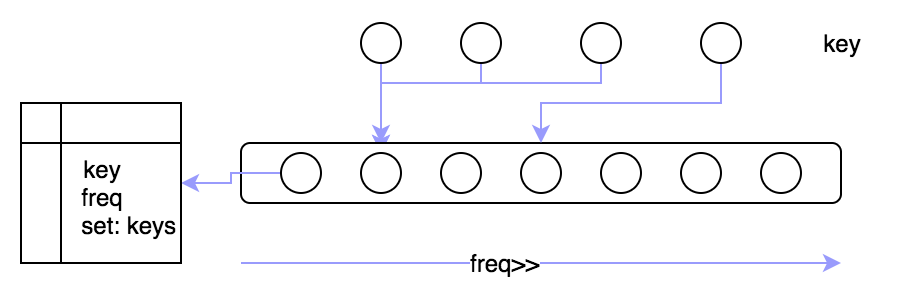
**146.** [**LRU Cache**](https://github.com/fu11211129/Algorithm/blob/master/ds-design/google/LRUCache.java)

**Follow Up: Distributed LRU**

We can maintain a hash-table that maps resource (it could be key of KV pair) to corresponding machine. Then, when a request comes in for , it will be directed to its corresponding machine, say . At the, it does the same thing as the single LRU cache strategy.

**Consistent Hashing**

**460. LFU Cache**

****

**432. All O’one Data Structure**

**Intuition & Algorithm**

**380.Insert & Delete Get-Random O(1)**

**381. Insert & Delete Get-Random O(1)**

**Design Rate Limiter**

For example, no more than 100 requests per second.

For example



if a request comes at 0.81s, it seems that the local , but we should allow this request, why? because we only care how many requests comes in a window of 1 second, we don’t need to care how request distributed in the 1s window. So a sequence of request of should be allowed under the limitation of 5 qps.

**359. Logger Rate Limiter**

**636. Exclusive Time of Functions**

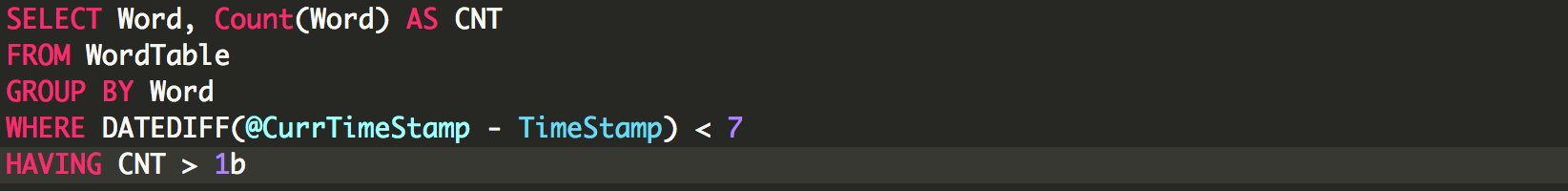
Solution:

**D**[**esign PokemonGo**](http://www.1point3acres.com/bbs/thread-287893-1-1.html)**.**

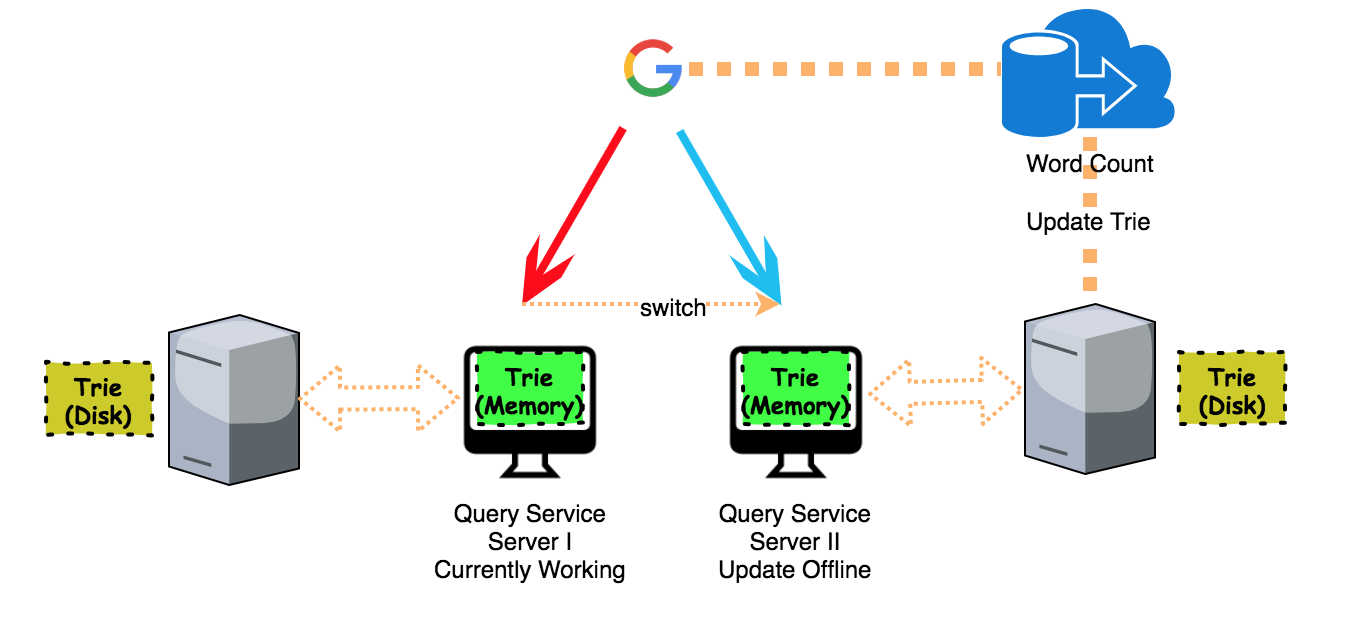
**Design Type-head**

Workable solution: when user types a prefix, say into google search textbox, it automatically makes a request to the, and the scan the char by char in the Tire. At node of “m”, there is a list of nodes, say which is associated with it (“am”) are the top 10 hot words starting with prefix .

Since the hot words will be changed every day, so we need to update the trie built so far with the most up-to-date searched words or logs, thus the comes into use for this purpose. This service works at servers to ask data from search logs which are stored in database, doing kind of like sql-queries thing:



At this stage, **Data Collection Service** maintains a “bigtable”, then it begins to update the trie. Because we cannot interrupt a working **Query Service** machines, we can copy the trie from disk to disk, and load the trie into another machine’s memory. Last, we update the trie using the **most up-to-date hot words big-table.** After it finished, we can switch the type-head query flows goes into the updated-trie machine.



Scalability: hire multiple Query Service machine

For example, the word has been search 20b times, then we need to update it to trie four time (because ). We need to update . Here, we can compute hash value for all the four prefix, and use or consistent hashing to target the query service machine.

**Storage for search words log**

For example, a word will be counted only if its occurrence is more than .

# Reservoir Sampling

**382. Linked List Random Node**

**398. Random Pick Index**

**336. Palindrome Pairs**

Given a list of unique words, find all pairs of distinct indices such that is palindromic string.

Solution: is palindromic string.

# Dynamic Programming

**238. Product of Array Except Self**

**Minimum number deletion chars** **ForUsAll**

making the all characters in ascending order after delete. For example, , delete minimum number characters

**10. Wildcard Matching**

Q. ‘?’ matches 1 character and ‘\*’ matches 0,1, multiple characters.

**K Sum Problem**

**Intuition & Algorithm**

Defineas the number of options of selectingnumbers from the firstnumbers that sum to. So for thenumber, there two options can be selected. Option, not selecting thnumber, so. Option #2, selecting the number, so

**68. Text Justification Uber**

Solution: Greedy Algorithm, fit as many words as possible per line

**Follow Up: Balance the amount of Empty Space Allocated per line**

For example, when justify the text with the maximum 6 characters per line, there are at-least two ways of justification.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

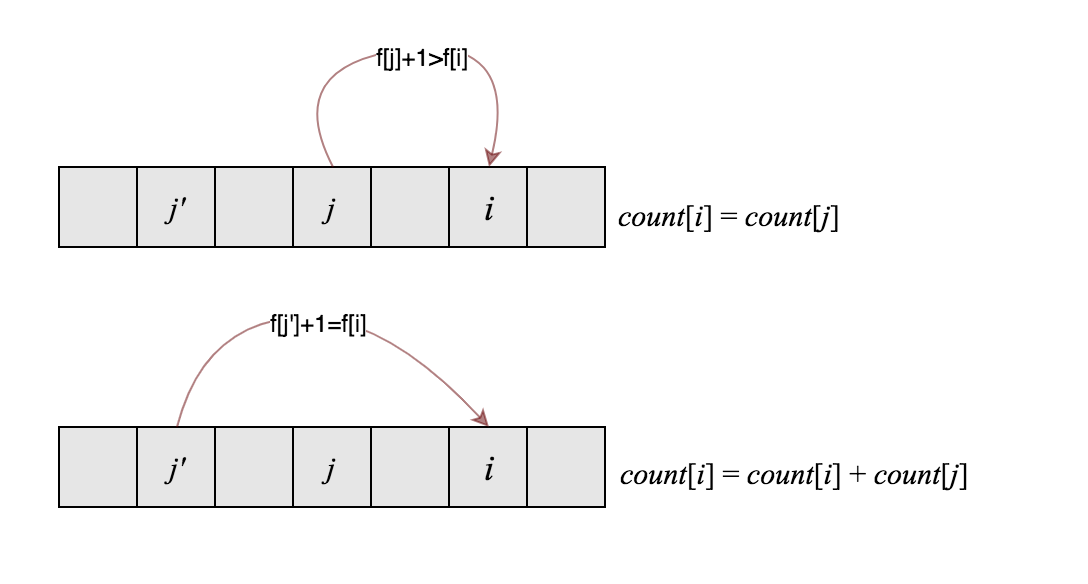
The cost of empty space of first plan is: , while that of second plan is , obviously, the second plan is better meaning the empty space allocated per line is more balanced.

**Intuition & Algorithm**

Dynamic Programming, first we compute costs of all possible lines in a 2D table , The value indicates the cost to put words from to in a single line where and are indexes of words in the input sequences. If a sequence of words fromtocannot fit in a single line, then is considered (to avoid it from being a part of the solution). Once we have the table constructed, we can calculate total cost using following recursive formula. In the following formula, is the optimized total cost for arranging words from .

**673. Number of Longest Increasing Subsequence** Facebook

Given an unsorted array of integers, find the number of longest increasing subsequence.

****

**Intuition & Algorithm**

Apply DP method by definingas the LIS ending up with indexAlso, defineas the number of LIS ending up with index

**10. Regular Expression Matching**

**Intuition & Algorithm**

‘.’ Matches 1 character,

**97. Interleaving String**

Given three strings,determine ifis formed by interleaving

**Intuition & Algorithm**

**5. Longest Palindromic Substring**

Solution I: Time Complexity:, Space Cost:

Solution II: Time Complexity:, Space Cost:

**516. Longest Palindromic Subsequence**

Given a string s, find the longest palindromic subsequence's length in s. You may assume that the maximum length of s is 1000.

**410. Split Array Largest Sum**

Split the array intoconsecutive subarrays s.t. the largest subarray’s sum smallest.

**Intuition & Algorithm**

Binary-Search by initializing the range. Then, each time picking up the middle value as the, and testify whether we can split array intochunks with the largest sum equals to the. If we can, we continue test smaller largest sum by. Otherwise,

**53. Maximum Subarray**

**Constant Space Solution**

**Follow Up: Maximum Subarray with At-least Length of K**

**691. Stickers to Spell Word IXL**

**42. Trapping Rain Water**

**62. Unique Paths & 63. Unique Path II**

**174. Dungeon Game**

**64. Minimum Path Sum**

**474. Zeros & Ones**

Find the maximum number of strings that you can form with given and. Each and can be used at most once.

**Intuition & Algorithm**

Defineas the max number of strings that we can form with given and. Apply “01” backpack problem solver.

**322. Coin Change**

Find minimum amount of coin change making up total amount, each coin can be used multiple times.

**Intuition & Algorithm**

Apply multiple backpack problem solver.

**279. Perfect Square Google**

**472. Concatenated Words**

**140. Word Break II**

**139. Word Break II Google Uber Twitter Snapchat Dropbox**

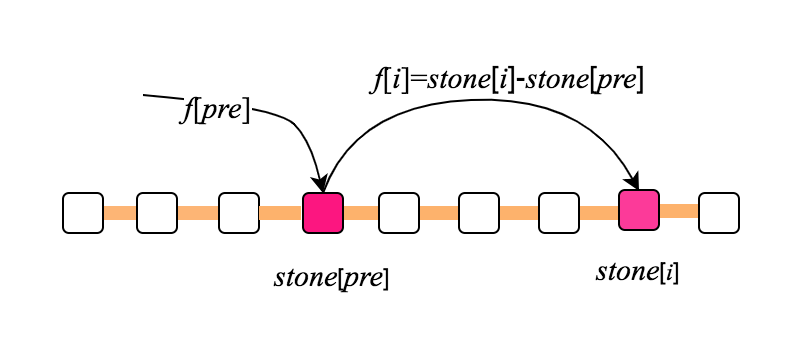
**131. Palindrome Partitioning**

**132. [Palindrome Partitioning II](https://github.com/fu11211129/Algorithm/blob/master/dp/PalindromePartitioningII.java)**

**416. Partition Equal Subset Sum eBay**

**403. Frog Jump Snapchat**

Given a list of stones' positions (in units) in sorted ascending order, determine if the frog is able to cross the river by landing on the last stone. Initially, the frog is on the first stone and assume the first jump must be 1 unit. If the frog's last jump was k units, then its next jump must be either k - 1, k, or k + 1 units. Note that the frog can only jump in the forward direction.



**Intuition & Algorithm**

The idea here is to maximize the jumps reach to, so we defineas the maximal jumps reach to.

**376. Wiggle Subsequence**

**120. Triangle Google**

**368. Largest Divisible Subset**

**Stock Problem Series**

**121. Best Time to Buy & Sell Stock I**

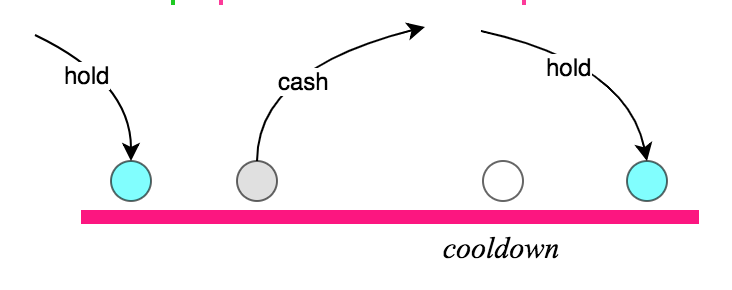
**122. Best Time to Buy & Sell Stock II**

Find all ascending subarraysuch that

**123. Best Time to Buy & Sell Stock III**

**188. Best Time to Buy & Sell Stock IV**

**309. Best Time to Buy & Sell Stock with Cooldown Google**

****

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

At the end of theday, we maintain, the maximum profit we have sold the stock at-hand, and, the maximum profit we could have if we owned a share of stock.

**Constant Space Optimization**

**198. House Robber I LinkedIn Airbnb**

**213. House Robber II (Rob in a Circle)**

**337. House Robber III (Rob in a BT)**

**70. Climbing Stairs Apple Adobe**

Apply the same formula as Fibonacci equation, can use fast exponentiation matrix algorithm.

**72. Edit Distance**

**115. Distinct Subsequences Google**

**87. Scramble String**

**95. Unique Binary Search Trees I Snapchat**

**96. Unique Binary Search Trees II**

**304. Range Sum Query 2D – Immutable**

**308. Range Sum Query 2D - Mutable**

**Intuition & Algorithm**

[**Fenwick Tree**](https://github.com/fu11211129/Algorithm/blob/master/fenwicktree/google/RangeSum2DMutable.java)

**Time Cost:** per update or get sum operation.

**Intuition & Algorithm**

Compress the 2d matrix into 1d array, then apply the algorithm the same as “find subarray with largest sum”.

**Time Cost:** for initialization,for update andfor get-sum

**303. Range Sum Query Immutable**

**307.** [**Range Sum Query Mutable**](https://github.com/fu11211129/Algorithm/blob/master/fenwicktree/google/RangeSum1DMutable.java)

**Mutable**: Segment tree ()

**312. Burst Balloons Google Snapchat**

**Intuition & Algorithm**

Add two balls (value=1) to the head & tail of ball list. By definingas the maximum score can get from. Suppose the last balloon to be removed isthen we have such formula:

**354. Russian Doll Envelopes**

Solution: Sort envelopes by width, then by length if two envelopes share the same width

**435. Non-overlapping Intervals**

Sort the envelopes/intervals list (first length, then width), then apply LIS mode algorithm.

**300. Longest Increasing Subsequence**

**221. Maximal Square** Facebook Apple Airbnb

**674. Longest Continuous Increasing Subsequence** Facebook

**651. 4 Keys Keyboard** Google Microsoft

(A, Ctrl-A, Ctrl-C, Ctrl-V), print as much ‘A’ as possible

**Intuition & Algorithm**

Defineas the maximum number of A’s printed withsteps. Assume we already know how many A’s can be printed with the best strategy insteps, namely. It is clearly that the current best strategy is to collect all the A’s already have (Ctrl-A), then copy all of them (Ctrl-C) and finally paste them back to the screen (Ctrl-V) one time or multiple times. So it takes at-least 3 steps to finish this series. For example,e,then the best strategy here is Ctrl-A, Ctrl-C, Ctrl-V, Ctrl-V. As a result, we can got.

**650. 2 Keys Keyboard** Microsoft

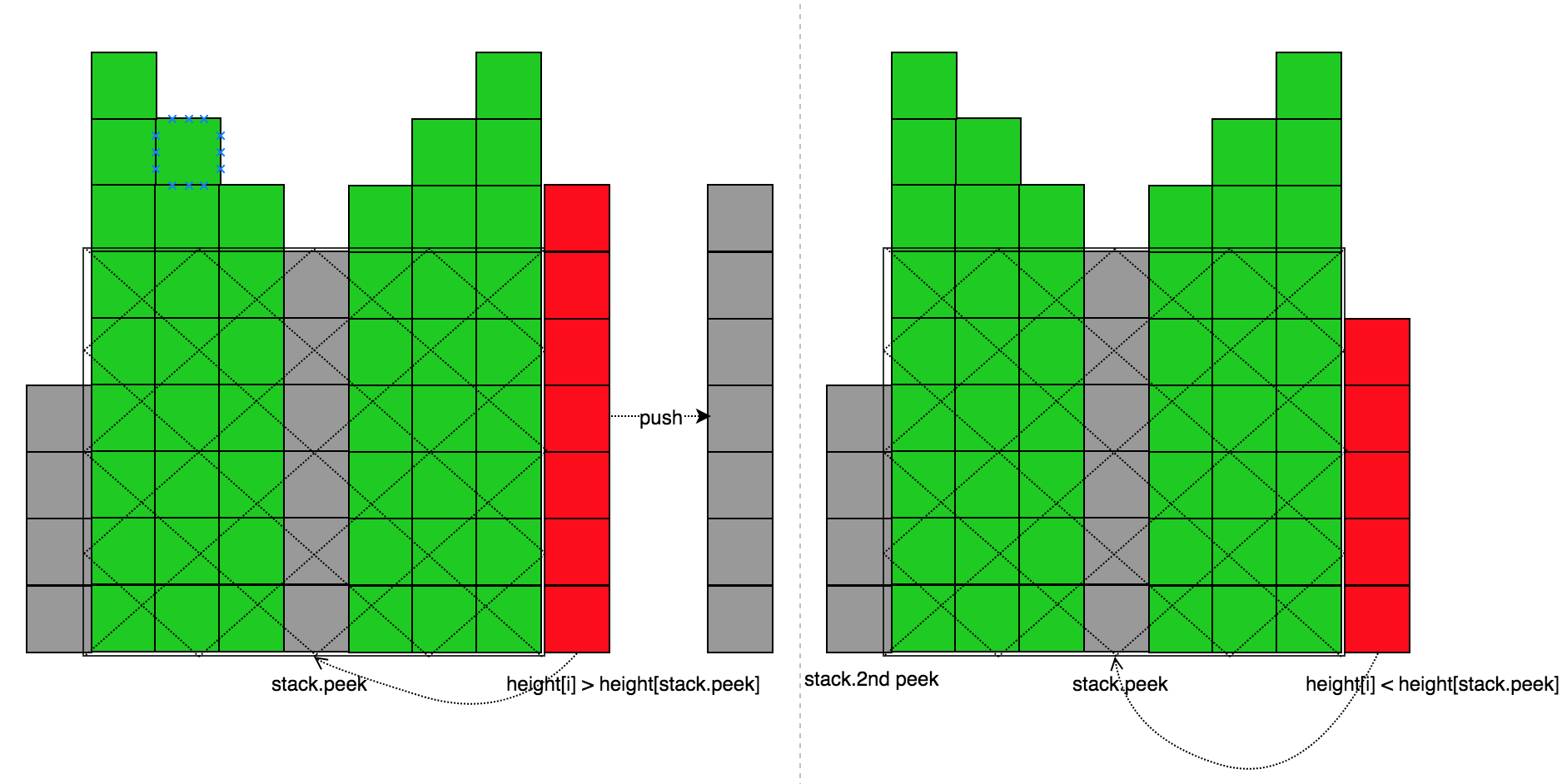
the minimum step of printing n ‘A’ s

**Intuition & Algorithm**

CopyAll & Paste, print as much ‘A’ as possible. Assume the minimum number of steps for printingA’s is. Then the best strategy is. For example, then the minimum cost is because a series ofaction is required.

**85. Maximal Rectangle**

**84. Largest Rectangle in Histogram** Facebook

****

**Intuition & Algorithm**

Always maintain a stack which maintains the ascending order of, as the above diagram suggests, the maximal rectangular area form by is:

**363. Max Sum of Rectangle No Larger Than k** Google

**Intuition & Algorithm**

Convert 2d problem into “find max subarray no larger than k”

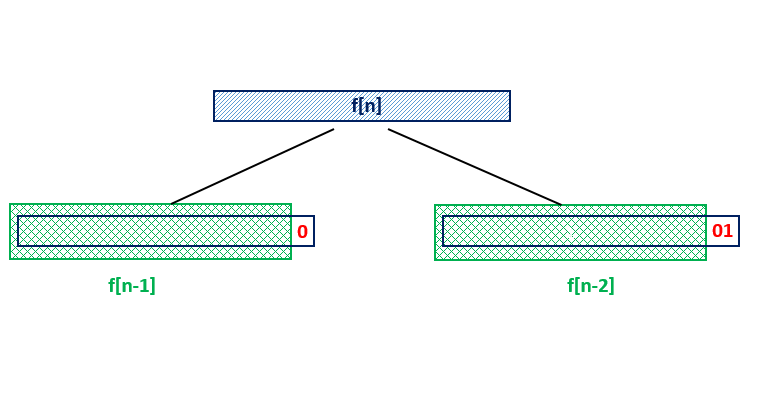
**Follow Up: Max Sum Rectangle**

**Find a rectangular area whose sum equals s**

**600. Non-negative integers without consecutive ones Pocket Gems**

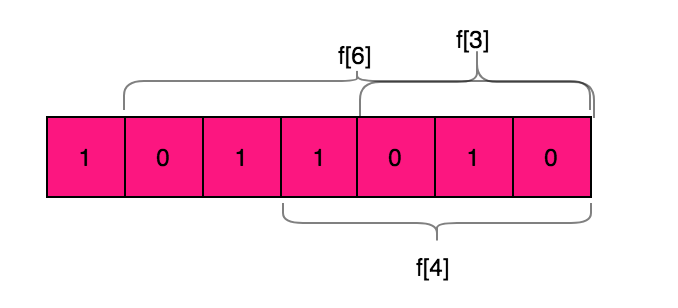
**Intuition**

Suppose, we need to find the count of binary numbers withbits such that these numbers don't contain consecutive 1's. In order to do so, we can look at the problem in a recursive fashion. Supposegives the count of such binary numbers withbits. To determine the value of, which is the requirement, we can consider the cases shown below:



From the above figure, we can see that if we know the value ofand In order to generate the required binary numbers withbits, we can append ato all the binary numbers contained in without creating an invalid number. But, we can't append ato all these numbers, since it could lead to the presence of two consecutive ones in the newly generated numbers. Thus, for the currently generated numbers to end with a, we need to ensure that the second last position is always. Thus, we need to fix aat the end of all the numbers contained inThus, in total, we get

For example, , at the digit of 5,3,2 () they are all 1’s. Thus, , since are consecutive 1’s, so it stops counting at



**Intuition & Algorithm**

First, calculate the number of integers without consecutive ones in length of

Second, for each of bit set to onesum all correspondingup until consecutive one’s occurs.

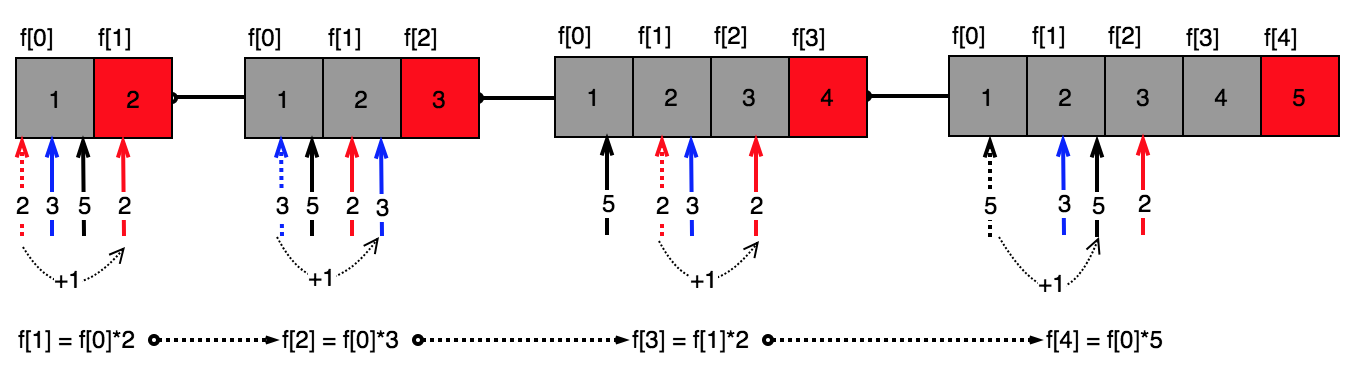
Finally, if consecutive one’s occurs, return

**152. Maximum Product Subarray LinkedIn**

**313. Super Ugly Number**

**264. Ugly Number I**

**263. Ugly Number II**

****

**91. Decode Ways Facebook Microsoft Uber**

**338. Counting Bits**

**329. Longest Increasing Path in a Matrix Google**

**241. Different Ways to Add Parentheses**

Given a string of numbers and operators, return all possible results from computing all the different possible ways to group numbers and operators. The valid operators are.

# Tree Set & Tries & Linked List

**456. 132 Pattern**

**Breaking Bad** Uber

Given a list of words, and a string , wrap all substring of which show up in with , and capitalize the first letter of such substring. For example,

(note: both show up in the . It should choose the longer one to wrap out.

<https://drive.google.com/open?id=1QdTRFv-RNSd3CypcFNz1s7fYdOtGQuPV>

**421. Maximum XOR of Two Numbers in an Array**

**211. Add & Search Word (Single word)**

**212. Add & Search Word II (Multiple words)** Google Airbnb Microsoft

**327. Count of Range Sum**

Solution: use variable sum to keep track the prefix sum so far, use TreeMap to store it.

if a subarray , then there must exists a prefix sum:

**24. Swap Nodes in Pairs**

**25. Reverse Nodes in k-group** Facebook Microsoft

Each time we find k nodes and reverse the order, then recursively make a call to itself dealing with the remaining nodes.

**237. Delete Node in a Linked List Microsoft Apple Adobe**

# B-Search & Array & Two Pts

**48. Rotate image**

**50. Pow (x, n)** Google Facebook Bloomberg LinkedIn

**56. Merge Intervals** Google Facebook Microsoft Bloomberg LinkedIn Twitter Yelp

Given a collection of intervals, merge all overlapping intervals.

**Follow Up: Merge Two Sorted Interval List /**

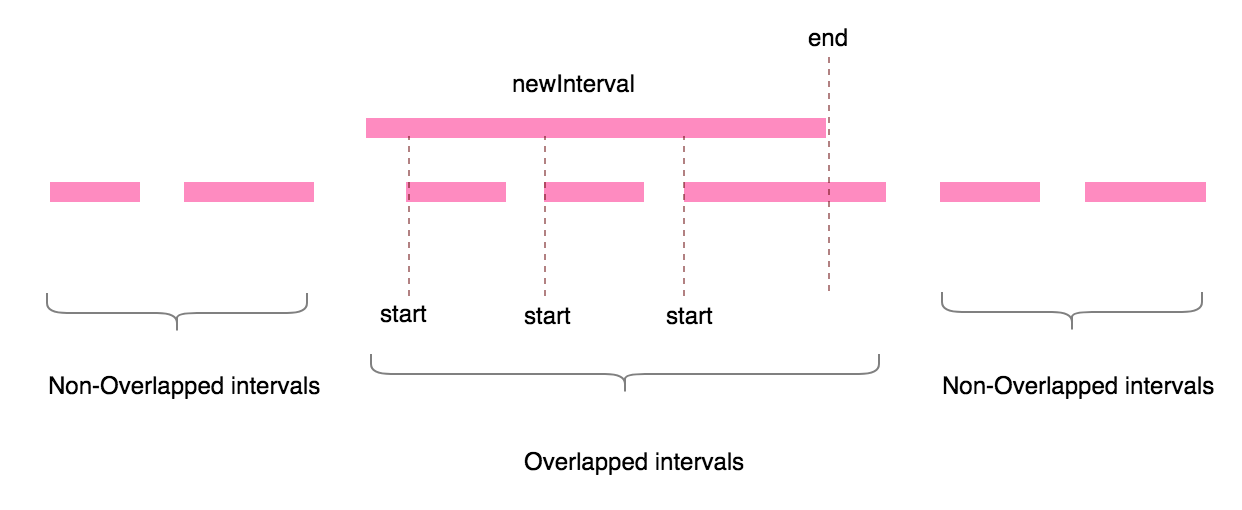
**Collect All Intersection of Two Sorted Interval List**

**[Intuition & Algorithm](https://github.com/fu11211129/Algorithm/blob/master/fb-interview/MergeTwoSortedIntervalLists.java)**

**Like Merge Two Sorted Arrays**

**153. Find Minimum in Rotated Sorted Array** Facebook

**57. Insert Interval** Google LinkedIn Facebook

****

**Intuition & Algorithm**

First, scan those non-overlapped intervals. Second, scan those overlapped intervals, Third, scan the remained intervals.

**154. Find Minimum in Rotated Sorted Array II**

**461. Hamming Distance** Facebook

**477. Total Hamming Distance Facebook**

Say for any particular bit position, count the number of elements with this bit **ON**. (i.e. this particular bit is 1). Let this count be. Hence the number of elements with this bit **OFF** (i.e. 0) isin anelement array). Thus, at this bit, the total hamming distance is:

**277. Find the Celebrity**

**Intuition & Algorithm**

Apply two pointers solver, since the celebrity doesn’t know anyone else in the party, and the other guys are not known for the celebrity.

**554. Brick Wall Facebook**

There is a brick wall in front of you. The wall is rectangular and has several rows of bricks. The bricks have the same height but different width. You want to draw a vertical line from the top to the bottom and cross the least bricks. If your line go through the edge of a brick, then the brick is not considered as crossed. You need to find out how to draw the line to cross the least bricks and return the number of crossed bricks.

**Intuition & Algorithm**

Find the position where the most of bricks’ edges occurs.

**287. Find the Duplicate Number Google**

All number in an array of size are in the range of

Solution: slow, fast pointers.

To be noticed, the array companied with such properties forms a cycle if a graph is drawn with index and its corresponding value, for example, an array like , a circle exists in the path by starting with index

Solution 2: negatives flag

**Closet & farthest Pair in 2 Sorted Arrays**

**Farthest**

**350. Intersection of Two arrays**

Solution: like merge sort

**Follow Up: one array, say is too large to fit into memory.**

Solution: read into memory using hash-set, and read chunk by chunk into memory, and then find the intersection.

**Follow Up: both two arrays are too large to fit into memory.**

Solution: using external sort against two arrays (on disk), and each time read top 2 elements from 2 arrays.

**75. Sort Colors** Uber Didi Labs

Time Complexity: suppose

, thus, the total amount of operations will be

Another Solution: use 3 delimiters: ,

**283. Move Zeros** FacebookBloomberg

Given an array, write a function to move all 0's to the end of it while maintaining the relative order of the non-zero elements.

**Intuition & Algorithm**

**260. Single Number III**

There are only 2 numbers show up only once, while the remaining numbers show up twice.

**Intuition & Algorithm**

all numbers, suppose there two distinct number are, then afterall numbers, it gives out. Since, we can make use of the trickwhich gives the right-most bit set to be “1”.so one of them say , and . For all the other numbers except , we can useto exactly divide those number into two independent groups.

So, in group I, all elements , while in group II, all elements . Also, since each number except occur twice in the group, we can do XOR again inside the group, and the result just gives out the first single number , and the same thing happen in the second group.

**268. Missing Number**

**Solution I: XOR**

**Solution II: Sum**

**Find Popular Element in an Array**

**190. Reverse Bits**

For example,

**Follow Up:**

Find lower or upper bound of searched item in sorted array

**Interview Q. Rotate Array** **Clock wisely in Place VMware**

**231. Power of Two**

**136. Single Number**

**137. Single Number II**

All numbers show uptimes while only one number which shows up only once

**209. Minimum Size Subarray Sum** Facebook

All integers in array are positives, find

**Intuition & Algorithm**

Apply two pointers, and always maintain the window sum no exceeds target. When window sum exceeds or equals to target, try to minimize the window size by moving the left pointer.

**Follow Up: Integers are not all positive.**

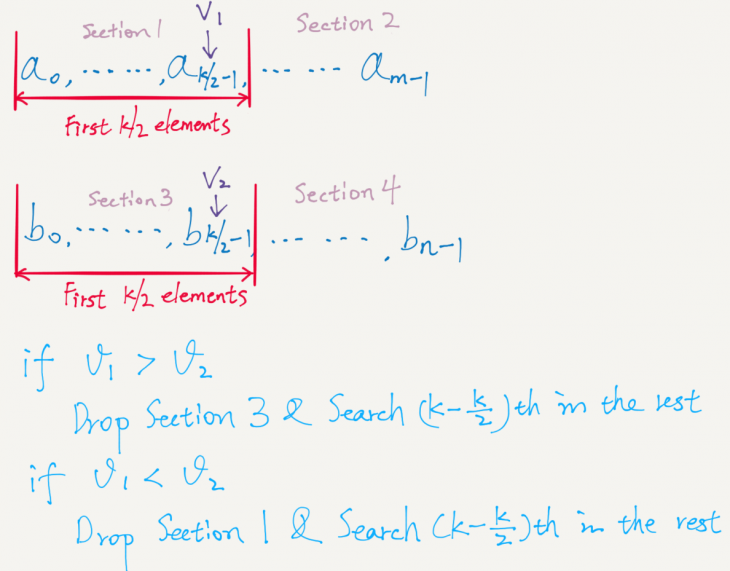
**Intuition & Algorithm**

Hash Map

**413. Arithmetic Slices**

**4. Median of Two Sorted Arrays**

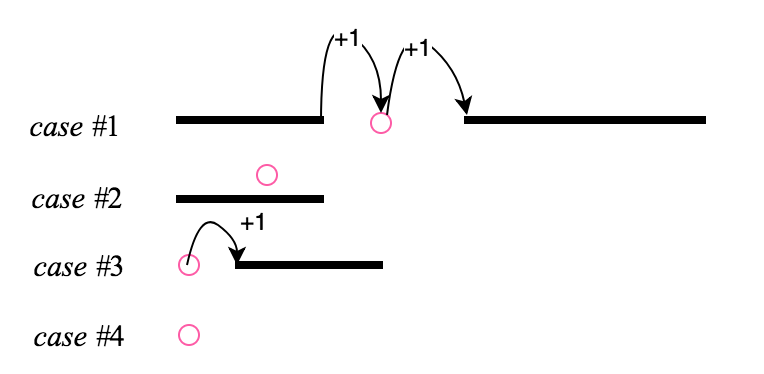
**Google Amazon Microsoft Apple Zenefits Yahoo Adobe Dropbox**



**Intuition & Algorithm**

Compare the two middle elements of A & B, then drop the smaller half.

**352. Data Stream as Disjoint Intervals**

****

**348. Design Tic-Tac-Toe Google Microsoft**

Two players play on agrid, and the diagram below suggests X wins

**Algorithm**

**33. Search in Rotated Sorted Array**

**34. Search for a Range**

Search for the lower & upper boundary of given value

**35. Search insert position**

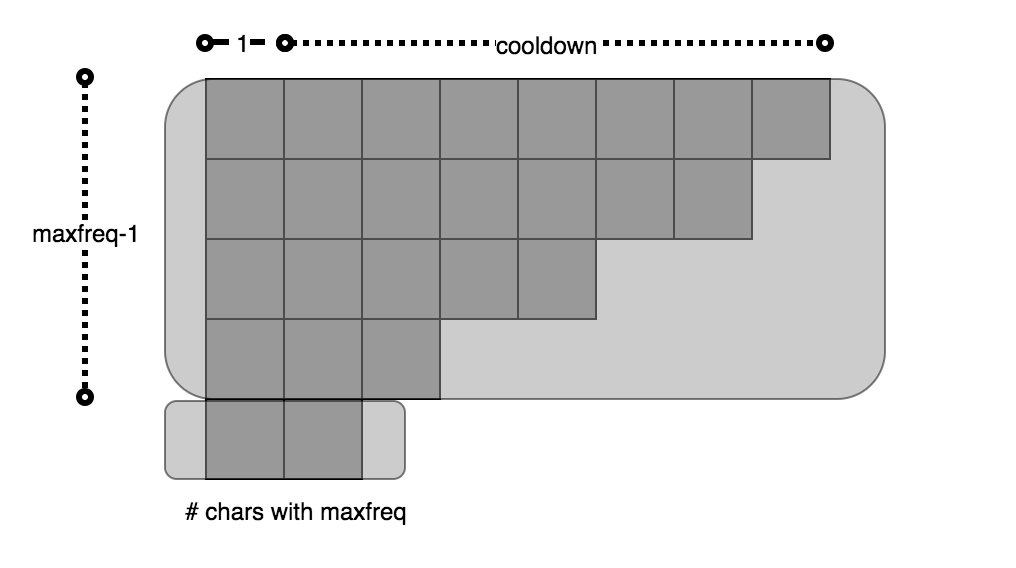
**240. Search a 2D Matrix II Google Apple Amazon**

**Intuition & Algorithm**

Starting searching from the right-upper corner

Time Cost:

**621. Task Scheduler Facebook**

****

**Algorithm**

**767. Reorganize String Uber Google**

**Algorithm**

priority queue

Each time take two nodes (if applicable) out of pq.

Then, if push it back to pq, the same for

Corner case, if one char whose frequency

Solution II: count sort, quite like Q. 621

**3. Longest Substring Without Repeating Characters**

**389. Find the difference Google**

**Algorithm**

**409. Longest Palindrome Google**

Given a string which consists of lowercase or uppercase letters, find the length of the longest palindromes that can be built with those letters.

**11. Container with Most Water Bloomberg**

Given n vertical lines:

**15. 3Sum & 18. 4Sum**

**454. 4Sum II**

Solution: hash A + B first, and then check C + D in hash map

**26. Remove Duplicates from Sorted Array Facebook Microsoft Bloomberg**

**27. Remove Element**

**392. Is Subsequence Uber**

Follow Up: Check a list of string where

**204. Count Primes**

**451. Sort Characters by Frequency** **Bloomberg**

Solution I: Bucket Sort

Time Complexity: , Space Complexity:

:

**692. Top K Frequent Words** Amazon Bloomberg Uber Yelp Pocket Gems

Solution: quite like 451

**Follow Up: Get K most frequent words from word stream.** Bloomberg

The solution above (451) works well when the number of words is not that large, the strategy here is trading time for space. But in many cases, the number of words is very large, say , while the number of different words is relatively small compare to it, say so it is possible to do word count with hash-map, but not possible to allocate such number of buckets for bucket sort. So, an optimized solution is Quick-Select, it takes time and **space**, where is the number of words in text.

**76. Minimum Window Substring &**

**567. Permutation in String &**

**438. Find All Anagrams in a String Uber**

**138. Copy List with Random Pointer Amazon**

**133. Clone Graph** Google Facebook Uber Pocket Gems

do bfs on graph, and make copy for each of node:

**31. Next Permutation** Facebook Google Microsoft Amazon Samsung LinkedIn

Generate next slightly greater number with same digits

**Intuition & Algorithm**



Find firstst.from rightmost, then find firstin range ofs.tfrom right-most. Next, swap. Finally, sort

**Next Palindromic Number**

Consider three conditions:

**80. Remove Duplicates from Sorted Array II** Facebook

Given a sorted array, remove the duplicates in-place such that duplicates appeared at most twice and return the new length.

**Intuition & Algorithm**

**442. Find all Duplicates in an Array** Pocket Gems

**Intuition & Algorithm**

Negative Labeling, sincewe can label

**41. First Missing Positive**

**448. Find All numbers disappeared in an array**

Solution: firstly swap all positive integers to the left, for example, , then apply “negative mark” algorithm.

**Common Chars in strings**

**318. Maximum Product of Word Lengths**

Find the maximum value of where the two words do not share common letters.

**Intuition & Algorithm**

encode string using binary format, like , so checking if two strings share the same characters just need to check or not (intime). Thus, the overall time complexity is

# BT & BST & Iterator

**302.** [**Smallest Rectangle Enclosing Black Pixels**](https://github.com/fu11211129/Algorithm/blob/master/Array/google/SmallestRectangleEnclosingBlackPixels.java) **Google**

An image is represented by a binary matrix with 0 as a white pixel and 1 as a black pixel. The black pixels are connected, i.e., there is only one black region. Pixels are connected horizontally and vertically. Given the location (x, y) of one of the black pixels, return the area of the smallest (axis-aligned) rectangle that encloses all black pixels.

**Intuition & Algorithm**

Apply B-Search to find the left, right, top & bottom boundary of 1.

**Time Costs**

**235. LCA of BST & BT**

**236. LCA of BT**

**Interview Q.** [**2nd Largest Element in BST**](https://drive.google.com/open?id=1sOeQZb2sZ2i7Srm66uS4J58_R6jpUzEV) **NewsOnChat**

Time Complexity:

**652.** [**Find Duplicate Subtrees**](https://leetcode.com/submissions/detail/138874388/) **Google**

Solution: Preorder + Hash-map

Since same tree structure correspond to the same preorder serialization. I use a hash-map to map the tree preorder serialization to the occurrence times. Once one serialization shows up 2 or more than twice, I add it to the result.

**173. Binary Search Tree Iterator** Google Facebook Microsoft LinkedIn

**Intuition & Algorithm**

Can Refer to <https://www.geeksforgeeks.org/inorder-tree-traversal-without-recursion/>

**98. Validate Binary Search Tree Facebook Microsoft Amazon Bloomberg**

**99. Recover Binary Search Tree**

Two nodes are mistakenly swapped in BST, asked to find them and swap back

**285. In-order Successor in BST Facebook Microsoft Pocket Gems**

**572. Subtree of Another Tree Facebook eBay**

Given two non-empty binary trees s and t, check whether tree t has exactly the same structure and node values with a subtree of s

**Intuition & Algorithm**

Two BTs has the same structure & node values **if & only-if** two BTs share the same pre-order serialized representation.

Time Costs:

**Intuition & Algorithm**

For each of node in treecheck if every subtree is the same as.

Time Costs:

**145. Binary Tree Post-Order Traversal**

**Intuition & Algorithm**

**199. Binary Tree Right Side View NewsOnChat**

**226. Invert Binary Tree**

Solution: reverse left-sub tree & right-sub tree first, then swap left & right pointer.

**100. Same Tree**

**101. Symmetric Tree**

**297. Serialize & De-serialize BT**

**Google Facebook Microsoft Amazon Bloomberg Uber LinkedIn Yahoo**

Solution: preorder

Time Complexity:, Space Cost:, there arenodes(including null node), each node takesfor storage.

**Follow Up:** Succinct serialization & de-serialization **Google**

Use two lists, one is called structure, another is called data, both are stored in pre-order.

**102. Binary Tree Level Order Traversal & II (107)**

**103. Binary Tree Zigzag Level Order Traversal**

Solution: BFS

**144. Binary Tree Preorder Traversal & Post-order (145)**

**450. Delete Node in BST**

**104. Maximum Depth of Binary Tree**

Solution:

**110. Balanced BT Bloomberg**

**Interview Q. Preorder to BST**

**323. Number of Connected Components in an Undirected Graph**

Solution: union-find

**543. Diameter of Binary Tree Google Facebook**

Solution: pre-calculate height for each node in BT, then store the mapping into hash-map, finally calculate the diameter like this:

[**Longest Path inTree**](https://drive.google.com/open?id=1l1l5LBv7Tj8wZiuaCc2JQx6dDzWZeCpb) **Jingchi.ai**

Solution: pre-calculate height for each node in BT, then store the mapping into hash-map, finally

:

**549. Binary Tree Longest Consecutive Sequence II Google**

**Follow Up: N-ary Tree Longest Consecutive Sequence**

**Largest Complete Tree ForUsAll**

, to be noticed, it takes the minimal one of left subtree & right subtree plus 1 as its complete tree height, which is different than our node’s height computation where

**105. Construct BT from Preorder & In-order Traversal**

**106. Construct BT from in-order & Post-order Traversal**

**108. Convert Sorted Array to BST**

**109. Convert Sorted List to BST** Zenefits

**255. Verify Preorder Sequence in Binary Search Tree**

**Algorithm**

**Algorithm**

**114. Flatten Binary Tree to Linked List**

**426. Convert Binary Search Tree to Sorted Doubly Linked List** Facebook

**116. Populating Next Right Pointers in each Node**

**117. Populating Next Right Pointers in each Node II Microsoft**

**Queue & Stack & Greedy& Heap**

**155. Minimum Stack**

**Follow Up: Maximum & Minimum Stack ForUsAll**

solution: maintain three variables:

**Algorithm**

**232. Implement Queue using Stacks Microsoft Bloomberg**

solution: apply two stacks, one is called “in”, another is called “out”.

When new element comes in, push it in the “in” stack, in case of the pop operation of queue, there are two cases two be handled, if the “out” stack is not empty, then pop the top one, otherwise, push all the elements in the “in” stack and pop the top one of “out” stack.

**346. Moving Average from Data Stream Uber**

**341. Flatten Nested List Iterator Uber**

Solution: DFS / Stack

Stack Solution:

**225. Implement Stack using Queues**

solution:

):

**321. Create Maximum Number Google**

**316. Remove Duplicate Letter** Google

Given a string which contains only lowercase letters, remove duplicate letters so that every letter appears once and only once. You must make sure your result is the smallest in lexicographical order among all possible results.

**402. Remove K Digits** Google Snapchat

Given a non-negative integer num represented as a string, remove k digits from the number so that the new number is the smallest possible.

**Intuition & Algorithm**

The same idea as **316. Remove Duplicate Letter.**

**224. Basic Calculator**

**227. Basic Calculator II** Uber

**II. only +, -, \*, /**

)

)

)

)

**150. Evaluate Reverse Polish Notation**

**20. Valid Parentheses**

**Interview Q: Heap Sort** Didi Labs

Build the max-heap first, then each time shrink the array by swapping the last element with the first one, and reconstruct the max-heap again.

**295. Find Median from Data Stream**

**Intuition & Algorithm**

Maintain two heaps, one max-heap, storing the smaller half, and one min-heap, storing the larger half.

Solution I: max-heap, store the smaller half, min-heap, for the larger half.

**Intuition & Algorithm**

Array-list for storing, quick-select for fetching the median.

**239. Sliding Window Maximum**

**Follow Up: Sliding Window Max Difference**

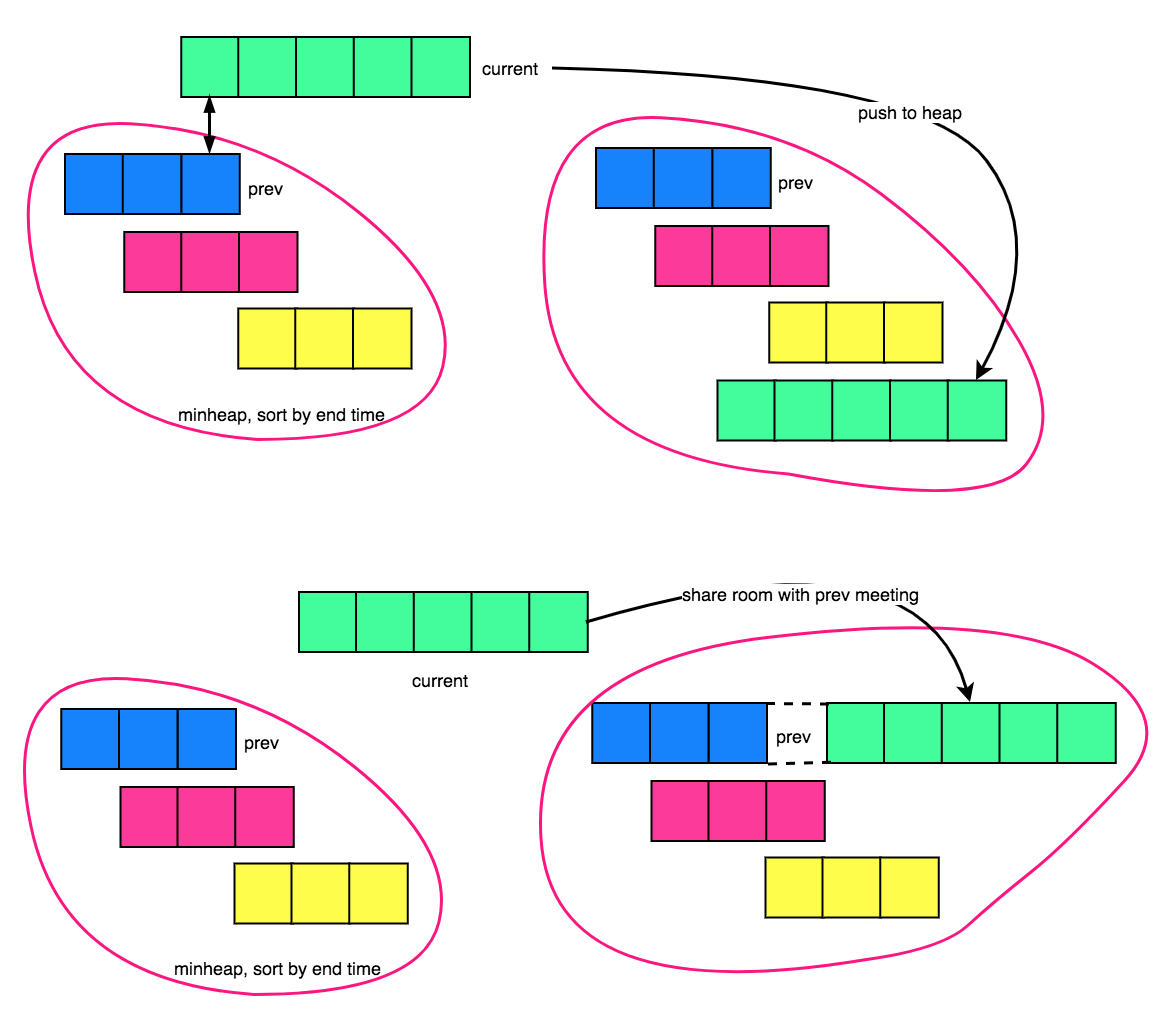
Solution: it just needs an another deque maintaining an ascending order of integers.

**407. Trapping Rain Water II**

**252. Meeting Rooms I** Facebook

I: Given a list of conference time intervals, check if one can take all intervals.

**253.** [**Meeting Rooms II**](https://github.com/fu11211129/Algorithm/blob/master/greedy/google/MeetingRoomII.java)Facebook

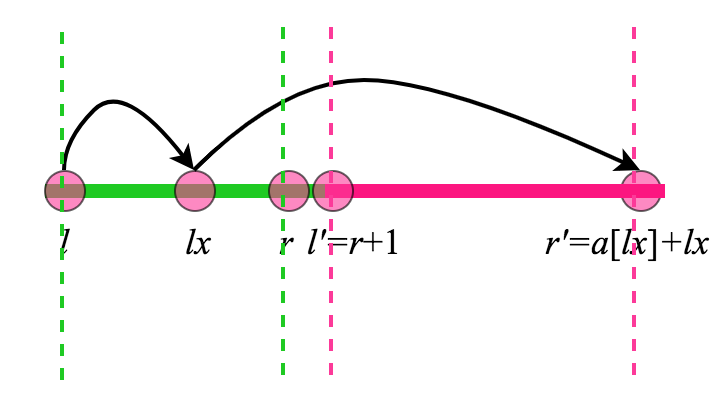
Find minimum rooms for arranging n meeting intervals.

**Find the point where maximum intervals overlap Uber**

**Intuition & Algorithm**

**55. Jump Game I**

**45. Jump Game II Microsoft**

****

**Intuition & Algorithm**

Each time select one pointwhich maximize the value of. And fnally update the window from

**343. Integer Break**

**397. Integer Replacement**

**14. Longest Common Prefix Yelp**

Apply Trie algorithm

:

**49. Group Anagrams**

Solution I:

Solution II: assign an unique prime number product for string.

**187. Repeated DNA Sequence**

**151. Reverse words in a string**

Solution: reverse each word, then reverse the whole string.

**6. Zigzag Conversion**

**214. Shortest Palindrome**

Apply KMP algorithm of

# DFS & BFS

**360. Valid Sudoku & Sudoku Solver**

**51. N-Queens & II (52)** Zenefits

**78. Subsets (Duplicated Subsets allowed)**

**90. Subsets II (No Duplicated Subsets allowed)** Facebook

Given a collection of integers that might contain duplicates,, return all possible subsets. The solution set must **not contain duplicate subsets**.

**Intuition & Algorithm**

**79. Word Search**

**Intuition**

DFS + Backtrack

**22. Generate Parentheses Google Uber Zenefits**

**46. Permutation**

**47. Permutation II**

**384. Shuffle an Array**

**39. Combination Sum**

**40.** [**Combination Sum II**](https://github.com/fu11211129/Algorithm/blob/master/Array/snapchat/CombinationSumNoDuplicate.java) **Uber**

**Intuition & Algorithm**

The key idea of avoiding duplicates is

**54. Spiral Matrix & II (59) Google Microsoft Uber**

**112. Path Sum I**

**113. Path Sum II**

**129. Sum Root to Leaf Numbers**

**II. Solution: DFS + backtrack**

**400. N-th Digit**

**43. Multiply Strings Uber Facebook Twitter**

**50. Pow(x, n)**

**372. Super Pow**

**149. Max Points on a Line**

**342. Power of Four Uber**

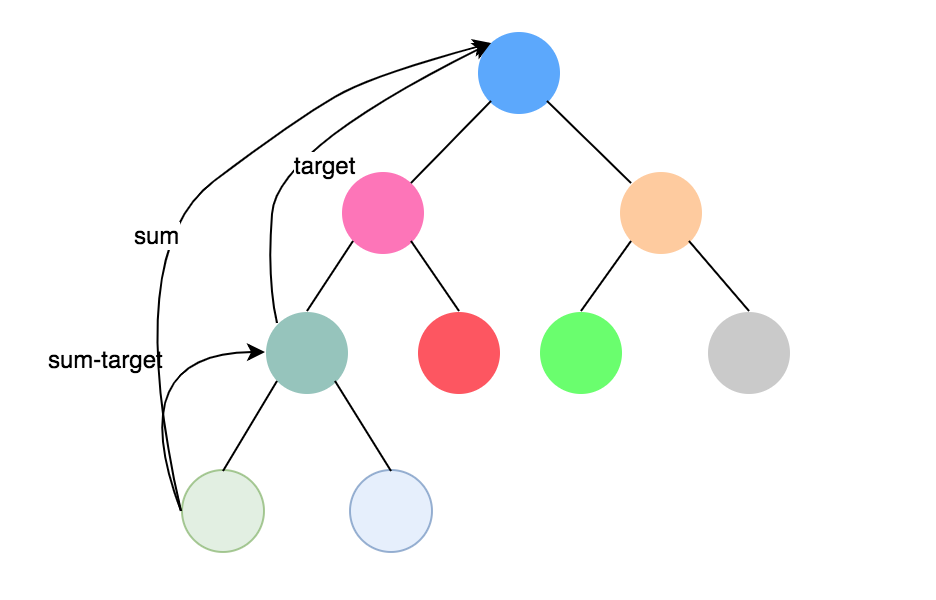
**Solution II:**

**326. Power of Three**

# Hash Map & Tree Map & BIT

**437. Path Sum III**

You are given a binary tree in which each node contains an integer value. Find the number of paths that sum to a given value. The path does not need to start or end at the root or a leaf, but it must go downwards (traveling only from parent nodes to child nodes).



**Intuition & Algorithm**

The same idea as “count how many subarrays that sum to target”, useto keep track of the sum from root node to current node, andto record the number of times that the sum of root nodes down to current node shows up.

**Number of subarrays having sum exactly equal to k** GeeksForGeeks

1. **Two Sum**

**Interview Q. Uber**

Implement a hash-map which supports 3 operations: put, get & putAll(value), to be noticed, putAll(value) updates all KV pairs with the new value.

**Interview Q.** Uber

Given a hash-map, like “apple”:10, “banana”:50, “peach”:30

The value part suggests the relative weight, this question ask you to randomly output a key in accordance with the corresponding weight.

The above algorithm takes time

**Solution II:**

If the input is given in a array format (or we can export the KV pairs in hash-map into an array), we can search the target using binary search

**525. Contiguous Array Facebook**

Given a 01 array, find the maximum length of a contiguous subarray with equal number of 0 and 1.

**Intuition & Algorithm**

Translate the, then this problem is to find such a longest subarray whose sum equals to zero.

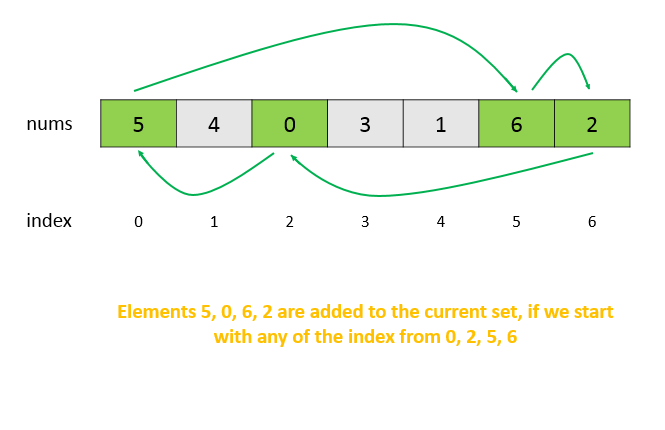
**325. Maximum Size Subarray Sum Equals k Facebook Palantir**

Follow Up: if it is asked to find the minimum size subarray, then just remove the blue line of code above.

**565. Array Nesting Apple**

A zero-indexed arrayof lengthcontains all integers fromto. Find and return the longest length of set S, wheresubjected to the rule below.

i.e. , form a cycle with length of 4.



**Intuition & Algorithm**

**17. Letter Combinations of a Phone Number** **Uber**

**Follow Up:** Given a phone number list & word dictionary, find the phone number which maps the most number of words.

**200. Number of Islands** Google Facebook Microsoft Amazon Zenefits

Given a 2d grid map of '1's (land) and '0's (water), count the number of islands. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

**Algorithm**

Nothing trivial, just do DFS by starting from each island (1)

**305. Number of Islands II Google**

**Intuition & Algorithm**

Union-Find, for each of newly added island, we increase the number of islands by 1. Then, find 4-directionally adjacent island and union them, if the union two islands have already in the same group, do nothing. Otherwise, decreasing the number of islands by 1.

**128. Longest Consecutive Sequence**

**Solution I: hash-map**

**644.** [**Maximum Average Subarray II**](https://github.com/fu11211129/Algorithm/blob/master/Array/google/MaxAvgSubarrayII.java) **Google**

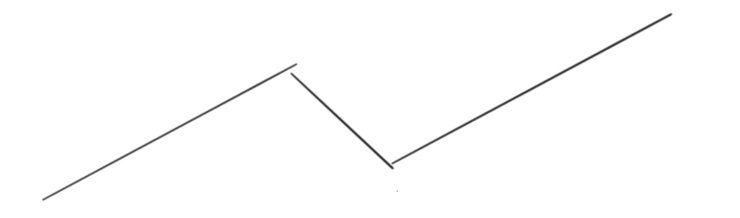
Find a subarray with a length () whose average is the largest.

Solution: binary guess or Simulated Annealing algorithm. Since certainly, such an average satisfying the condition must be in the range of , so we can continuously enumerate avg in, and then check if there is a subarray*.*

*s.t. or .*

**665. Non-decreasing Array Google**

Check if an array can be non-decreasing by modifying at-most 1 element.



the above picture suggests an example of such array that making an array non-decreasing by modifying at-most one element.

So we just need to find such an position i such that , and then I make nums[i], nums[i+1] equal by eitherr(case i) or(case ii)**.** if in either of cases, is in non-decreasing order and is also in non-decreasing order. Then return true, return false otherwise.

**670. Maximum Swap Facebook**

**Intuition & Algorithm**

At each digit of the input number in order, if there is a larger digit that occurs later, we know that the best swap must occur with the digit we are currently considering. We will compute , the index of the last occurrence of digit(if it exists).

Afterwards, when scanning the number from left to right, if there is a larger digit in the future, we will swap it with the largest such digit; if there are multiple such digits, we will swap it with the one that occurs the latest.

**Heap-Sort Didi Labs**

**K-dimensional Tree**

common questions:

range query (find all points inside the given range )

**269. Alien Dictionary**

examples: input: words[] = {"baa", "abcd", "abca", "cab", "cad"}

output: Order of characters is 'b', 'd', 'a', 'c'

**271. Closest Binary Search Tree Value & 272**

**272. Find k Closet Values**

Solution 2. flatten BST into sorted list, then find k closet node on list

# Graph & Topo-sort & Union-Find

**127. Word Ladder & II (126)** Uber Zillow

**Interview Q. Bipartite graph check** Uber

**433. Minimum Genetic Mutation Twitter**

Determine the minimum number of mutations needed to mutate from "start" to "end". If there is no such a mutation, return -1.

v

**207. Course Schedule**

**210. Course Schedule II** Facebook Zenefits

There are a total of n courses you have to take, labeled from 0 to n-1. Some courses may have prerequisites, for example to take course 0 you have to first take course 1, which is expressed as a pair:. Given the total number of courses and a list of prerequisite pairs, return the ordering of courses you should take to finish all courses. There may be multiple correct orders, you just need to return one of them. If it is impossible to finish all courses, return an empty array.

**Intuition & Algorithm**

Topological sort

**310. Minimum Height Trees Google**

For an undirected graph with tree characteristics, we can choose any node as the root. The result graph is then a rooted tree. Among all possible rooted trees, those with minimum height are called minimum height trees (MHTs). Given such a graph, write a function to find all the MHTs and return a list of their root labels.

**Intuition & Algorithm**

Topological Sorting on an undirected graph, we keep removing the node whose indegree is 1, that is, the leaf node. We update minHeightTrees in each level, the last one remained will be the list containing roots.

# Math & Geometry & Number Theory

**365. Water and Jug Problem (Number Theory)**

You are given two jugs with capacities x and y liters. There is an infinite amount of water supply available. You need to determine whether it is possible to measure exactly z liters using these two jugs. For example,

Solution: this is a kind of Euclidean problem, where I can assume you did times operation on , times operations on to measure , socan be measured if & only if .

**367. Valid Perfect Square LinkedIn**

Given a positive integer, determine if the given number is perfect square or not.

**Intuition & Algorithm**

**168. Excel Sheet Column Title** Facebook Microsoft Zenefits

Given a positive integer, return its corresponding column title as appear in an Excel sheet.

For example,

**469. Convex Polygon Google**

**Area of Polygon**

**Intersection point of ray & triangle-3d.**

Triangle-3d can be presented as:

Ray can be presented as: .

Supposing the intersection point lies on the triangle is , so we have:

because the normal vector of plane (based on triangle) is , and , and , , so the intersection point is

**Fibonacci progression**

Solution:

**Check if one number n is a Fibonacci number**

**# Find Path In a Maze**

solution: A\* algorithm.

# String & Easy

**166. Fraction to Recurring Decimal**

**29. Divide two Integers**

**526. Beautiful Arrangement**

**482. License Key Formatting**

just familiar with several functions:

**12. Integer to Roman & Roman to Integer (13)** Oscar Health

**65. Valid Number**

For example,

**Interview Q. Given a word dictionary & a sentence**

For example,

Solution: grab all substring of sentence at first, and check each substring exists in dictionary or not.

Optimization: avoid checking too-long substring.

**648. Replace Words Uber**

**Replace words in a string with shortest prefix in dict.**

**412. Fizz Buzz**

**415. Add Strings**

**695. Max Area of Island**

Solution: simple dfs

Note: cut branches of dfs tree by turn the “island” to water during dfs.

<https://leetcode.com/submissions/detail/140076776/>

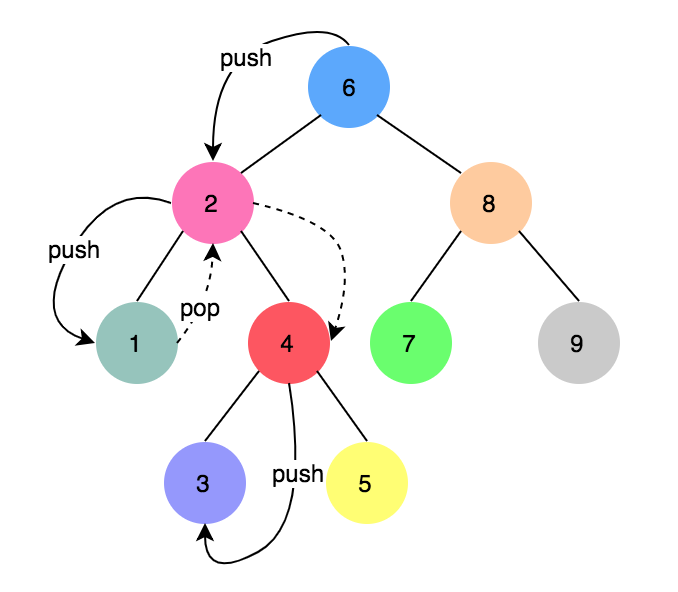
**67. Add Binary** Facebook

**38. Count and Say** Facebook

**463. Island Perimeter**

**94.** [**Binary Tree In-order Traversal**](https://github.com/fu11211129/Algorithm/blob/master/btree/microsoft/BinaryTreeInorderTraversal.java) **Microsoft**

**Non-recursive Solution**

****

**202. Happy Number**

**217. Contains Duplicate**

**219. Contains Duplicate II**

Find if there exists 2 indexes

**83. Remove Duplicates from Sorted List & II (82)**

**II. Remove all duplicated number only remain single number**

**242. Valid Anagram**

**147. Insertion Sort List**

**148. Sort List**

**387. First Unique Character in a String**

**290. Word Pattern I & II (291) Uber**

Given a and a string , find if follows the same pattern

**205. Isomorphic Strings**

Q. check two strings s & t are isomorphic strings

Solution: like solution of 290 & 291, use hash-map & hash-set

**II. No empty space between words in a string**

Solution: DFS, time

**66. Plus One**

**369. Plus One Linked List Google**

Idea: reverse the linked list, plus one on first node and do carry-over on each node whose value exceeds 10, finally reverse the linked list again.

<https://github.com/fu11211129/Algorithm/blob/master/list/google/PlusOneLinkedList.java>

**169. Majority Element Zenefits Adobe**

Solution: voting algorithm.

**86. Partition List**

**88. Merge Sorted Array Facebook Microsoft Bloomberg**

Given two sorted integer arraysand, mergeintoas one sorted array.

**Intuition**

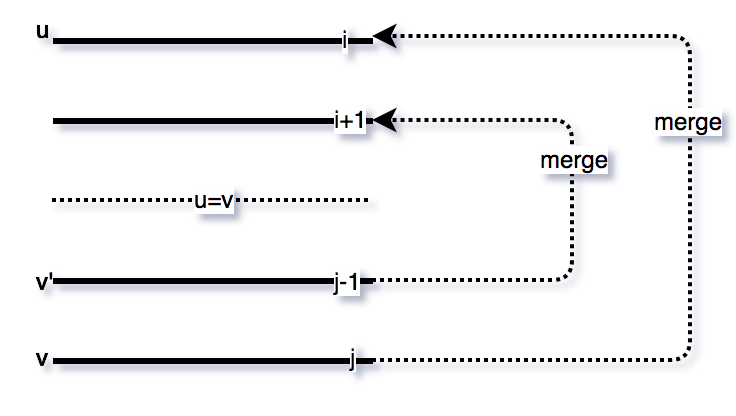
Scan from the back

**23. Merge k Sorted Lists Facebook**

**Intuition**

Initialize two pointers, one points to the first list, another point to the last one. Each time merge two lists together into first one.

**Algorithm**

****

**Algorithm**

Min Heap, Time Cost: & Space Cost:

**412. Fizz Buzz**

**758. Bold Words in String** Google

Solution I:

Optimization: since for real English words, there is no word with a length greater than 10, so we do not need to get all substring of, instead, just need to retrieve the , then the time complexity can be reduced from where