Java solution and explanation using invariants

476 views

+11
votes

I find it useful to reason about binary search problems using invariants. While there are many solutions posted here, neither of them provide (in my opinion) a good explanation about why they work. I just spent some time thinking about this and I thought it might be a good idea to share my thoughts.

Assume we initialize left = 0, right = nums.length - 1. The invariant I'm using is the following:

nums[left - 1] < nums[left] && nums[right] > nums[right + 1]

That basically means that in the current interval we're looking, [left, right] the function started increasing to left and will eventually decrease at right. The behavior between [left, right] falls into the following 3 categories:

1) nums[left] > nums[left + 1]. From the invariant, nums[left - 1] < nums[left] => left is a peak

2) The function is increasing from left to right i.e. nums[left] < nums[left + 1] < .. < nums[right - 1] < nums[right]. From the invariant, nums[right] > nums[right + 1] => right is a peak

3) the function increases for a while and then decreases (in which case the point just before it starts decreasing is a peak) e.g. 2 5 6 3 (6 is the point in question)

As shown, if the invariant above holds, there is at least a peak between [left, right]. Now we need to show 2 things:

I) the invariant is initially true. Since left = 0 and right = nums.length - 1 initially and we know that nums[-1] = nums[nums.length] = -oo, this is obviously true

II) At every step of the loop the invariant gets reestablished. If we consider the code in the loop, we have mid = (left + right) / 2 and the following 2 cases:

a) nums[mid] < nums[mid + 1]. It turns out that the interval [mid + 1, right] respects the invariant (nums[mid] < nums[mid + 1] -> part of the cond + nums[right] > nums[right + 1] -> part of the invariant in the previous loop iteration)

b) nums[mid] > nums[mid + 1]. Similarly, [left, mid] respects the invariant (nums[left - 1] < nums[left] -> part of the invariant in the previous loop iteration and nums[mid] > nums[mid + 1] -> part of the cond)

As a result, the invariant gets reestablished and it will also hold when we exit the loop. In that case we have an interval of length 2 i.e. right = left + 1. If nums[left] > nums[right], using the invariant (nums[left - 1] < nums[left]), we get that left is a peak. Otherwise right is the peak (nums[left] < nums[right] and nums[right] < nums[right + 1] from the invariant).

Back to Problem (/problems/find-peak-ele

All categories (../)

OJ (../oj) (31,385)
Two Sum (../oj/two-sum) (368)
Add Two Numbers (../oj/add-two-numbers) (224)
Longest Substring Without Repeating Characters (../oj/longest-substring-without-repeating-characters)
Median of Two (242) Sorted Arrays (../oj/median-of-two-sorted-arrays) (155)
Longest Palindromic Substring (../oj/longest-palindromic-substring) (160)
ZigZag Conversion (../oj/zigzag-conversion) (285)
Reverse Integer (../oj/reverse-integer)
String to Integer (211) (atoi) (../oj/string-to-integer-atoi) (202)
Palindrome Number (../oj/palindrome-number) (215)
Regular Expression Matching

```
public int findPeakElement(int[] nums) {
    int N = nums.length;
    if (N == 1) {
        return 0;
    }

    int left = 0, right = N - 1;
    while (right - left > 1) {
        int mid = left + (right - left) / 2;
        if (nums[mid] < nums[mid + 1]) {
            left = mid + 1;
        } else {
            right = mid;
        }
    }

    return (left == N - 1 || nums[left] > nums[left + 1]) ? left : right;
}
```

I hope this makes things clear despite the long explanation.

asked (../68999/java-solution-and-explanation-using-invariants) **Nov 11, 2015** in **Find Peak Element (../oj/find-peak-element)** by **cosmin79 (../user/cosmin79)** (710 points)
edited **Nov 12, 2015** by **cosmin79 (../user/cosmin79)**

Answer comment

shouldnt it be left = mid+1;

commented (../68999/java-solution-and-explanation-using-invariants?show=69051#c69051) Nov 11, 2015 by **raghuveer3 (../user/raghuveer3)** reply

you are correct. I've updated the explanation and the code.

commented (../68999/java-solution-and-explanation-using-invariants?show=69136#c69136) Nov 12, 2015 by **cosmin79 (../user/cosmin79)** reply

Excellent explanation. Thank you very much for posting.

commented (../68999/java-solution-and-explanation-using-invariants?show=72367#c72367) Nov 30, 2015 by **stephenc227 (../user/stephenc227)** reply

very good job!

commented (../68999/java-solution-and-explanation-using-invariants?show=74014#c74014) Dec 10, 2015 by **GWTW (../user/GWTW)** reply

2 Answers

+3 votes

I wanted to visually see how it works, so here's an example I worked out:

```
         5   5                           5
        / \ / \                         / \
       4   4   4                       4  −∞
      /         \                     /
     3           3           3       3
    /             \         / \     /
   2               2       2   2   2
  /                 \     /     \ /
−∞                   1   1       1
                      \ /
                       0
   0 1 2 3 4 5 6 7 8 910111213141516171819 indices
   2,3,4,5,4,5,4,3,2,1,0,1,2,3,2,1,2,3,4,5 (20 nums)
   l                 m                   r l=0, m=9, r=19
   l       m         r                     l=0, m=4, r= 9
             l   m   r                     l=5, m=7, r= 9
             l>m r                         l=5, m=6, r= 7
  return n[l] > n[l + 1])? l : r
```

(../oj/regular-expression-matching) (161)
Container With Most Water (../oj/container-with-most-water) (54)
Integer to Roman (../oj/integer-to-roman) (92)
Roman to Integer (../oj/roman-to-integer) (137)
Longest Common Prefix (../oj/longest-common-prefix) (192)
3Sum (../oj/3sum) (187)
3Sum Closest (../oj/3sum-closest) (66)
Letter Combinations of a Phone Number (../oj/letter-combinations-of-a-phone-number) (150)
4Sum (../oj/4sum) (107)
Remove Nth Node From End of List (../oj/remove-nth-node-from-end-of-list) (165)
Valid Parentheses (../oj/valid-parentheses) (187)
Merge Two Sorted Lists (../oj/merge-two-sorted-lists) (160)
Generate Parentheses (../oj/generate-parentheses) (168)
Merge k Sorted Lists (../oj/merge-k-sorted-lists) (180)
Swap Nodes in Pairs (../oj/swap-nodes-in-pairs) (147)
Reverse Nodes in k-Group (../oj/reverse-nodes-in-k-group) (118)
Remove Duplicates from Sorted Array (../oj/remove-duplicates-from-sorted-array) (205)
Remove Element (../oj/remove-element) (207)
Implement strStr() (../oj/implement-strstr) (173)
Divide Two Integers (../oj/divide-two-integers) (120)
Substring with

answered (../68999/java-solution-and-explanation-using-invariants?show=74164#a74164) **Dec 12, 2015** by **TWiStErRob (../user/TWiStErRob)** (3,620 points)

ask related question comment

0
votes

```
public int findPeakElement(int[] nums) {
    int left = 0;
    int right = nums.length - 1;
    while(left < right) {
        int mid = left + (right - left) / 2;
        if(nums[mid] < nums[mid + 1]) {
            left = mid + 1;
        } else {
            right = mid;
        }
    }
    return left;
}
```

simpler Java code

answered (../68999/java-solution-and-explanation-using-invariants?show=75077#a75077) **Dec 19, 2015** by **BettyFlying (../user/BettyFlying)** (290 points)

ask related question comment

Concatenation of All Words (../oj/substring-with-concatenation-of-all-words)	(104)
Next Permutation (../oj/next-permutation)	(115)
Longest Valid Parentheses (../oj/longest-valid-parentheses)	(118)
Search in Rotated Sorted Array (../oj/search-in-rotated-sorted-array)	(99)
Search for a Range (../oj/search-for-a-range)	(146)
Search Insert Position (../oj/search-insert-position)	(129)
Valid Sudoku (../oj/valid-sudoku)	(143)
Sudoku Solver (../oj/sudoku-solver)	(104)
Count and Say (../oj/count-and-say)	(182)
Combination Sum (../oj/combination-sum)	(127)
Combination Sum II (../oj/combination-sum-ii)	(88)
First Missing Positive (../oj/first-missing-positive)	(112)
Trapping Rain Water (../oj/trapping-rain-water)	(104)
Multiply Strings (../oj/multiply-strings)	(90)
Wildcard Matching (../oj/wildcard-matching)	(92)
Jump Game II (../oj/jump-game-ii)	(94)
Permutations (../oj/permutations)	(154)
Permutations II (../oj/permutations-ii)	(136)
Rotate Image (../oj/rotate-image)	(118)
Anagrams (../oj/anagrams)	(126)
Pow(x, n) (../oj/powx-n)	(127)
N-Queens (../oj/n-queens)	(98)
N-Queens II (../oj/n-queens-ii)	(69)
Maximum Subarray (../oj/maximum-	

subarray) (134)
Spiral Matrix
(../oj/spiral-matrix)
Jump Game (121)
(../oj/jump-game)
Merge Intervals (133)
(../oj/merge-intervals)
Insert Interval (128)
(../oj/insert-interval)
Length of Last (93)
Word (../oj/length-of-
last-word) (150)
Spiral Matrix II
(../oj/spiral-matrix-ii)
Permutation (101)
Sequence
(../oj/permutation-
sequence) (99)
Rotate List
(../oj/rotate-list) (101)
Unique Paths
(../oj/unique-paths)
Unique Paths II (113)
(../oj/unique-paths-ii)
Minimum Path (88)
Sum (../oj/minimum-
path-sum) (97)
Valid Number
(../oj/valid-number)
Plus One (152)
(../oj/plus-one) (152)
Add Binary
(../oj/add-binary)
Text (151)
Justification
(../oj/text-
justification) (75)
Sqrt(x) (../oj/sqrtx)
Climbing Stairs (106)
(../oj/climbing-stairs)
Simplify Path (166)
(../oj/simplify-path)
Edit Distance (57)
(../oj/edit-distance)
Set Matrix Zeroes(67)
(../oj/set-matrix-
zeroes) (84)
Search a 2D Matrix
(../oj/search-a-2d-
matrix) (125)
Sort Colors
(../oj/sort-colors)
Minimum (134)
Window Substring
(../oj/minimum-
window-substring)
Combinations (95)
(../oj/combinations)
Subsets (133)
(../oj/subsets) (153)
Word Search
(../oj/word-search)
Remove (144)
Duplicates from
Sorted Array II

(../oj/remove-duplicates-from-sorted-array-ii)	(108)
Search in Rotated Sorted Array II (../oj/search-in-rotated-sorted-array-ii)	(47)
Remove Duplicates from Sorted List II (../oj/remove-duplicates-from-sorted-list-ii)	(121)
Remove Duplicates from Sorted List (../oj/remove-duplicates-from-sorted-list)	(119)
Largest Rectangle in Histogram (../oj/largest-rectangle-in-histogram)	(61)
Maximal Rectangle (../oj/maximal-rectangle)	(50)
Partition List (../oj/partition-list)	(95)
Scramble String (../oj/scramble-string)	(49)
Merge Sorted Array (../oj/merge-sorted-array)	(167)
Gray Code (../oj/gray-code)	(116)
Subsets II (../oj/subsets-ii)	(82)
Decode Ways (../oj/decode-ways)	(105)
Reverse Linked List II (../oj/reverse-linked-list-ii)	(121)
Restore IP Addresses (../oj/restore-ip-addresses)	(100)
Binary Tree Inorder Traversal (../oj/binary-tree-inorder-traversal)	(103)
Unique Binary Search Trees II (../oj/unique-binary-search-trees-ii)	(77)
Unique Binary Search Trees (../oj/unique-binary-search-trees)	(101)
Interleaving String (../oj/interleaving-string)	(101)
Validate Binary Search Tree (../oj/validate-binary-search-tree)	(175)

Recover Binary Search Tree (../oj/recover-binary-search-tree)	(79)
Same Tree (../oj/same-tree)	(140)
Symmetric Tree (../oj/symmetric-tree)	(160)
Binary Tree Level Order Traversal (../oj/binary-tree-level-order-traversal)	(139)
Binary Tree Zigzag Level Order Traversal (../oj/binary-tree-zigzag-level-order-traversal)	(111)
Maximum Depth of Binary Tree (../oj/maximum-depth-of-binary-tree)	(125)
Construct Binary Tree from Preorder and Inorder Traversal (../oj/construct-binary-tree-from-preorder-and-inorder-traversal)	(82)
Construct Binary Tree from Inorder and Postorder Traversal (../oj/construct-binary-tree-from-inorder-and-postorder-traversal)	(68)
Binary Tree Level Order Traversal II (../oj/binary-tree-level-order-traversal-ii)	(104)
Convert Sorted Array to Binary Search Tree (../oj/convert-sorted-array-to-binary-search-tree)	(95)
Convert Sorted List to Binary Search Tree (../oj/convert-sorted-list-to-binary-search-tree)	(96)
Balanced Binary Tree (../oj/balanced-binary-tree)	(161)
Minimum Depth of Binary Tree (../oj/minimum-depth-of-binary-tree)	(157)
Path Sum (../oj/path-sum)	(151)
Path Sum II (../oj/path-sum-ii)	(117)
Flatten Binary Tree to Linked List	

(../oj/flatten-binary-tree-to-linked-list)

Distinct (149) Subsequences (../oj/distinct-subsequences) (60)

Populating Next Right Pointers in Each Node (../oj/populating-next-right-pointers-in-each-node) (165)

Populating Next Right Pointers in Each Node II (../oj/populating-next-right-pointers-in-each-node-ii) (127)

Pascal's Triangle (../oj/pascals-triangle) (148)

Pascal's Triangle II (../oj/pascals-triangle-ii) (141)

Triangle (../oj/triangle) (133)

Best Time to Buy and Sell Stock (../oj/best-time-to-buy-and-sell-stock)

Best Time to (138) Buy and Sell Stock II (../oj/best-time-to-buy-and-sell-stock-ii)

Best Time to Buy (85) and Sell Stock III (../oj/best-time-to-buy-and-sell-stock-iii) (81)

Binary Tree Maximum Path Sum (../oj/binary-tree-maximum-path-sum)

Valid (100) Palindrome (../oj/valid-palindrome) (144)

Word Ladder II (../oj/word-ladder-ii)

Word Ladder (116) (../oj/word-ladder)

Longest (148) Consecutive Sequence (../oj/longest-consecutive-sequence) (115)

Sum Root to Leaf Numbers (../oj/sum-root-to-leaf-numbers) (109)

Surrounded Regions (../oj/surrounded-regions) (177)

Palindrome

Partitioning (../oj/palindrome-partitioning)	(112)
Palindrome Partitioning II (../oj/palindrome-partitioning-ii)	(66)
Clone Graph (../oj/clone-graph)	(115)
Gas Station (../oj/gas-station)	(109)
Candy (../oj/candy)	(125)
Single Number (../oj/single-number)	(106)
Single Number II (../oj/single-number-ii)	(94)
Copy List with Random Pointer (../oj/copy-list-with-random-pointer)	(142)
Word Break (../oj/word-break)	(166)
Word Break II (../oj/word-break-ii)	(162)
Linked List Cycle (../oj/linked-list-cycle)	(117)
Linked List Cycle II (../oj/linked-list-cycle-ii)	(110)
Reorder List (../oj/reorder-list)	(127)
Binary Tree Preorder Traversal (../oj/binary-tree-preorder-traversal)	(130)
Binary Tree Postorder Traversal (../oj/binary-tree-postorder-traversal)	(170)
LRU Cache (../oj/lru-cache)	(210)
Insertion Sort List (../oj/insertion-sort-list)	(143)
Sort List (../oj/sort-list)	(208)
Max Points on a Line (../oj/max-points-on-a-line)	(146)
Evaluate Reverse Polish Notation (../oj/evaluate-reverse-polish-notation)	(142)
Reverse Words in a String (../oj/reverse-words-in-a-string)	(247)
Maximum Product Subarray (../oj/maximum-product-subarray)	(108)
Find Minimum in Rotated Sorted	

Array (../oj/find-minimum-in-rotated-sorted-array)	(149)
Find Minimum in Rotated Sorted Array II (../oj/find-minimum-in-rotated-sorted-array-ii)	(98)
Min Stack (../oj/min-stack)	(209)
Binary Tree Upside Down (../oj/binary-tree-upside-down)	(30)
Read N Characters Given Read4 (../oj/read-n-characters-given-read4)	(38)
Read N Characters Given Read4 II - Call multiple times (../oj/read-n-characters-given-read4-ii-call-multiple-times)	(22)
Longest Substring with At Most Two Distinct Characters (../oj/longest-substring-with-at-most-two-distinct-characters)	(31)
Intersection of Two Linked Lists (../oj/intersection-of-two-linked-lists)	(170)
One Edit Distance (../oj/one-edit-distance)	(32)
Find Peak Element (../oj/find-peak-element)	(119)
Missing Ranges (../oj/missing-ranges)	(39)
Maximum Gap (../oj/maximum-gap)	(61)
Compare Version Numbers (../oj/compare-version-numbers)	(173)
Fraction to Recurring Decimal (../oj/fraction-to-recurring-decimal)	(69)
Two Sum II - Input array is sorted (../oj/two-sum-ii-input-array-is-sorted)	(12)
Excel Sheet Column Title (../oj/excel-sheet-column-title)	(105)
Majority Element (../oj/majority-element)	(154)
Two Sum III - Data	

structure design (../oj/two-sum-iii-data-structure-design)	(22)
Excel Sheet Column Number (../oj/excel-sheet-column-number)	(116)
Factorial Trailing Zeroes (../oj/factorial-trailing-zeroes)	(72)
Binary Search Tree Iterator (../oj/binary-search-tree-iterator)	(94)
Dungeon Game (../oj/dungeon-game)	(84)
Combine Two Tables (../oj/combine-two-tables)	(11)
Second Highest Salary (../oj/second-highest-salary)	(39)
Nth Highest Salary (../oj/nth-highest-salary)	(21)
Rank Scores (../oj/rank-scores)	(28)
Largest Number (../oj/largest-number)	(151)
Consecutive Numbers (../oj/consecutive-numbers)	(24)
Employees Earning More Than Their Managers (../oj/employees-earning-more-than-their-managers)	(19)
Duplicate Emails (../oj/duplicate-emails)	(20)
Customers Who Never Order (../oj/customers-who-never-order)	(16)
Department Highest Salary (../oj/department-highest-salary)	(37)
Department Top Three Salaries (../oj/department-top-three-salaries)	(37)
Reverse Words in a String II (../oj/reverse-words-in-a-string-ii)	(27)
Repeated DNA Sequences (../oj/repeated-dna-sequences)	(151)
Best Time to Buy and Sell Stock IV	

(../oj/best-time-to-buy-and-sell-stock-iv)	(56)
Rotate Array (../oj/rotate-array)	(248)
Reverse Bits (../oj/reverse-bits)	(185)
Number of 1 Bits (../oj/number-of-1-bits)	(182)
Word Frequency (../oj/word-frequency)	(17)
Valid Phone Numbers (../oj/valid-phone-numbers)	(13)
Transpose File (../oj/transpose-file)	(19)
Tenth Line (../oj/tenth-line)	(12)
Delete Duplicate Emails (../oj/delete-duplicate-emails)	(28)
Rising Temperature (../oj/rising-temperature)	(30)
House Robber (../oj/house-robber)	(141)
Binary Tree Right Side View (../oj/binary-tree-right-side-view)	(144)
Number of Islands (../oj/number-of-islands)	(149)
Bitwise AND of Numbers Range (../oj/bitwise-and-of-numbers-range)	(94)
Happy Number (../oj/happy-number)	(164)
Remove Linked List Elements (../oj/remove-linked-list-elements)	(129)
Count Primes (../oj/count-primes)	(105)
Isomorphic Strings (../oj/isomorphic-strings)	(145)
Reverse Linked List (../oj/reverse-linked-list)	(161)
Course Schedule (../oj/course-schedule)	(126)
Implement Trie (Prefix Tree) (../oj/implement-trie-prefix-tree)	(100)
Minimum Size Subarray Sum (../oj/minimum-size-subarray-sum)	(105)
Course Schedule II	

(../oj/course-schedule-ii)	(93)
Add and Search Word - Data structure design (../oj/add-and-search-word-data-structure-design)	(80)
Word Search II (../oj/word-search-ii)	(58)
House Robber II (../oj/house-robber-ii)	(79)
Shortest Palindrome (../oj/shortest-palindrome)	(90)
Kth Largest Element in an Array (../oj/kth-largest-element-in-an-array)	(98)
Combination Sum III (../oj/combination-sum-iii)	(96)
Contains Duplicate (../oj/contains-duplicate)	(110)
The Skyline Problem (../oj/the-skyline-problem)	(56)
Contains Duplicate II (../oj/contains-duplicate-ii)	(116)
Contains Duplicate III (../oj/contains-duplicate-iii)	(68)
Maximal Square (../oj/maximal-square)	(66)
Count Complete Tree Nodes (../oj/count-complete-tree-nodes)	(93)
Rectangle Area (../oj/rectangle-area)	(98)
Basic Calculator (../oj/basic-calculator)	(88)
Implement Stack using Queues (../oj/implement-stack-using-queues)	(103)
Invert Binary Tree (../oj/invert-binary-tree)	(129)
Basic Calculator II (../oj/basic-calculator-ii)	(77)
Summary Ranges (../oj/summary-ranges)	(133)
Majority Element II (../oj/majority-element-ii)	(73)
Kth Smallest Element in a BST (../oj/kth-	

smallest-element-in-
a-bst) (127)
Power of Two
(../oj/power-of-two)
Implement (100)
Queue using
Stacks
(../oj/implement-
queue-using-stacks)
Number of Digit (79)
One (../oj/number-of-
digit-one) (60)
Palindrome Linked
List (../oj/palindrome-
linked-list) (180)
Lowest Common
Ancestor of a Binary
Search Tree
(../oj/lowest-
common-ancestor-
of-a-binary-search-
tree) (121)
Lowest Common
Ancestor of a Binary
Tree (../oj/lowest-
common-ancestor-
of-a-binary-tree) (95)
Delete Node in a
Linked List
(../oj/delete-node-in-
a-linked-list) (77)
Product of Array
Except Self
(../oj/product-of-
array-except-self)(96)
Sliding Window
Maximum
(../oj/sliding-window-
maximum) (92)
Search a 2D Matrix II
(../oj/search-a-2d-
matrix-ii) (92)
Different Ways to
Add Parentheses
(../oj/different-ways-
to-add-parentheses)
Valid Anagram (65)
(../oj/valid-anagram)
Shortest Word (142)
Distance
(../oj/shortest-word-
distance) (24)
Shortest Word
Distance II
(../oj/shortest-word-
distance-ii) (18)
Shortest Word
Distance III
(../oj/shortest-word-
distance-iii) (25)
Strobogrammatic
Number
(../oj/strobogrammatic-
number) (22)
Strobogrammatic
Number II

(../oj/strobogrammatic-number-ii)	(34)
Strobogrammatic Number III (../oj/strobogrammatic-number-iii)	(24)
Group Shifted Strings (../oj/group-shifted-strings)	(36)
Count Univalue Subtrees (../oj/count-univalue-subtrees)	(29)
Flatten 2D Vector (../oj/flatten-2d-vector)	(34)
Meeting Rooms (../oj/meeting-rooms)	(20)
Meeting Rooms II (../oj/meeting-rooms-ii)	(47)
Factor Combinations (../oj/factor-combinations)	(30)
Verify Preorder Sequence in Binary Search Tree (../oj/verify-preorder-sequence-in-binary-search-tree)	(14)
Paint House (../oj/paint-house)	(18)
Binary Tree Paths (../oj/binary-tree-paths)	(121)
Add Digits (../oj/add-digits)	(90)
3Sum Smaller (../oj/3sum-smaller)	(31)
Single Number III (../oj/single-number-iii)	(43)
Graph Valid Tree (../oj/graph-valid-tree)	(50)
Trips and Users (../oj/trips-and-users)	(17)
Ugly Number (../oj/ugly-number)	(56)
Ugly Number II (../oj/ugly-number-ii)	(59)
Paint House II (../oj/paint-house-ii)	(32)
Palindrome Permutation (../oj/palindrome-permutation)	(23)
Palindrome Permutation II (../oj/palindrome-permutation-ii)	(25)
Missing Number (../oj/missing-number)	(127)
Alien Dictionary (../oj/alien-dictionary)	(51)

Closest Binary Search Tree Value (../oj/closest-binary-search-tree-value)	(24)
Encode and Decode Strings (../oj/encode-and-decode-strings)	(29)
Closest Binary Search Tree Value II (../oj/closest-binary-search-tree-value-ii)	(46)
Integer to English Words (../oj/integer-to-english-words)	(60)
H-Index (../oj/h-index)	(116)
H-Index II (../oj/h-index-ii)	(38)
Paint Fence (../oj/paint-fence)	(31)
Find the Celebrity (../oj/find-the-celebrity)	(20)
First Bad Version (../oj/first-bad-version)	(76)
Perfect Squares (../oj/perfect-squares)	(83)
Wiggle Sort (../oj/wiggle-sort)	(23)
Zigzag Iterator (../oj/zigzag-iterator)	(42)
Expression Add Operators (../oj/expression-add-operators)	(37)
Move Zeroes (../oj/move-zeroes)	(172)
Peeking Iterator (../oj/peeking-iterator)	(54)
Inorder Successor in BST (../oj/inorder-successor-in-bst)	(28)
Walls and Gates (../oj/walls-and-gates)	(39)
Find the Duplicate Number (../oj/find-the-duplicate-number)	(75)
Unique Word Abbreviation (../oj/unique-word-abbreviation)	(30)
Game of Life (../oj/game-of-life)	(63)
Word Pattern (../oj/word-pattern)	(100)
Word Pattern II (../oj/word-pattern-ii)	(26)
Nim Game (../oj/nim-game)	(59)

Flip Game (../oj/flip-game)	(21)
Flip Game II (../oj/flip-game-ii)	(25)
Find Median from Data Stream (../oj/find-median-from-data-stream)	(54)
Best Meeting Point (../oj/best-meeting-point)	(22)
Serialize and Deserialize Binary Tree (../oj/serialize-and-deserialize-binary-tree)	(52)
Binary Tree Longest Consecutive Sequence (../oj/binary-tree-longest-consecutive-sequence)	(37)
Bulls and Cows (../oj/bulls-and-cows)	(69)
Longest Increasing Subsequence (../oj/longest-increasing-subsequence)	(64)
Remove Invalid Parentheses (../oj/remove-invalid-parentheses)	(38)
Smallest Rectangle Enclosing Black Pixels (../oj/smallest-rectangle-enclosing-black-pixels)	(19)
Range Sum Query - Immutable (../oj/range-sum-query-immutable)	(62)
Range Sum Query 2D - Immutable (../oj/range-sum-query-2d-immutable)	(53)
Number of Islands II (../oj/number-of-islands-ii)	(29)
Additive Number (../oj/additive-number)	(53)
Range Sum Query - Mutable (../oj/range-sum-query-mutable)	(55)
Range Sum Query 2D - Mutable (../oj/range-sum-query-2d-mutable)	(17)
Best Time to Buy and Sell Stock with Cooldown (../oj/best-time-to-buy-and-sell-stock-with-	

cooldown)	(42)
Minimum Height Trees (../oj/minimum-height-trees)	(29)
Sparse Matrix Multiplication (../oj/sparse-matrix-multiplication)	(19)
Burst Balloons (../oj/burst-balloons)	(12)
Super Ugly Number (../oj/super-ugly-number)	(40)
Binary Tree Vertical Order Traversal (../oj/binary-tree-vertical-order-traversal)	(30)
Count of Smaller Numbers After Self (../oj/count-of-smaller-numbers-after-self)	(42)
Remove Duplicate Letters (../oj/remove-duplicate-letters)	(34)
Shortest Distance from All Buildings (../oj/shortest-distance-from-all-buildings)	(21)
Maximum Product of Word Lengths (../oj/maximum-product-of-word-lengths)	(39)
Bulb Switcher (../oj/bulb-switcher)	(24)
Generalized Abbreviation (../oj/generalized-abbreviation)	(35)
Create Maximum Number (../oj/create-maximum-number)	(28)
Coin Change (../oj/coin-change)	(47)
Number of Connected Components in an Undirected Graph (../oj/number-of-connected-components-in-an-undirected-graph)	(30)
Wiggle Sort II (../oj/wiggle-sort-ii)	(35)
Maximum Size Subarray Sum Equals k (../oj/maximum-size-subarray-sum-equals-k)	(12)
Power of Three (../oj/power-of-three)	

Count of Range (34)
Sum (../oj/count-of-range-sum) (14)

Interview Questions (../interview-questions) (145)