LeetCode
Online Portal for IT Interview

0 moonlin

About Activity Discuss Members Online Judge

Online Judge

Search

Below are a collection of questions for you to practice. Click on the question's title to expand the question description. Read the question and try to solve it by clicking on the *"Solve this problem"* link. You may start typing your code in the coding panel (bottom right side).

Once done, run your solution against the judge's secret input to see if you've solved it correctly. It's that easy!

Happy coding and remember to Follow or Like LeetCode and get the latest update when a new question is added!

- Facebook
- Twitter

IMPORTANT:
The *Solution* object is instantiated only *once* and is reused for each test case input. ***When declaring a class member variable, be extra cautious and remember to reset the variable!***

Welcome, *moonlin*! You have solved **126 / 132** problems.

Questions List: *(Click on title to expand)* sort questions by: [title] [freshness]

You are solving problem "Sum Nodes in Lists". » Click here to switch back to interrupted problem show LeetCode coding panel ↑

✔ **3Sum Closest** 6070 / 15852 Jan 18 '12

Given an array *S* of *n* integers, find three integers in *S* such that the sum is closest to a given number, target. Return the sum of the three integers. You may assume that each input would have exactly one solution.

```
For example, given array S = {-1 2 1 -4}, and target = 1.

The sum that is closest to the target is 2. (-1 + 2 + 1 = 2).
```

» Solve this problem (link to this question)

✔ **3Sum** 9830 / 38290 Jan 18 '12

Given an array *S* of *n* integers, are there elements *a*, *b*, *c* in *S* such that *a* + *b* + *c* = 0? Find all unique triplets in the array which gives the sum of zero.

Note:

- Elements in a triplet (*a*, *b*, *c*) must be in non-descending order. (ie, *a* ? *b* ? *c*)
- The solution set must not contain duplicate triplets.

```
For example, given array S = {-1 0 1 2 -1 -4},

A solution set is:
(-1, 0, 1)
(-1, -1, 2)
```

» Solve this problem (link to this question)

✔ **4Sum** 6382 / 19016 Jan 27 '12

Given an array *S* of *n* integers, are there elements *a*, *b*, *c*, and *d* in *S* such that *a* + *b* + *c* + *d* = target? Find all unique quadruplets in the array which gives the sum of target.

Note:

- Elements in a quadruplet (*a*, *b*, *c*, *d*) must be in non-descending order. (ie, *a* ? *b* ? *c* ? *d*)
- The solution set must not contain duplicate quadruplets.

```
For example, given array S = {1 0 -1 0 -2 2}, and target = 0.

A solution set is:
(-1,  0, 0, 1)
(-2, -1, 1, 2)
(-2,  0, 0, 2)
```

» Solve this problem (link to this question)

✅ **Add Binary** 5763 / 16556 Apr 2 '12

Given two binary strings, return their sum (also a binary string).

For example,
a = `"11"`
b = `"1"`
Return `"100"`.

» Solve this problem (link to this question)

✅ **Add Two Numbers** 9607 / 32573 Nov 1 '11

You are given two linked lists representing two non-negative numbers. The digits are stored in reverse order and each of their nodes contain a single digit. Add the two numbers and return it as a linked list.

Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)
Output: 7 -> 0 -> 8

» Solve this problem (link to this question)

✅ **Anagrams** 5542 / 18397 Mar 19 '12

Given an array of strings, return all groups of strings that are anagrams.

Note: All inputs will be in lower-case.

» Solve this problem (link to this question)

✅ **Balanced Binary Tree** 7722 / 18479 Oct 9 '12

Given a binary tree, determine if it is height-balanced.

For this problem, a height-balanced binary tree is defined as a binary tree in which the depth of the two subtrees of *every* node never differ by more than 1.

» Solve this problem (link to this question)

✅ **Best Time to Buy and Sell Stock III** 6064 / 18525 Nov 7 '12

Say you have an array for which the i^{th} element is the price of a given stock on day i.

Design an algorithm to find the maximum profit. You may complete at most *two* transactions.

Note:
You may not engage in multiple transactions at the same time (ie, you must sell the stock before you buy again).

» Solve this problem (link to this question)

✅ **Best Time to Buy and Sell Stock II** 6007 / 11762 Oct 31 '12

Say you have an array for which the i^{th} element is the price of a given stock on day i.

Design an algorithm to find the maximum profit. You may complete as many transactions as you like (ie, buy one and sell one share of the stock multiple times). However, you may not engage in multiple transactions at the same time (ie, you must sell the stock before you buy again).

» Solve this problem (link to this question)

✅ **Best Time to Buy and Sell Stock** 7527 / 16932 Oct 30 '12

Say you have an array for which the i^{th} element is the price of a given stock on day i.

If you were only permitted to complete at most one transaction (ie, buy one and sell one share of the stock), design an algorithm to find the maximum profit.

» Solve this problem (link to this question)

✅ **Binary Tree Inorder Traversal** 7385 / 16684 Aug 27 '12

Given a binary tree, return the *inorder* traversal of its nodes' values.

For example:
Given binary tree `{1,#,2,3}`,

```
   1
    \
     2
    /
```

```
   3
```

return `[1,3,2]`.

Note: Recursive solution is trivial, could you do it iteratively?

confused what `"{1,#,2,3}"` means? > read more on how binary tree is serialized on OJ.

» Solve this problem (link to this question)

✅ **Binary Tree Level Order Traversal II** 4449 / 10092 Oct 1 '12

Given a binary tree, return the *bottom-up level order* traversal of its nodes' values. (ie, from left to right, level by level from leaf to root).

For example:
Given binary tree `{3,9,20,#,#,15,7}`,

```
    3
   / \
  9  20
    /  \
   15   7
```

return its bottom-up level order traversal as:

```
[
  [15,7]
  [9,20],
  [3],
]
```

confused what `"{1,#,2,3}"` means? > read more on how binary tree is serialized on OJ.

» Solve this problem (link to this question)

✅ **Binary Tree Level Order Traversal** 5832 / 14746 Sep 29 '12

Given a binary tree, return the *level order* traversal of its nodes' values. (ie, from left to right, level by level).

For example:
Given binary tree `{3,9,20,#,#,15,7}`,

```
    3
   / \
  9  20
    /  \
   15   7
```

return its level order traversal as:

```
[
  [3],
  [9,20],
  [15,7]
]
```

confused what `"{1,#,2,3}"` means? > read more on how binary tree is serialized on OJ.

» Solve this problem (link to this question)

✅ **Binary Tree Maximum Path Sum** 7312 / 26768 Nov 8 '12

Given a binary tree, find the maximum path sum.

The path may start and end at any node in the tree.

For example:
Given the below binary tree,

```
       1
      / \
     2   3
```

Return `6`.

» Solve this problem (link to this question)

✅ **Binary Tree Zigzag Level Order Traversal** 4378 / 12129 Sep 29 '12

Given a binary tree, return the *zigzag level order* traversal of its nodes' values. (ie, from left to right, then right to left for the next level and alternate between).

For example:
Given binary tree `{3,9,20,#,#,15,7}`,

```
    3
   / \
  9  20
    /  \
   15   7
```

return its zigzag level order traversal as:

```
[
  [3],
  [20,9],
  [15,7]
]
```

confused what `"{1,#,2,3}"` means? > read more on how binary tree is serialized on OJ.

» Solve this problem (link to this question)

✅ **Climbing Stairs** 6879 / 13083 Apr 3 '12

You are climbing a stair case. It takes *n* steps to reach to the top.

Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

» Solve this problem (link to this question)

✅ **Combination Sum II** 4784 / 14324 Mar 7 '12

Given a collection of candidate numbers (***C***) and a target number (***T***), find all unique combinations in ***C*** where the candidate numbers sums to ***T***.

Each number in ***C*** may only be used **once** in the combination.

Note:

- All numbers (including target) will be positive integers.
- Elements in a combination (a_1, a_2, …, a_k) must be in non-descending order. (ie, $a_1 \le a_2 \le \ldots \le a_k$).
- The solution set must not contain duplicate combinations.

For example, given candidate set `10,1,2,7,6,1,5` and target `8`,
A solution set is:
`[1, 7]`
`[1, 2, 5]`
`[2, 6]`
`[1, 1, 6]`

» Solve this problem (link to this question)

✅ **Combination Sum** 6099 / 17151 Mar 7 '12

Given a set of candidate numbers (***C***) and a target number (***T***), find all unique combinations in ***C*** where the candidate numbers sums to ***T***.

The **same** repeated number may be chosen from ***C*** unlimited number of times.

Note:

- All numbers (including target) will be positive integers.
- Elements in a combination (a_1, a_2, …, a_k) must be in non-descending order. (ie, $a_1 \le a_2 \le \ldots \le a_k$).
- The solution set must not contain duplicate combinations.

For example, given candidate set `2,3,6,7` and target `7`,
A solution set is:
`[7]`
`[2, 2, 3]`

» Solve this problem (link to this question)

✔ **Combinations** 5708 / 14105 Apr 18 '12

Given two integers n and k, return all possible combinations of k numbers out of $1 \dots n$.

For example,
If $n = 4$ and $k = 2$, a solution is:

```
[
  [2,4],
  [3,4],
  [2,3],
  [1,2],
  [1,3],
  [1,4],
]
```

» Solve this problem (link to this question)

✔ **Construct Binary Tree from Inorder and Postorder Traversal** 4493 / 12665 Sep 30 '12

Given inorder and postorder traversal of a tree, construct the binary tree.

Note:
You may assume that duplicates do not exist in the tree.

» Solve this problem (link to this question)

✔ **Construct Binary Tree from Preorder and Inorder Traversal** 4869 / 14001 Sep 30 '12

Given preorder and inorder traversal of a tree, construct the binary tree.

Note:
You may assume that duplicates do not exist in the tree.

» Solve this problem (link to this question)

✔ **Container With Most Water** 6382 / 14504 Jan 9 '12

Given n non-negative integers a_1, a_2, ..., a_n, where each represents a point at coordinate (i, a_i). n vertical lines are drawn such that the two endpoints of line i is at (i, a_i) and $(i, 0)$. Find two lines, which together with x-axis forms a container, such that the container contains the most water.

Note: You may not slant the container.

» Solve this problem (link to this question)

✔ **Convert Sorted Array to Binary Search Tree** 5459 / 11289 Oct 2 '12

Given an array where elements are sorted in ascending order, convert it to a height balanced BST.

» Solve this problem (link to this question)

✔ **Convert Sorted List to Binary Search Tree** 5768 / 16298 Oct 3 '12

Given a singly linked list where elements are sorted in ascending order, convert it to a height balanced BST.

» Solve this problem (link to this question)

✔ **Count and Say** 4871 / 13011 Mar 6 '12

The count-and-say sequence is the sequence of integers beginning as follows:
`1, 11, 21, 1211, 111221, ...`

`1` is read off as `"one 1"` or `11`.
`11` is read off as `"two 1s"` or `21`.
`21` is read off as `"one 2"`, then `one 1"` or `1211`.

Given an integer n, generate the n^{th} sequence.

Note: The sequence of integers will be represented as a string.

» Solve this problem (link to this question)

✔ **Decode Ways** 6747 / 26583 Jun 25 '12

A message containing letters from `A-Z` is being encoded to numbers using the following mapping:

```
'A' -> 1
'B' -> 2
```

```
...
'Z' -> 26
```

Given an encoded message containing digits, determine the total number of ways to decode it.

For example,
Given encoded message `"12"`, it could be decoded as `"AB"` (1 2) or `"L"` (12).

The number of ways decoding `"12"` is 2.

» Solve this problem (link to this question)

✔ **Distinct Subsequences** 6266 / 17972 Oct 19 '12

Given a string `S` and a string `T`, count the number of distinct subsequences of `T` in `S`.

A subsequence of a string is a new string which is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (ie, `"ACE"` is a subsequence of `"ABCDE"` while `"AEC"` is not).

Here is an example:
`S` = `"rabbbit"`, `T` = `"rabbit"`

Return `3`.

» Solve this problem (link to this question)

Divide Two Integers 6447 / 27837 Feb 18 '12

Divide two integers without using multiplication, division and mod operator.

» Solve this problem (link to this question)

✔ **Edit Distance** 5171 / 14897 Apr 4 '12

Given two words *word1* and *word2*, find the minimum number of steps required to convert *word1* to *word2*. (each operation is counted as 1 step.)

You have the following 3 operations permitted on a word:

a) Insert a character
b) Delete a character
c) Replace a character

» Solve this problem (link to this question)

✔ **First Missing Positive** 5758 / 17962 Mar 8 '12

Given an unsorted integer array, find the first missing positive integer.

For example,
Given `[1,2,0]` return `3`,
and `[3,4,-1,1]` return `2`.

Your algorithm should run in $O(n)$ time and uses constant space.

» Solve this problem (link to this question)

✔ **Flatten Binary Tree to Linked List** 7105 / 21371 Oct 14 '12

Given a binary tree, flatten it to a linked list in-place.

For example,
Given

```
      1
     / \
    2   5
   / \   \
  3   4   6
```

The flattened tree should look like:

```
1
 \
  2
   \
    3
     \
      4
       \
```

```
   5
    \
     6
```

click to show hints.

» Solve this problem (link to this question)

✔ **Generate Parentheses** 6413 / 14970 Feb 13 '12

Given *n* pairs of parentheses, write a function to generate all combinations of well-formed parentheses.

For example, given *n* = 3, a solution set is:

`"((()))", "(()())", "(())()", "()(())", "()()()"`

» Solve this problem (link to this question)

✔ **Gray Code** 3727 / 8371 May 20 '12

The gray code is a binary numeral system where two successive values differ in only one bit.

Given a non-negative integer *n* representing the total number of bits in the code, print the sequence of gray code. A gray code sequence must begin with 0.

For example, given *n* = 2, return `[0,1,3,2]`. Its gray code sequence is:

```
00 - 0
01 - 1
11 - 3
10 - 2
```

Note:
For a given *n*, a gray code sequence is not uniquely defined.

For example, `[0,2,3,1]` is also a valid gray code sequence according to the above definition.

For now, the judge is able to judge based on one instance of gray code sequence. Sorry about that.

» Solve this problem (link to this question)

✔ **Implement strStr()** 7053 / 22390 Feb 18 '12

Implement strStr().

Returns a pointer to the first occurrence of needle in haystack, or **null** if needle is not part of haystack.

» Solve this problem (link to this question)

✔ **Insert Interval** 5301 / 18671 Mar 27 '12

Given a set of *non-overlapping* intervals, insert a new interval into the intervals (merge if necessary).

You may assume that the intervals were initially sorted according to their start times.

Example 1:
Given intervals `[1,3],[6,9]`, insert and merge `[2,5]` in as `[1,5],[6,9]`.

Example 2:
Given `[1,2],[3,5],[6,7],[8,10],[12,16]`, insert and merge `[4,9]` in as `[1,2],[3,10],[12,16]`.

This is because the new interval `[4,9]` overlaps with `[3,5],[6,7],[8,10]`.

» Solve this problem (link to this question)

Integer to Roman 4166 / 10195 Jan 15 '12

Given an integer, convert it to a roman numeral.

Input is guaranteed to be within the range from 1 to 3999.

» Solve this problem (link to this question)

✔ **Interleaving String** 7096 / 24208 Aug 31 '12

Given *s1*, *s2*, *s3*, find whether *s3* is formed by the interleaving of *s1* and *s2*.

For example,

Given:
s1 = `"aabcc"`,
s2 = `"dbbca"`,

When *s3* = `"aadbbcbcac"`, return true.
When *s3* = `"aadbbbaccc"`, return false.

» Solve this problem (link to this question)

✔ Jump Game II

7347 / 20162 Mar 17 '12

Given an array of non-negative integers, you are initially positioned at the first index of the array.

Each element in the array represents your maximum jump length at that position.

Your goal is to reach the last index in the minimum number of jumps.

For example:
Given array A = `[2,3,1,1,4]`

The minimum number of jumps to reach the last index is `2`. (Jump `1` step from index 0 to 1, then `3` steps to the last index.)

» Solve this problem (link to this question)

✔ Jump Game

7493 / 17569 Mar 25 '12

Given an array of non-negative integers, you are initially positioned at the first index of the array.

Each element in the array represents your maximum jump length at that position.

Determine if you are able to reach the last index.

For example:
A = `[2,3,1,1,4]`, return `true`.

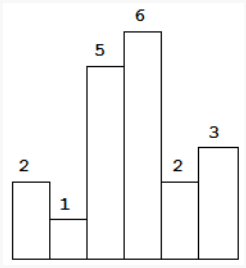
A = `[3,2,1,0,4]`, return `false`.

» Solve this problem (link to this question)

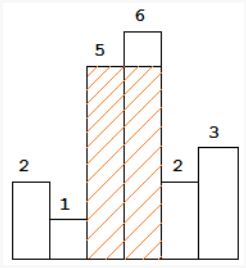
✔ Largest Rectangle in Histogram

6550 / 21145 Apr 23 '12

Given *n* non-negative integers representing the histogram's bar height where the width of each bar is 1, find the area of largest rectangle in the histogram.



Above is a histogram where width of each bar is 1, given height = `[2,1,5,6,2,3]`.



The largest rectangle is shown in the shaded area, which has area = `10` unit.

For example,
Given height = `[2,1,5,6,2,3]`,
return `10`.

» Solve this problem (link to this question)

✔ Length of Last Word

5030 / 12158 Mar 27 '12

Given a string *s* consists of upper/lower-case alphabets and empty space characters ` `, return the length of last word in the string.

If the last word does not exist, return 0.

Note: A word is defined as a character sequence consists of non-space characters only.

For example,
Given *s* = `"Hello World"`,
return `5`.

» Solve this problem (link to this question)

✅ **Letter Combinations of a Phone Number** 5504 / 16698 Jan 27 '12

Given a digit string, return all possible letter combinations that the number could represent.

A mapping of digit to letters (just like on the telephone buttons) is given below.

Input: Digit string "23"
Output: ["ad", "ae", "af", "bd", "be", "bf", "cd", "ce", "cf"].

Note:
Although the above answer is in lexicographical order, your answer could be in any order you want.

» Solve this problem (link to this question)

✅ **Longest Common Prefix** 6659 / 18460 Jan 17 '12

Write a function to find the longest common prefix string amongst an array of strings.

» Solve this problem (link to this question)

✅ **Longest Consecutive Sequence** 7913 / 22936 Feb 14

Given an unsorted array of integers, find the length of the longest consecutive elements sequence.

For example,
Given `[100, 4, 200, 1, 3, 2]`,
The longest consecutive elements sequence is `[1, 2, 3, 4]`. Return its length: `4`.

Your algorithm should run in O(*n*) complexity.

» Solve this problem (link to this question)

✅ **Longest Palindromic Substring** 8767 / 28389 Nov 11 '11

Given a string *S*, find the longest palindromic substring in *S*. You may assume that the maximum length of *S* is 1000, and there exists one unique longest palindromic substring.

» Solve this problem (link to this question)

✅ **Longest Substring Without Repeating Characters** 11147 / 33568 May 16 '11

Given a string, find the length of the longest substring without repeating characters. For example, the longest substring without repeating letters for "abcabcbb" is "abc", which the length is 3. For "bbbbb" the longest substring is "b", with the length of 1.

» Solve this problem (link to this question)

✅ **Longest Valid Parentheses** 6113 / 22070 Mar 1 '12

Given a string containing just the characters `'('` and `')'`, find the length of the longest valid (well-formed) parentheses substring.

For `"(()"`, the longest valid parentheses substring is `"()"`, which has length = 2.

Another example is `")()())"`, where the longest valid parentheses substring is `"()()"`, which has length = 4.

» Solve this problem (link to this question)

✅ **Maximal Rectangle** 3792 / 13475 Apr 24 '12

Given a 2D binary matrix filled with 0's and 1's, find the largest rectangle containing all ones

and return its area.

» Solve this problem (link to this question)

✔ **Maximum Depth of Binary Tree** 6343 / 9043 Sep 30 '12

Given a binary tree, find its maximum depth.

The maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.

» Solve this problem (link to this question)

✔ **Maximum Subarray** 5966 / 12657 Mar 21 '12

Find the contiguous subarray within an array (containing at least one number) which has the largest sum.

For example, given the array `[-2,1,-3,4,-1,2,1,-5,4]`,
the contiguous subarray `[4,-1,2,1]` has the largest sum = `6`.

click to show more practice.

» Solve this problem (link to this question)

✔ **Median of Two Sorted Arrays** 7725 / 38502 Mar 28 '11

There are two sorted arrays A and B of size m and n respectively. Find the median of the two sorted arrays. The overall run time complexity should be O(log (m+n)).

» Solve this problem (link to this question)

✔ **Merge Intervals** 5655 / 20100 Mar 27 '12

Given a collection of intervals, merge all overlapping intervals.

For example,
Given `[1,3],[2,6],[8,10],[15,18]`,
return `[1,6],[8,10],[15,18]`.

» Solve this problem (link to this question)

✔ **Merge k Sorted Lists** 6213 / 20674 Feb 14 '12

Merge *k* sorted linked lists and return it as one sorted list. Analyze and describe its complexity.

» Solve this problem (link to this question)

✔ **Merge Sorted Array** 6055 / 13640 May 20 '12

Given two sorted integer arrays A and B, merge B into A as one sorted array.

Note:
You may assume that A has enough space to hold additional elements from B. The number of elements initialized in A and B are *m* and *n* respectively.

» Solve this problem (link to this question)

✔ **Merge Two Sorted Lists** 6381 / 13388 Mar 30 '12

Merge two sorted linked lists and return it as a new list. The new list should be made by splicing together the nodes of the first two lists.

» Solve this problem (link to this question)

✔ **Minimum Depth of Binary Tree** 7310 / 18192 Oct 10 '12

Given a binary tree, find its minimum depth.

The minimum depth is the number of nodes along the shortest path from the root node down to the nearest leaf node.

» Solve this problem (link to this question)

✔ **Minimum Path Sum** 4584 / 10117 Mar 29 '12

Given a *m* x *n* grid filled with non-negative numbers, find a path from top left to bottom right which *minimizes* the sum of all numbers along its path.

Note: You can only move either down or right at any point in time.

» Solve this problem (link to this question)

✔ **Minimum Window Substring** 4849 / 20420 Apr 15 '12

Given a string S and a string T, find the minimum window in S which will contain all the

characters in T in complexity O(n).

For example,
S = `"ADOBECODEBANC"`
T = `"ABC"`

Minimum window is `"BANC"`.

Note:
If there is no such window in S that covers all characters in T, return the emtpy string `""`.

If there are multiple such windows, you are guaranteed that there will always be only one unique minimum window in S.

» Solve this problem (link to this question)

✔ **Multiply Strings** 4589 / 17324 Mar 12 '12

Given two numbers represented as strings, return multiplication of the numbers as a string.

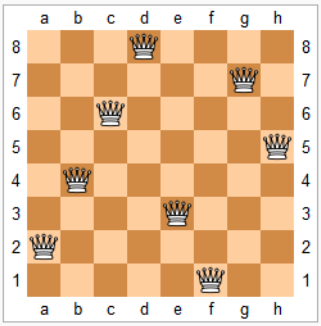
Note: The numbers can be arbitrarily large and are non-negative.

» Solve this problem (link to this question)

✔ **N-Queens II** 4720 / 10565 Mar 20 '12

Follow up for N-Queens problem.

Now, instead outputting board configurations, return the total number of distinct solutions.

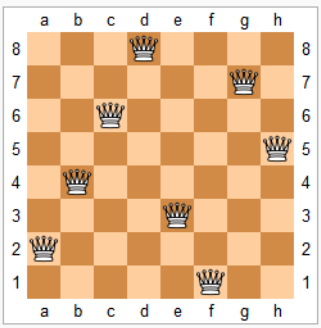


One solution to the eight queens puzzle

» Solve this problem (link to this question)

✔ **N-Queens** 3976 / 12479 Mar 20 '12

The *n*-queens puzzle is the problem of placing *n* queens on an *n*×*n* chessboard such that no two queens attack each other.



One solution to the eight queens puzzle

Given an integer *n*, return all distinct solutions to the *n*-queens puzzle.

Each solution contains a distinct board configuration of the *n*-queens' placement, where `'Q'` and `'.'` both indicate a queen and an empty space respectively.

For example,
There exist two distinct solutions to the 4-queens puzzle:

```
[
 [".Q..",  // Solution 1
  "...Q",
  "Q...",
  "..Q."],

 ["..Q.",  // Solution 2
```

```
  "Q...",
  "...Q",
  ".Q.."]
]
```

» Solve this problem (link to this question)

✅ **Next Permutation** 4805 / 13610 Feb 25 '12

Implement next permutation, which rearranges numbers into the lexicographically next greater permutation of numbers.

If such arrangement is not possible, it must rearrange it as the lowest possible order (ie, sorted in ascending order).

The replacement must be in-place, do not allocate extra memory.

Here are some examples. Inputs are in the left-hand column and its corresponding outputs are in the right-hand column.
`1,2,3` → `1,3,2`
`3,2,1` → `1,2,3`
`1,1,5` → `1,5,1`

» Solve this problem (link to this question)

✅ **Palindrome Number** 7619 / 19000 Jan 4 '12

Determine whether an integer is a palindrome. Do this without extra space.

click to show spoilers.

» Solve this problem (link to this question)

✅ **Palindrome Partitioning II** 13297 / 47208 Mar 1

Given a string *s*, partition *s* such that every substring of the partition is a palindrome.

Return the minimum cuts needed for a palindrome partitioning of *s*.

For example, given *s* = `"aab"`,
Return `1` since the palindrome partitioning `["aa","b"]` could be produced using 1 cut.

» Solve this problem (link to this question)

✅ **Palindrome Partitioning** 7285 / 24780 Feb 28

Given a string *s*, partition *s* such that every substring of the partition is a palindrome.

Return all possible palindrome partitioning of *s*.

For example, given *s* = `"aab"`,
Return

```
[
  ["aa","b"],
  ["a","a","b"]
]
```

» Solve this problem (link to this question)

✅ **Partition List** 5164 / 15877 Apr 30 '12

Given a linked list and a value *x*, partition it such that all nodes less than *x* come before nodes greater than or equal to *x*.

You should preserve the original relative order of the nodes in each of the two partitions.

For example,
Given `1->4->3->2->5->2` and *x* = 3,
return `1->2->2->4->3->5`.

» Solve this problem (link to this question)

✅ **Pascal's Triangle II** 5210 / 12287 Oct 29 '12

Given an index *k*, return the k^{th} row of the Pascal's triangle.

For example, given *k* = 3,
Return `[1,3,3,1]`.

Note:
Could you optimize your algorithm to use only $O(k)$ extra space?

» Solve this problem (link to this question)

Pascal's Triangle

4857 / 10981 Oct 28 '12

Given *numRows*, generate the first *numRows* of Pascal's triangle.

For example, given *numRows* = 5,
Return

```
[
     [1],
    [1,1],
   [1,2,1],
  [1,3,3,1],
 [1,4,6,4,1]
]
```

» Solve this problem (link to this question)

Path Sum II

6392 / 17591 Oct 14 '12

Given a binary tree and a sum, find all root-to-leaf paths where each path's sum equals the given sum.

For example:
Given the below binary tree and `sum = 22`,

```
              5
             / \
            4   8
           /   / \
          11  13  4
         /  \    / \
        7    2  5   1
```

return

```
[
   [5,4,11,2],
   [5,8,4,5]
]
```

» Solve this problem (link to this question)

Path Sum

6653 / 15901 Oct 14 '12

Given a binary tree and a sum, determine if the tree has a root-to-leaf path such that adding up all the values along the path equals the given sum.

For example:
Given the below binary tree and `sum = 22`,

```
              5
             / \
            4   8
           /   / \
          11  13  4
         /  \      \
        7    2      1
```

return true, as there exist a root-to-leaf path `5->4->11->2` which sum is 22.

» Solve this problem (link to this question)

Permutation Sequence

4190 / 14555 Mar 28 '12

The set `[1,2,3,…,n]` contains a total of *n*! unique permutations.

By listing and labeling all of the permutations in order,
We get the following sequence (ie, for *n* = 3):

- `"123"`
- `"132"`
- `"213"`
- `"231"`
- `"312"`
- `"321"`

Given n and k, return the k^{th} permutation sequence.

Note: Given n will be between 1 and 9 inclusive.

» Solve this problem (link to this question)

✅ **Permutations II** 5365 / 13887 Mar 17 '12

Given a collection of numbers that might contain duplicates, return all possible unique permutations.

For example,
`[1,1,2]` have the following unique permutations:
`[1,1,2]`, `[1,2,1]`, and `[2,1,1]`.

» Solve this problem (link to this question)

✅ **Permutations** 6281 / 14628 Mar 17 '12

Given a collection of numbers, return all possible permutations.

For example,
`[1,2,3]` have the following permutations:
`[1,2,3]`, `[1,3,2]`, `[2,1,3]`, `[2,3,1]`, `[3,1,2]`, and `[3,2,1]`.

» Solve this problem (link to this question)

✅ **Plus One** 5323 / 12479 Apr 2 '12

Given a number represented as an array of digits, plus one to the number.

» Solve this problem (link to this question)

✅ **Populating Next Right Pointers in Each Node II** 5299 / 13177 Oct 28 '12

Follow up for problem *"Populating Next Right Pointers in Each Node"*.

What if the given tree could be any binary tree? Would your previous solution still work?

Note:

- You may only use constant extra space.

For example,
Given the following binary tree,

```
        1
      /  \
     2    3
    / \    \
   4   5    7
```

After calling your function, the tree should look like:

```
        1 -> NULL
      /  \
     2 -> 3 -> NULL
    / \    \
   4-> 5 -> 7 -> NULL
```

» Solve this problem (link to this question)

✅ **Populating Next Right Pointers in Each Node** 5938 / 12053 Oct 28 '12

Given a binary tree

```
struct TreeLinkNode {
  TreeLinkNode *left;
  TreeLinkNode *right;
  TreeLinkNode *next;
}
```

Populate each next pointer to point to its next right node. If there is no next right node, the next pointer should be set to `NULL`.

Initially, all next pointers are set to `NULL`.

Note:

- You may only use constant extra space.
- You may assume that it is a perfect binary tree (ie, all leaves are at the same level, and every parent has two children).

For example,
Given the following perfect binary tree,

```
        1
      /  \
     2    3
    / \  / \
   4  5  6  7
```

After calling your function, the tree should look like:

```
        1 -> NULL
      /  \
     2 -> 3 -> NULL
    / \  / \
   4->5->6->7 -> NULL
```

» Solve this problem (link to this question)

✅ **Pow(x, n)** 10568 / 27368 Mar 20 '12

Implement pow(x, n).

» Solve this problem (link to this question)

✅ **Recover Binary Search Tree** 4967 / 16821 Sep 1 '12

Two elements of a binary search tree (BST) are swapped by mistake.

Recover the tree without changing its structure.

Note:
A solution using $O(n)$ space is pretty straight forward. Could you devise a constant space solution?

confused what `"{1,#,2,3}"` means? > read more on how binary tree is serialized on OJ.

» Solve this problem (link to this question)

✅ **Regular Expression Matching** 7137 / 26894 Jan 8 '12

Implement regular expression matching with support for `'.'` and `'*'`.

```
'.' Matches any single character.
'*' Matches zero or more of the preceding element.

The matching should cover the entire input string (not partial).

The function prototype should be:
bool isMatch(const char *s, const char *p)

Some examples:
isMatch("aa","a") ? false
isMatch("aa","aa") ? true
isMatch("aaa","aa") ? false
isMatch("aa", "a*") ? true
isMatch("aa", ".*") ? true
isMatch("ab", ".*") ? true
isMatch("aab", "c*a*b") ? true
```

» Solve this problem (link to this question)

✅ **Remove Duplicates from Sorted Array II** 4602 / 11239 Apr 19 '12

Follow up for "Remove Duplicates":
What if duplicates are allowed at most *twice*?

For example,
Given sorted array A = `[1,1,1,2,2,3]`,

Your function should return length = `5`, and A is now `[1,1,2,2,3]`.

» Solve this problem (link to this question)

✅ **Remove Duplicates from Sorted Array** 6735 / Feb 16 '12

14262

Given a sorted array, remove the duplicates in place such that each element appear only *once* and return the new length.

Do not allocate extra space for another array, you must do this in place with constant memory.

For example,
Given input array A = `[1,1,2]`,

Your function should return length = `2`, and A is now `[1,2]`.

» Solve this problem (link to this question)

✅ **Remove Duplicates from Sorted List II**

5564 / 16823 Apr 22 '12

Given a sorted linked list, delete all nodes that have duplicate numbers, leaving only *distinct* numbers from the original list.

For example,
Given `1->2->3->3->4->4->5`, return `1->2->5`.
Given `1->1->1->2->3`, return `2->3`.

» Solve this problem (link to this question)

✅ **Remove Duplicates from Sorted List**

6081 / 11943 Apr 22 '12

Given a sorted linked list, delete all duplicates such that each element appear only *once*.

For example,
Given `1->1->2`, return `1->2`.
Given `1->1->2->3->3`, return `1->2->3`.

» Solve this problem (link to this question)

✅ **Remove Element**

6347 / 13137 Feb 16 '12

Given an array and a value, remove all instances of that value in place and return the new length.

The order of elements can be changed. It doesn't matter what you leave beyond the new length.

» Solve this problem (link to this question)

✅ **Remove Nth Node From End of List**

6630 / 17308 Jan 28 '12

Given a linked list, remove the n^{th} node from the end of list and return its head.

For example,

```
Given linked list: 1->2->3->4->5, and n = 2.

After removing the second node from the end, the linked list becomes 1->2->3->5.
```

Note:
Given *n* will always be valid.
Try to do this in one pass.

» Solve this problem (link to this question)

✅ **Restore IP Addresses**

4543 / 16957 Aug 8 '12

Given a string containing only digits, restore it by returning all possible valid IP address combinations.

For example:
Given `"25525511135"`,

return `["255.255.11.135", "255.255.111.35"]`. (Order does not matter)

» Solve this problem (link to this question)

✅ **Reverse Integer**

8021 / 14285 Dec 26 '11

Reverse digits of an integer.

Example1: x = 123, return 321
Example2: x = -123, return -321

click to show spoilers.

» Solve this problem (link to this question)

✅ **Reverse Linked List II**

5508 / Jun 27 '12

18773

Reverse a linked list from position *m* to *n*. Do it in-place and in one-pass.

For example:
Given `1->2->3->4->5->NULL`, *m* = 2 and *n* = 4,

return `1->4->3->2->5->NULL`.

Note:
Given *m*, *n* satisfy the following condition:
1 ? *m* ? *n* ? length of list.

» Solve this problem (link to this question)

✅ **Reverse Nodes in k-Group** 4578 / 15390 Feb 16 '12

Given a linked list, reverse the nodes of a linked list *k* at a time and return its modified list.

If the number of nodes is not a multiple of *k* then left-out nodes in the end should remain as it is.

You may not alter the values in the nodes, only nodes itself may be changed.

Only constant memory is allowed.

For example,
Given this linked list: `1->2->3->4->5`

For *k* = 2, you should return: `2->1->4->3->5`

For *k* = 3, you should return: `3->2->1->4->5`

» Solve this problem (link to this question)

✅ **Roman to Integer** 3996 / 8336 Jan 15 '12

Given a roman numeral, convert it to an integer.

Input is guaranteed to be within the range from 1 to 3999.

» Solve this problem (link to this question)

Rotate Image 4493 / 10153 Mar 18 '12

You are given an *n* x *n* 2D matrix representing an image.

Rotate the image by 90 degrees (clockwise).

Follow up:
Could you do this in-place?

» Solve this problem (link to this question)

✅ **Rotate List** 5071 / 17192 Mar 28 '12

Given a list, rotate the list to the right by *k* places, where *k* is non-negative.

For example:
Given `1->2->3->4->5->NULL` and *k* = `2`,
return `4->5->1->2->3->NULL`.

» Solve this problem (link to this question)

✅ **Same Tree** 6423 / 10037 Sep 3 '12

Given two binary trees, write a function to check if they are equal or not.

Two binary trees are considered equal if they are structurally identical and the nodes have the same value.

» Solve this problem (link to this question)

✅ **Scramble String** 4644 / 14931 Apr 30 '12

Given a string *s1*, we may represent it as a binary tree by partitioning it to two non-empty substrings recursively.

Below is one possible representation of *s1* = `"great"`:

```
    great
   /    \
  gr    eat
 / \    /  \
g   r  e   at
```

```
          / \
         a   t
```

To scramble the string, we may choose any non-leaf node and swap its two children.

For example, if we choose the node `"gr"` and swap its two children, it produces a scrambled string `"rgeat"`.

```
    rgeat
   /    \
  rg    eat
 / \    /  \
r   g  e   at
           / \
          a   t
```

We say that `"rgeat"` is a scrambled string of `"great"`.

Similarly, if we continue to swap the children of nodes `"eat"` and `"at"`, it produces a scrambled string `"rgtae"`.

```
    rgtae
   /    \
  rg    tae
 / \    /  \
r   g  ta  e
       / \
      t   a
```

We say that `"rgtae"` is a scrambled string of `"great"`.

Given two strings *s1* and *s2* of the same length, determine if *s2* is a scrambled string of *s1*.

» Solve this problem (link to this question)

✅ **Search a 2D Matrix** 5288 / 11782 Apr 7 '12

Write an efficient algorithm that searches for a value in an *m* x *n* matrix. This matrix has the following properties:

- Integers in each row are sorted from left to right.
- The first integer of each row is greater than the last integer of the previous row.

For example,

Consider the following matrix:

```
[
  [1,   3,  5,  7],
  [10, 11, 16, 20],
  [23, 30, 34, 50]
]
```

Given **target** = `3`, return `true`.

» Solve this problem (link to this question)

✅ **Search for a Range** 5829 / 15540 Mar 3 '12

Given a sorted array of integers, find the starting and ending position of a given target value.

Your algorithm's runtime complexity must be in the order of *O*(log *n*).

If the target is not found in the array, return `[-1, -1]`.

For example,
Given `[5, 7, 7, 8, 8, 10]` and target value 8,
return `[3, 4]`.

» Solve this problem (link to this question)

✅ **Search in Rotated Sorted Array II** 4077 / 10753 Apr 20 '12

Follow up for "Search in Rotated Sorted Array":
What if *duplicates* are allowed?

Would this affect the run-time complexity? How and why?

Write a function to determine if a given target is in the array.

» Solve this problem (link to this question)

✅ **Search in Rotated Sorted Array** 7237 / 18907 Mar 3 '12

Suppose a sorted array is rotated at some pivot unknown to you beforehand.

(i.e., `0 1 2 4 5 6 7` might become `4 5 6 7 0 1 2`).

You are given a target value to search. If found in the array return its index, otherwise return -1.

You may assume no duplicate exists in the array.

» Solve this problem (link to this question)

✅ **Search Insert Position** 5789 / 11556 Mar 3 '12

Given a sorted array and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

You may assume no duplicates in the array.

Here are few examples.
`[1,3,5,6]`, 5 → 2
`[1,3,5,6]`, 2 → 1
`[1,3,5,6]`, 7 → 4
`[1,3,5,6]`, 0 → 0

» Solve this problem (link to this question)

✅ **Set Matrix Zeroes** 4561 / 10629 Apr 6 '12

Given a *m* x *n* matrix, if an element is 0, set its entire row and column to 0. Do it in place.

Follow up:
Did you use extra space?
A straight forward solution using O(*mn*) space is probably a bad idea.
A simple improvement uses O(*m* + *n*) space, but still not the best solution.
Could you devise a constant space solution?
» Solve this problem (link to this question)

✅ **Simplify Path** 3445 / 13388 Apr 4 '12

Given an absolute path for a file (Unix-style), simplify it.

For example,
path = `"/home/"`, => `"/home"`
path = `"/a/./b/../../c/"`, => `"/c"`

click to show corner cases.

» Solve this problem (link to this question)

✅ **Sort Colors** 5950 / 14106 Apr 9 '12

Given an array with *n* objects colored red, white or blue, sort them so that objects of the same color are adjacent, with the colors in the order red, white and blue.

Here, we will use the integers 0, 1, and 2 to represent the color red, white, and blue respectively.

Note:
You are not suppose to use the library's sort function for this problem.

click to show follow up.

» Solve this problem (link to this question)

✅ **Spiral Matrix II** 3346 / 7666 Mar 28 '12

Given an integer *n*, generate a square matrix filled with elements from 1 to *n*² in spiral order.

For example,
Given *n* = `3`,

You should return the following matrix:

```
[
 [ 1, 2, 3 ],
 [ 8, 9, 4 ],
 [ 7, 6, 5 ]
```

```
]
```

» Solve this problem (link to this question)

✅ **Spiral Matrix** 4405 / 15905 Mar 25 '12

Given a matrix of *m* x *n* elements (*m* rows, *n* columns), return all elements of the matrix in spiral order.

For example,
Given the following matrix:

```
[
 [ 1, 2, 3 ],
 [ 4, 5, 6 ],
 [ 7, 8, 9 ]
]
```

You should return `[1,2,3,6,9,8,7,4,5]`.

» Solve this problem (link to this question)

✅ **Sqrt(x)** 8283 / 28329 Apr 3 '12

Implement `int sqrt(int x)`.

Compute and return the square root of *x*.

» Solve this problem (link to this question)

✅ **String to Integer (atoi)** 7626 / 35090 Dec 27 '11

Implement atoi to convert a string to an integer.

Hint: Carefully consider all possible input cases. If you want a challenge, please do not see below and ask yourself what are the possible input cases.

Notes: It is intended for this problem to be specified vaguely (ie, no given input specs). You are responsible to gather all the input requirements up front.

spoilers alert... click to show requirements for atoi.

» Solve this problem (link to this question)

✅ **Subsets II** 4769 / 13419 Jun 25 '12

Given a collection of integers that might contain duplicates, *S*, return all possible subsets.

Note:

- Elements in a subset must be in non-descending order.
- The solution set must not contain duplicate subsets.

For example,
If *S* = `[1,2,2]`, a solution is:

```
[
  [2],
  [1],
  [1,2,2],
  [2,2],
  [1,2],
  []
]
```

» Solve this problem (link to this question)

✅ **Subsets** 6226 / 16269 Apr 18 '12

Given a set of distinct integers, *S*, return all possible subsets.

Note:

- Elements in a subset must be in non-descending order.
- The solution set must not contain duplicate subsets.

For example,
If *S* = `[1,2,3]`, a solution is:

```
[
  [3],
  [1],
  [2],
  [1,2,3],
  [1,3],
  [2,3],
  [1,2],
  []
]
```

» Solve this problem (link to this question)

✔ **Substring with Concatenation of All Words** 5895 / 21700 Feb 24 '12

You are given a string, **S**, and a list of words, **L**, that are all of the same length. Find all starting indices of substring(s) in S that is a concatenation of each word in L exactly once and without any intervening characters.

For example, given:
S: `"barfoothefoobarman"`
L: `["foo", "bar"]`

You should return the indices: `[0,9]`.
(order does not matter).

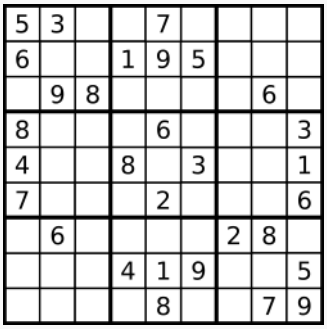
» Solve this problem (link to this question)

✔ **Sudoku Solver** 3375 / 12622 Mar 4 '12

Write a program to solve a Sudoku puzzle by filling the empty cells.

Empty cells are indicated by the character `'.'`.

You may assume that there will be only one unique solution.



A sudoku puzzle...

...and its solution numbers marked in red.

» Solve this problem (link to this question)

✔ **Sum Root to Leaf Numbers** 8344 / 23632 Feb 19

Given a binary tree containing digits from `0-9` only, each root-to-leaf path could represent a number.

An example is the root-to-leaf path `1->2->3` which represents the number `123`.

Find the total sum of all root-to-leaf numbers.

For example,

```
    1
```

```
   / \
  2   3
```

The root-to-leaf path `1->2` represents the number `12`.
The root-to-leaf path `1->3` represents the number `13`.

Return the sum = 12 + 13 = `25`.

» Solve this problem (link to this question)

✅ **Surrounded Regions** 11049 / 41380 Feb 22

Given a 2D board containing `'X'` and `'O'`, capture all regions surrounded by `'X'`.

A region is captured by flipping all `'O'`s into `'X'`s in that surrounded region .

For example,

```
X X X X
X O O X
X X O X
X O X X
```

After running your function, the board should be:

```
X X X X
X X X X
X X X X
X O X X
```

» Solve this problem (link to this question)

✅ **Swap Nodes in Pairs** 6098 / 13527 Feb 15 '12

Given a linked list, swap every two adjacent nodes and return its head.

For example,
Given `1->2->3->4`, you should return the list as `2->1->4->3`.

Your algorithm should use only constant space. You may **not** modify the values in the list, only nodes itself can be changed.

» Solve this problem (link to this question)

✅ **Symmetric Tree** 7123 / 14917 Sep 24 '12

Given a binary tree, check whether it is a mirror of itself (ie, symmetric around its center).

For example, this binary tree is symmetric:

```
    1
   / \
  2   2
 / \ / \
3  4 4  3
```

But the following is not:

```
    1
   / \
  2   2
   \   \
   3    3
```

Note:
Bonus points if you could solve it both recursively and iteratively.

confused what `"{1,#,2,3}"` means? > read more on how binary tree is serialized on OJ.

» Solve this problem (link to this question)

Text Justification 3100 / 15986 Apr 3 '12

Given an array of words and a length *L*, format the text such that each line has exactly *L* characters and is fully (left and right) justified.

You should pack your words in a greedy approach; that is, pack as many words as you can in each

line. Pad extra spaces ' ' when necessary so that each line has exactly L characters.

Extra spaces between words should be distributed as evenly as possible. If the number of spaces on a line do not divide evenly between words, the empty slots on the left will be assigned more spaces than the slots on the right.

For the last line of text, it should be left justified and no extra space is inserted between words.

For example,
words: ["This", "is", "an", "example", "of", "text", "justification."]
L: 16.

Return the formatted lines as:

```
[
   "This    is    an",
   "example  of text",
   "justification.  "
]
```

Note: Each word is guaranteed not to exceed L in length.

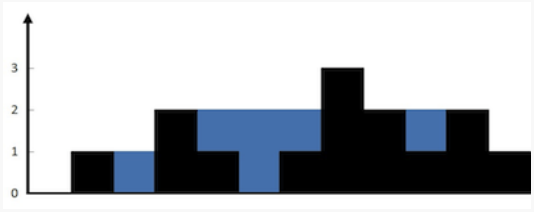
click to show corner cases.

» Solve this problem (link to this question)

✔ **Trapping Rain Water** 4508 / 11340 Mar 10 '12

Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it is able to trap after raining.

For example,
Given [0,1,0,2,1,0,1,3,2,1,2,1], return 6.



The above elevation map is represented by array [0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1]. In this case, 6 units of rain water (blue section) are being trapped. **Thanks Marcos** for contributing this image!

» Solve this problem (link to this question)

✔ **Triangle** 6503 / 17796 Oct 30 '12

Given a triangle, find the minimum path sum from top to bottom. Each step you may move to adjacent numbers on the row below.

For example, given the following triangle

```
[
     [2],
    [3,4],
   [6,5,7],
  [4,1,8,3]
]
```

The minimum path sum from top to bottom is 11 (i.e., 2 + 3 + 5 + 1 = 11).

Note:
Bonus point if you are able to do this using only O(n) extra space, where n is the total number of rows in the triangle.

» Solve this problem (link to this question)

✔ **Two Sum** 15259 / 48327 Mar 14 '11

Given an array of integers, find two numbers such that they add up to a specific target number.

The function twoSum should return indices of the two numbers such that they add up to the target, where index1 must be less than index2. Please note that your returned answers (both index1 and index2) are not zero-based.

You may assume that each input would have exactly one solution.

Input: numbers={2, 7, 11, 15}, target=9
Output: index1=1, index2=2

» Solve this problem (link to this question)

✔ Unique Binary Search Trees II

3664 / 10952 Aug 27 '12

Given *n*, generate all structurally unique **BST's** (binary search trees) that store values 1...*n*.

For example,
Given *n* = 3, your program should return all 5 unique BST's shown below.

```
   1         3     3      2      1
    \       /     /      / \      \
     3     2     1      1   3      2
    /     /       \                 \
   2     1         2                 3
```

confused what `"{1,#,2,3}"` means? > read more on how binary tree is serialized on OJ.

» Solve this problem (link to this question)

✔ Unique Binary Search Trees

5456 / 10302 Aug 27 '12

Given *n*, how many structurally unique **BST's** (binary search trees) that store values 1...*n*?

For example,
Given *n* = 3, there are a total of 5 unique BST's.

```
   1         3     3      2      1
    \       /     /      / \      \
     3     2     1      1   3      2
    /     /       \                 \
   2     1         2                 3
```

» Solve this problem (link to this question)

✔ Unique Paths II

4573 / 11497 Mar 29 '12

Follow up for "Unique Paths":

Now consider if some obstacles are added to the grids. How many unique paths would there be?

An obstacle and empty space is marked as `1` and `0` respectively in the grid.

For example,

There is one obstacle in the middle of a 3x3 grid as illustrated below.

```
[
  [0,0,0],
  [0,1,0],
  [0,0,0]
]
```

The total number of unique paths is `2`.

Note: *m* and *n* will be at most 100.

» Solve this problem (link to this question)

✔ Unique Paths

5981 / 13146 Mar 28 '12

A robot is located at the top-left corner of a *m* x *n* grid (marked 'Start' in the diagram below).

The robot can only move either down or right at any point in time. The robot is trying to reach the bottom-right corner of the grid (marked 'Finish' in the diagram below).

How many possible unique paths are there?

Above is a 3 x 7 grid. How many possible unique paths are there?

Note: *m* and *n* will be at most 100.

» Solve this problem (link to this question)

Valid Number

3716 / 20322 Apr 2 '12

Validate if a given string is numeric.

Some examples:
`"0"` => `true`
`" 0.1 "` => `true`
`"abc"` => `false`
`"1 a"` => `false`
`"2e10"` => `true`

Note: It is intended for the problem statement to be ambiguous. You should gather all requirements up front before implementing one.

» Solve this problem (link to this question)

✔ Valid Palindrome

7412 / 23481 Jan 13

Given a string, determine if it is a palindrome, considering only alphanumeric characters and ignoring cases.

For example,
`"A man, a plan, a canal: Panama"` is a palindrome.
`"race a car"` is *not* a palindrome.

Note:
Have you consider that the string might be empty? This is a good question to ask during an interview.

For the purpose of this problem, we define empty string as valid palindrome.

» Solve this problem (link to this question)

✔ Valid Parentheses

5976 / 15549 Jan 30 '12

Given a string containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string is valid.

The brackets must close in the correct order, `"()"` and `"()[]{}"` are all valid but `"(]"` and `"([)]"` are not.

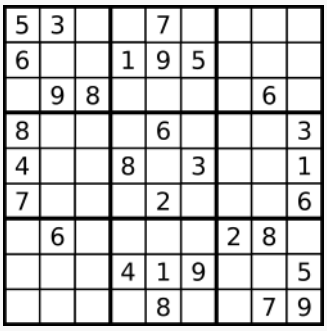
» Solve this problem (link to this question)

✔ Valid Sudoku

3545 / 10545 Mar 3 '12

Determine if a Sudoku is valid, according to: Sudoku Puzzles - The Rules.

The Sudoku board could be partially filled, where empty cells are filled with the character `'.'`.



A partially filled sudoku which is valid.

» Solve this problem (link to this question)

✔ Validate Binary Search Tree

6761 / 19400 Aug 31 '12

Given a binary tree, determine if it is a valid binary search tree (BST).

Assume a BST is defined as follows:

- The left subtree of a node contains only nodes with keys **less than** the node's key.
- The right subtree of a node contains only nodes with keys **greater than** the node's key.
- Both the left and right subtrees must also be binary search trees.

OJ's Binary Tree Serialization:
The serialization of a binary tree follows a level order traversal, where '#' signifies a path terminator where no node exists below.

Here's an example:

```
   1
  / \
 2   3
    /
   4
    \
     5
```

The above binary tree is serialized as `"{1,2,3,#,#,4,#,#,5}"`.

» Solve this problem (link to this question)

✅ **Wildcard Matching** 8261 / 32715 Mar 16 '12

Implement wildcard pattern matching with support for `'?'` and `'*'`.

```
'?' Matches any single character.
'*' Matches any sequence of characters (including the empty sequence).

The matching should cover the entire input string (not partial).

The function prototype should be:
bool isMatch(const char *s, const char *p)

Some examples:
isMatch("aa","a") ? false
isMatch("aa","aa") ? true
isMatch("aaa","aa") ? false
isMatch("aa", "*") ? true
isMatch("aa", "a*") ? true
isMatch("ab", "?*") ? true
isMatch("aab", "c*a*b") ? false
```

» Solve this problem (link to this question)

✅ **Word Ladder II** 7646 / 36433 Feb 11

Given two words (*start* and *end*), and a dictionary, find all shortest transformation sequence(s) from *start* to *end*, such that:

- Only one letter can be changed at a time
- Each intermediate word must exist in the dictionary

For example,

Given:
start = `"hit"`
end = `"cog"`
dict = `["hot","dot","dog","lot","log"]`

Return

```
[
  ["hit","hot","dot","dog","cog"],
  ["hit","hot","lot","log","cog"]
]
```

Note:

- All words have the same length.
- All words contain only lowercase alphabetic characters.

» Solve this problem (link to this question)

✅ **Word Ladder** 10499 / 37276 Feb 11

Given two words (*start* and *end*), and a dictionary, find the length of shortest transformation sequence from *start* to *end*, such that:

- Only one letter can be changed at a time
- Each intermediate word must exist in the dictionary

For example,

Given:
start = `"hit"`
end = `"cog"`
dict = `["hot","dot","dog","lot","log"]`

As one shortest transformation is `"hit" -> "hot" -> "dot" -> "dog" -> "cog"`,
return its length `5`.

Note:

- Return 0 if there is no such transformation sequence.
- All words have the same length.
- All words contain only lowercase alphabetic characters.

» Solve this problem (link to this question)

✔ **Word Search** 6762 / 21892 Apr 18 '12

Given a 2D board and a word, find if the word exists in the grid.

The word can be constructed from letters of sequentially adjacent cell, where "adjacent" cells are those horizontally or vertically neighboring. The same letter cell may not be used more than once.

For example,
Given **board** =

```
[
  ["ABCE"],
  ["SFCS"],
  ["ADEE"]
]
```

word = `"ABCCED"`, -> returns `true`,
word = `"SEE"`, -> returns `true`,
word = `"ABCB"`, -> returns `false`.

» Solve this problem (link to this question)

ZigZag Conversion 6562 / 18872 Dec 6 '11

The string `"PAYPALISHIRING"` is written in a zigzag pattern on a given number of rows like this: (you may want to display this pattern in a fixed font for better legibility)

```
P   A   H   N
A P L S I I G
Y   I   R
```

And then read line by line: `"PAHNAPLSIIGYIR"`

Write the code that will take a string and make this conversion given a number of rows:

```
string convert(string text, int nRows);
```

`convert("PAYPALISHIRING", 3)` should return `"PAHNAPLSIIGYIR"`.

» Solve this problem (link to this question)