DYNAMIC PROGRAMMING

Roadmap

Curated by tutort academy

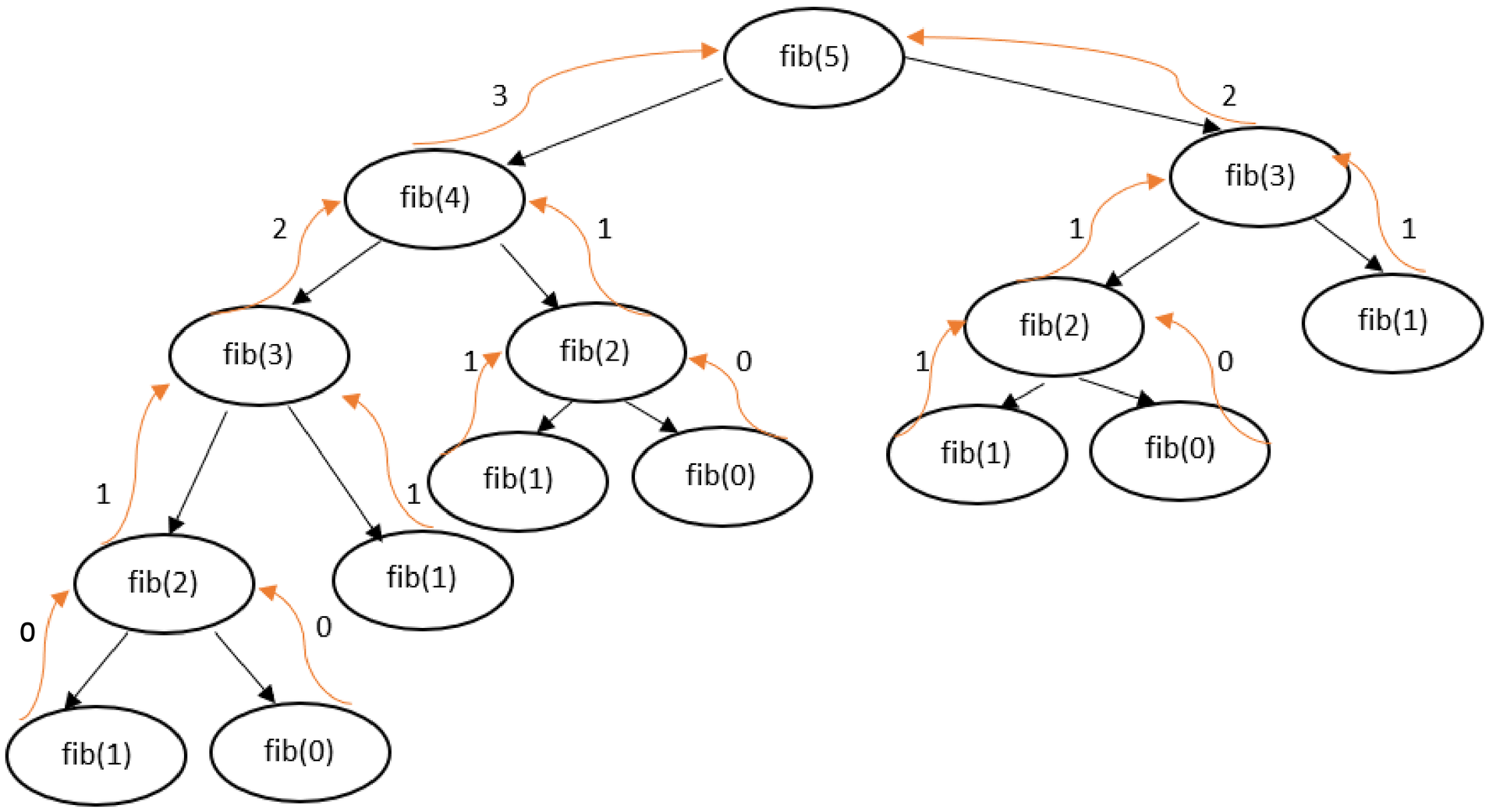
What is Dynamic programming?

Dynamic programming is an optimization technic used on top of recursion so that if the value is already calculated it is not recalculated if it is required again. It is done by storing the calculated value in a memory. We use lists/ array, maps, etc. to store the value when it is calculated for the first time.

Let's see this in example:

```
int fib( int num) {
if ( num <= 1)
return num;
return fib(num-1) + fib(num -2)
}
```

Recursive code of fibonacci number



Recursion tree stack of recursive code

As see can see that in the recursion tree stack of the recursion code that for fib(5), the code is calculating fib(3) two times, fib(2) three times and so on. So suppose for calculating fib(1000) there will be thousand of recalculations. To avoid this we can store the values in memory when the intermediate value is calculated (fib(3), fib(2), etc. in this case) for the first time. This is called memorization.

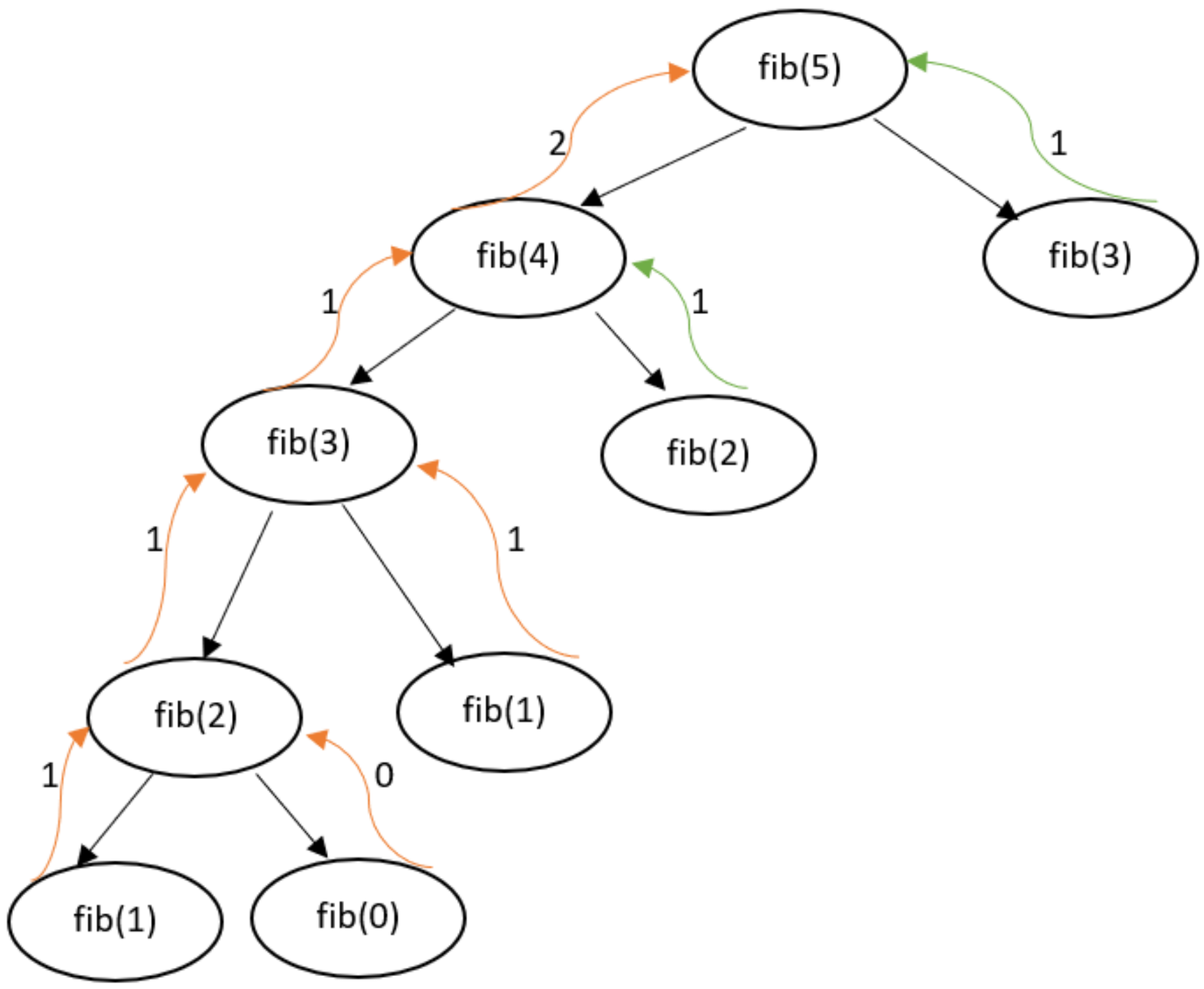
Time Complexity: $O(2^n)$

Space Complexity: O(n)

Now let's see the dynamic programming approach:

```
int fib( int num){
if( fib <= 1)
return num;
if( dp[num] != -1)
return dp[num];
return dp[num]= fib(num – 1) +fib( num -2 );
}
```

Memoized code of Fibonacci number



Recursion tree stack of memorized code

As we can see here that with dynamic programming we have cut down on the unnecessary tree calls by storing the calculated value in an array named dp. So, fib(3), fib(2) isn't recalculated. To check if the value of a certain number(num) is calculated or not we initialize dp (array) with a dummy data which cannot be the intermediate value (-1) in this case. So in every recursion call we check whether the number is calculated or not by checking dp[n] != -1 and if it satisfies the condition, then we return the value dp[n] from the dp array without calculating the intermediate value again.

Time Complexity: O(n)

Space Complexity: O(n)

In the above figure , the green links are returned from dp array as it is calculated before for that value.

There are two approaches to dynamic programming:

- **Top-down(Memoization)**: The top-down approach follows the memorization technique, while bottom-up approach follows the tabulation method. Here memorization is equal to the sum of recursion and caching. Recursion means calling the function itself, while caching means storing the intermediate results.
  It is preferred because It is very easy to understand , implement and debug. It also solves the subproblems only when it is required.
  Its only disadvantage is that it uses the recursion technique that occupies more memory in the call stack. Sometimes when the recursion is too deep, the stack overflow condition will occur.

- **Bottom-up (Tabulation)**: The bottom-up approach is also one of the techniques which can be used to implement the dynamic programming. It uses the tabulation technique to implement the dynamic programming approach. It solves the same kind of problems but it removes the recursion. If we remove the recursion, there is no stack overflow issue and no overhead of the recursive functions. In this tabulation technique, we solve the problems and store the results in a matrix.
  Its advantage is that this is used to avoid recursion , thus saving memory space while its little.

In the above example we have used top-down(memoization) approach for dynamic programming.

Note: It is highly recommended to have a good grip on recursion before starting dynamic programming.

Steps to memoize a recursive solution:

Any recursive solution to a problem can be memoized using these three steps:

- Create a dp[n+1] array initialized to -1.
- Whenever we want to find the answer of a particular value (say n), we first check whether the answer is already calculated using the dp array(i.e dp[n]!= -1). If yes, simply return the value from the dp array.
- If not, then we are finding the answer for the given value for the first time, we will use the recursive relation as usual but before returning from the function, we will set dp[n] to the solution we get.

Problems:

1 0-1 knapsack:

You are given weights and values of N items, put these items in a knapsack of capacity W to get the maximum total value in the knapsack. Note that we have only one quantity of each item.
In other words, given two integer arrays val[0..N-1] and wt[0..N-1] which represent values and weights associated with N items respectively. Also given an integer W which represents knapsack capacity, find out the maximum value subset of val[] such that sum of the weights of this subset is smaller than or equal to W. You cannot break an item, either pick the complete item or dont pick it (0-1 property).

Practice

Question Asked in:

2 Subset sum problem:

Given an array of non-negative integers, and a value sum, determine if there is a subset of the given set with sum equal to given sum.

Practice

Question Asked in:

3 Equal Sum Partition problem

Given an array arr[] of size N, check if it can be partitioned into two parts such that the sum of elements in both parts is the same.

Practice

Question Asked in:

4 Subset sum problem:

You are given an integer array nums and an integer target.

You want to build an expression out of nums by adding one of the symbols '+' and '-' before each integer in nums and then concatenate all the integers.

· For example, if nums = [2, 1], you can add a '+' before 2 and a '-' before 1 and concatenate them to build the expression "+2-1".

Return the number of different expressions that you can build, which evaluates to target.

Practice

Question Asked in:

5 Unbounded Knapsack:

Given a set of N items, each with a weight and a value, represented by the array w[] and val[] respectively. Also, a knapsack with weight limit W.
The task is to fill the knapsack in such a way that we can get the maximum profit. Return the maximum profit.

Note: Each item can be taken any number of times.

Practice

Question Asked in:

6 Coin change

You are given an integer array coins representing coins of different denominations and an integer amount representing a total amount of money.

Return the fewest number of coins that you need to make up that amount. If that amount of money cannot be made up by any combination of the coins, return -1.

You may assume that you have an infinite number of each kind of coin.

Practice

Question Asked in:

7 Rod Cutting

Given a rod of length N inches and an array of prices, price[ ]. price[i] denotes the value of a piece of length i. Determine the maximum value obtainable by cutting up the rod and selling the pieces.

Practice

Question Asked in:

8 Minimum Cost to Cut a Stick

Given a wooden stick of length n units. The stick is labelled from 0 to n. For example, a stick of length 6 is labelled as follows:

Given an integer array cuts where cuts[i] denotes a position you should perform a cut at. You should perform the cuts in order, you can change the order of the cuts as you wish.The cost of one cut is the length of the stick to be cut, the total cost is the sum of costs of all cuts. When you cut a stick, it will be split into two smaller sticks (i.e. the sum of their lengths is the length of the stick before the cut). Please refer to the first example for a better explanation.
Return the minimum total cost of the cuts.

Practice

Question Asked in:

9 Minimum cost for tickets

You have planned some train traveling one year in advance. The days of the year in which you will travel are given as an integer array days. Each day is an integer from 1 to 365.

Train tickets are sold in three different ways:
· a 1-day pass is sold for costs[0] dollars,
· a 7-day pass is sold for costs[1] dollars, and
· a 30-day pass is sold for costs[2] dollars.
The passes allow that many days of consecutive travel.

· For example, if we get a 7-day pass on day 2, then we can travel for 7 days: 2, 3, 4, 5, 6, 7, and 8.

Return the minimum number of dollars you need to travel every day in the given list of days.

Practice

Asked in:

10 Longest Common Subsequence

Given two strings, find the length of longest subsequence present in both of them. Both the strings are in uppercase latin alphabets.

Practice

Question Asked in:

11 Maximum Length of Repeated Subarray

Given two integer arrays nums1 and nums2, return the maximum length of a subarray that appears in both arrays.

Practice

Question Asked in:

12 Shortest Common Supersequence

Given two strings str1 and str2, return the shortest string that has both str1 and str2 as subsequences. If there are multiple valid strings, return any of them.

A string s is a subsequence of string t if deleting some number of characters from t (possibly 0) results in the string s.

Practice

Question Asked in:

13 Longest Palindromic Subsequence

Given a string s, find the longest palindromic subsequence's length in s.

A subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements.

Practice

Question Asked in:

14 Longest Repeating Subsequence

Given string str, find the length of the longest repeating subsequence such that it can be found twice in the given string.

The two identified subsequences A and B can use the same ith character from string str if and only if that ith character has different indices in A and B. For example, A = "xax" and B = "xax" then the index of first "x" must be different in the original string for A and B.

Practice

Question Asked in:

15 Print all LCS sequences

You are given two strings s and t. Now your task is to print all longest common sub-sequences in lexicographical order.

Practice

Question Asked in:

16 Delete Operation for Two Strings

Given two strings word1 and word2, return the minimum number of steps required to make word1 and word2 the same.

In one step, you can delete exactly one character in either string.

Practice

Question Asked in:

17 Edit Distance

Given two strings word1 and word2, return the minimum number of operations required to convert word1 to word2.
You have the following three operations permitted on a word:

- Insert a character
- Delete a character
- Replace a character

Practice

Question Asked in: Goldman Sachs

18 Unique Paths

There is a robot on an m x n grid. The robot is initially located at the top-left corner (i.e., grid[0][0]). The robot tries to move to the bottom-right corner (i.e., grid[m - 1][n - 1]). The robot can only move either down or right at any point in time.

Given the two integers m and n, return the number of possible unique paths that the robot can take to reach the bottom-right corner.

Practice

Question Asked in:

19 Unique Paths II

You are given an m x n integer array grid. There is a robot initially located at the top-left corner (i.e., grid[0][0]). The robot tries to move to the bottom-right corner (i.e., grid[m - 1][n - 1]). The robot can only move either down or right at any point in time.
An obstacle and space are marked as 1 or 0 respectively in grid. A path that the robot takes cannot include any square that is an obstacle.
Return the number of possible unique paths that the robot can take to reach the bottom-right corner.

Practice

Question Asked in:

20 Minimum Path Sum

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.

Practice

Question Asked in: Walmart

21 Rat in a Maze Problem

Consider a rat placed at (0, 0) in a square matrix of order N * N. It has to reach the destination at (N - 1, N - 1). Find all possible paths that the rat can take to reach from source to destination. The directions in which the rat can move are 'U'(up), 'D'(down), 'L' (left), 'R' (right). Value 0 at a cell in the matrix represents that it is blocked and rat cannot move to it while value 1 at a cell in the matrix represents that rat can be travel through it.

Note: In a path, no cell can be visited more than one time. If the source cell is 0, the rat cannot move to any other cell.

Practice

Question Asked in:

22 N-Queens

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

Given an integer n, return all distinct solutions to the n-queens puzzle. You may return the answer in any order.

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

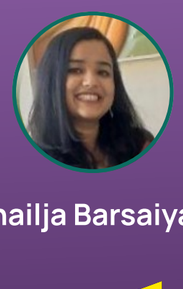
Practice

Asked in:

Gopal Yadav

zest

I took the Advanced DSA Course at Tutort Academy. Nishant Sir's explanation of the concepts was excellent. I thoroughly enjoyed the course. The course is also valid for a lifetime, and new material is added regularly. With their help, I cracked many top product based companies & currently working with Zest Money.



Shailja Barsaiyan

EY
Building a better working world

I recently finished the Data Science & ML course from Tutort Academy and got placed at EY. It's an excellent choice for learning data science. The best part was that the course teaches everything from the beginning to advanced level. Their resume-building session and mock interviews were a great help to crack the interviews. Thank you Tutort Academy for helping me achieve my goals!!

23 Dungeon Game

The demons had captured the princess and imprisoned her in the bottom-right corner of a dungeon. The dungeon consists of m x n rooms laid out in a 2D grid. Our valiant knight was initially positioned in the top-left room and must fight his way through dungeon to rescue the princess.

The knight has an initial health point represented by a positive integer. If at any point his health point drops to 0 or below, he dies immediately.

Some of the rooms are guarded by demons (represented by negative integers), so the knight loses health upon entering these rooms; other rooms are either empty (represented as 0) or contain magic orbs that increase the knight's health (represented by positive integers).

To reach the princess as quickly as possible, the knight decides to move only rightward or downward in each step. Return the knight's minimum initial health so that he can rescue the princess.

Note that any room can contain threats or power-ups, even the first room the knight enters and the bottom-right room where the princess is imprisoned.

Practice

Question Asked in:

24 Matrix Chain Multiplication

Given a sequence of matrices, find the most efficient way to multiply these matrices together. The efficient way is the one that involves the least number of multiplications.

The dimensions of the matrices are given in an array arr[] of size N (such that N = number of matrices + 1) where the ith matrix has the dimensions (arr[i-1] x arr[i]).

Practice

Question Asked in:

Tutort Benefits

1:1 Mentorship from Industry experts

24x7 Live 1:1 Video based doubt support

Special support for foreign students

Resume building & Mock Interview Preparations

25 Egg Dropping Problem

You are given N identical eggs and you have access to a K-floored building from 1 to K.

There exists a floor f where 0 <= f <= K such that any egg dropped from a floor higher than f will break, and any egg dropped from or below floor f will not break.

There are few rules given below.

· An egg that survives a fall can be used again.
· A broken egg must be discarded.
· The effect of a fall is the same for all eggs.
· If the egg doesn't break at a certain floor, it will not break at any floor below.
· If the eggs breaks at a certain floor, it will break at any floor above.

Return the minimum number of moves that you need to determine with certainty what the value of f is.

Practice

Asked in:

26 Palindrome Partitioning

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

Return all possible palindrome partitioning of s.

Practice

Question Asked in:

27 Word Break

Given a string s and a dictionary of strings wordDict, return true if s can be segmented into a space-separated sequence of one or more dictionary words.

Note that the same word in the dictionary may be reused multiple times in the segmentation.

Practice

Asked in:

28 Scramble String

We can scramble a string s to get a string t using the following algorithm:

If the length of the string is 1, stop.
If the length of the string is > 1, do the following:

- Split the string into two non-empty substrings at a random index, i.e., if the string is s, divide it to x and y where s = x + y.
- Randomly decide to swap the two substrings or to keep them in the same order. i.e., after this step, s may become s = x + y or s = y + x.
- Apply step 1 recursively on each of the two substrings x and y.

Given two strings s1 and s2 of the same length, return true if s2 is a scrambled string of s1, otherwise, return false.

Practice

Question Asked in:

29 Box Stacking

You are given a set of N types of rectangular 3-D boxes, where the ith box has height h, width w and length l. Your task is to create a stack of boxes which is as tall as possible, but you can only stack a box on top of another box if the dimensions of the 2-D base of the lower box are each strictly larger than those of the 2-D base of the higher box. Of course, you can rotate a box so that any side functions as its base.It is also allowable to use multiple instances of the same type of box. Your task is to complete the function maxHeight which returns the height of the highest possible stack so formed.

Practice

Question Asked in: Code Nation

30 Wildcard Matching

Given an input string (s) and a pattern (p), implement wildcard pattern matching with support for '?' and '*' where:

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

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

Practice

Question Asked in: Walmart

tutort

Start Your Upskilling with us

Join our Job-Oriented Programs

Explore More

www.tutort.net

Watch us on Youtube

Read more on Quora

Explore our courses

Advanced DSA & System Design Course

Full Stack Specialisation in Software Development

Follow us on

LinkedIn Instagram YouTube

Phone
+91-8712338901

E-mail
contact@tutort.net