GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

- Home
- Algorithms
- DS
- GATE
- Interview Corner
- Q&A
- C
- C++
- Java
- Books
- Contribute
- Ask a Q
- About

Array
Bit Magic
C/C++
Articles
GFacts
Linked List
MCQ
Misc
Output
String
Tree
Graph

Print all possible strings that can be made by placing spaces

Given a string you need to print all possible strings that can be made by placing spaces (zero or one) in between them.

```
Input:  str[] = "ABC"
Output: ABC
        AB C
        A BC
        A B C
```

Source: Amazon Interview Experience | Set 158, Round 1 ,Q 1.

We strongly recommend to minimize your browser and try this yourself first.

The idea is to use recursion and create a buffer that one by one contains all output strings having spaces. We keep updating buffer in every recursive call. If the length of given string is ‘n’ our updated string can have maximum length of n + (n-1) i.e. 2n-1. So we create buffer size of 2n (one extra character for string termination).
We leave 1st character as it is, starting from the 2nd character, we can either fill a space or a character. Thus one can write a recursive function like below.

```
// C++ program to print permutations of a given string with spaces.
#include <iostream>
#include <cstring>
using namespace std;

/* Function recursively prints the strings having space pattern.
   i and j are indices in 'str[]' and 'buff[]' respectively */
void printPatternUtil(char str[], char buff[], int i, int j, int n)
{
    if (i==n)
    {
        buff[j] = '\0';
        cout << buff << endl;
        return;
    }

    // Either put the character
    buff[j] = str[i];
    printPatternUtil(str, buff, i+1, j+1, n);

    // Or put a space followed by next character
    buff[j] = ' ';
    buff[j+1] = str[i];

    printPatternUtil(str, buff, i+1, j+2, n);
}

// This function creates buf[] to store individual output string and uses
// printPatternUtil() to print all permutations.
void printPattern(char *str)
{
    int n = strlen(str);

    // Buffer to hold the string containing spaces
    char buf[2*n]; // 2n-1 characters and 1 string terminator

    // Copy the first character as it is, since it will be always
    // at first position
    buf[0] = str[0];

    printPatternUtil(str, buf, 1, 1, n);
}

// Driver program to test above functions
int main()
{
    char *str = "ABCDE";
    printPattern(str);
    return 0;
}
```

Output:

```
ABCD
ABC D
AB CD
AB C D
A BCD
A BC D
A B CD
A B C D
```

Time Complexity: Since number of Gaps are n-1, there are total 2^(n-1) patters each having length ranging from n to 2n-1. Thus overall complexity would be O(n*(2^n)).

This article is contributed by **Gaurav Sharma**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Related Topics:

- Recursively print all sentences that can be formed from list of word lists
- Check if a given sequence of moves for a robot is circular or not
- Find the longest substring with k unique characters in a given string
- Function to find Number of customers who could not get a computer
- Find maximum depth of nested parenthesis in a string
- Find all distinct palindromic sub-strings of a given string
- Find if a given string can be represented from a substring by iterating the substring “n” times
- Suffix Tree Application 6 – Longest Palindromic Substring

http://ideone.com/U5Cill

Reply • Share ›

rohit • 3 months ago

package com.rkjalan.study.ds.String;

public class StringWithSpace {

public static void main(String[] args) {

printComb("abc");

}

private static void printComb(String s) {

if(s==null || s.isEmpty())

return;

printComb("a",s,1);

}

private static void printComb(String p, String s,int i) {

see more

Reply • Share ›

Ekta Goel • 4 months ago

Very similar recursive implementation: http://ideone.com/9vB2HR

Reply • Share ›

Kenneth • 4 months ago

DP solution:
http://ideone.com/8l2RUy

Reply • Share ›

aviator31 → Kenneth • 4 months ago

How can it be a DP solution when u have to print all the strings anyway ??
exhaustive printing has to be done right ??

2 • Reply • Share ›

Dharmvir Singh • 4 months ago

Isn't this is a subset of finding the possible permutations of string except that all starting from first letter.

Reply • Share ›

Fisnik • 4 months ago

Here is another solution (C++), based on bitsets, does not require any recursion:
http://ideone.com/8FSWvq

Reply • Share ›

Manoj Patial • 4 months ago

java implementation

```
public class Stringwithsapce {

public static void printStrings(String s, int i, String temp, int n)

{

temp = temp + s.charAt(i);

if( i== n-1)

{

System.out.println(temp);

return;

}

//temp = temp + s.charAt(i);
```

see more

1 Reply • Share ›

Srini • 4 months ago

Suppose there are n characters, then there are n-1 positions where a space can be inserted. Each position can either take a 0 (no space) or 1 . So if we just iterate from 0 to n-1 in binary form, we cover all such use cases.

Eg for ABC, we have 00,01,10,11 correspond to ABC, AB C, A BC, A B C

6 Reply • Share ›

peng li ➚ Srini • 3 months ago

Good method. Will this lead space complexity to o(n)?

Reply • Share ›

Gaurav Sharma ➚ Srini • 4 months ago

yes, you can do it like that also :)

Reply • Share ›

d_k ➚ Gaurav Sharma • 4 months ago

http://www.geeksforgeeks.org/f...

plz help me for solving this problem

^ | v • Reply • Share ›

geekyprateek ➚ Srini • 4 months ago

http://ideone.com/kRAhj5

Programme based on your idea!!

^ | v • Reply • Share ›

Srini ➚ Srini • 4 months ago

Correction .. 0 to 2^(n-1)

^ | v • Reply • Share ›

- Interview Experiences
- Advanced Data Structures
- Dynamic Programming
- Greedy Algorithms
- Backtracking
- Pattern Searching
- Divide & Conquer
- Mathematical Algorithms
- Recursion
- Geometric Algorithms

Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C

- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

- Follow @GeeksforGeeks

Recent Comments

- Ashish Aggarwal

  Try Data Structures and Algorithms Made Easy -...

  Algorithms · 17 minutes ago

- Vlad

  Thanks. Very interesting lectures.

  Expected Number of Trials until Success · 1 hour ago

- cfh

  My implementation which prints the index of the...

  Longest Even Length Substring such that Sum of First and Second Half is same · 1 hour ago

- Gaurav pruthi

  forgot to see that part ;)

  Bloomberg Interview | Set 1 (Phone Interview) · 1 hour ago

- saeid aslami

  thanks

  Greedy Algorithms | Set 7 (Dijkstra's shortest path algorithm) · 1 hour ago

- Cracker

  Implementation:...

  Implement Stack using Queues · 2 hours ago