Write a program to print all permutations of a given string

Difficulty Level : Medium • Last Updated : 04 Jun, 2021

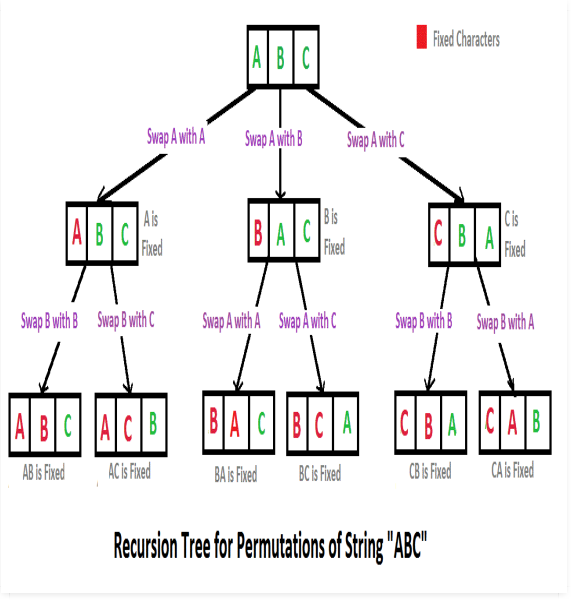
A permutation, also called an "arrangement number" or "order," is a rearrangement of the elements of an ordered list S into a one-to-one correspondence with S itself. A string of length n has n! permutation.

Source:
Mathword(http://mathworld.wolfram.com/Permutation.html)

Below are the permutations of string ABC.
ABC ACB BAC BCA CBA CAB

Recommended: Please solve it on "***PRACTICE***" first, before moving

Here is a solution that is used as a basis in backtracking.



C++

```
// C++ program to print al
// permutations with duplic
#include <bits/stdc++.h>
using namespace std;


// Function to print permut
// This function takes thre
// 1. String
// 2. Starting index of the
// 3. Ending index of the s
void permute(string a, int
{
    // Base case
    if (l == r)
        cout<<a<<endl;
    else
    {
        // Permutations mac
        for (int i = l; i
        {

            // Swapping dor
            swap(a[l], a[i

            // Recursion ca
            permute(a, l+1,

            //backtrack
            swap(a[l], a[i
        }
    }
}

// Driver Code
int main()
{
    string str = "ABC";
    int n = str.size();
    permute(str, 0, n-1);
    return 0;
}

// This is code is contribu
```

C

```c
// C program to print all p
#include <stdio.h>
#include <string.h>

/* Function to swap values
void swap(char *x, char *y
{
    char temp;
    temp = *x;
    *x = *y;
    *y = temp;
}

/* Function to print permut
This function takes three p
1. String
2. Starting index of the st
3. Ending index of the str
void permute(char *a, int
{
int i;
if (l == r)
    printf("%s\n", a);
else
{
    for (i = l; i <= r; i+
    {
        swap((a+l), (a+i))
        permute(a, l+1, r)
        swap((a+l), (a+i))
    }
}
}

/* Driver program to test a
int main()
{
    char str[] = "ABC";
    int n = strlen(str);
    permute(str, 0, n-1);
    return 0;
}
```

Java

```java
// Java program to print al
// given string.
public class Permutation
```

```
{
    public static void mai
    {
        String str = "ABC",
        int n = str.length
        Permutation permuta
        permutation.permute
    }

    /**
    * permutation function
    * @param str string to
    * @param l starting inc
    * @param r end index
    */
    private void permute(S
    {
        if (l == r)
            System.out.prir
        else
        {
            for (int i = l
            {
                str = swap
                permute(str
                str = swap
            }
        }
    }

    /**
    * Swap Characters at po
    * @param a string value
    * @param i position 1
    * @param j position 2
    * @return swapped strir
    */
    public String swap(Str
    {
        char temp;
        char[] charArray =
        temp = charArray[i]
        charArray[i] = char
        charArray[j] = temp
        return String.value
    }

}

// This code is contributed
```

Python

```
# Python program to print a
# duplicates allowed

def toString(List):
    return ''.join(List)

# Function to print permuta
# This function takes three
# 1. String
# 2. Starting index of the
# 3. Ending index of the st
def permute(a, l, r):
    if l==r:
        print toString(a)
    else:
        for i in xrange(l,
            a[l], a[i] = a
            permute(a, l+1,
            a[l], a[i] = a

# Driver program to test th
string = "ABC"
n = len(string)
a = list(string)
permute(a, 0, n-1)

# This code is contributed
```

C#

```
// C# program to print all
// permutations of a given
using System;

class GFG
{
    /**
    * permutation function
    * @param str string to
    calculate permutation
    * @param l starting inc
    * @param r end index
    */
    private static void pe

    {
```

```
            if (l == r)
                Console.WriteL:
            else
            {
                for (int i = l
                {
                    str = swap
                    permute(st:
                    str = swap
                }
            }
        }

        /**
        * Swap Characters at p
        * @param a string valu
        * @param i position 1
        * @param j position 2
        * @return swapped stri
        */
        public static String s

        {
            char temp;
            char[] charArray =
            temp = charArray[i
            charArray[i] = cha
            charArray[j] = tem
            string s = new str
            return s;
        }

    // Driver Code
    public static void Main()
    {
        String str = "ABC";
        int n = str.Length;
        permute(str, 0, n-1);
    }
    }

    // This code is contribute
```

PHP

```
<?php
// PHP program to print al
// permutations of a given
```

```
/**
* permutation function
* @param str string to
* calculate permutation fo
* @param l starting index
* @param r end index
*/
function permute($str, $l,
{
    if ($l == $r)
        echo $str. "\n";
    else
    {
        for ($i = $l; $i <
        {
            $str = swap($s
            permute($str, 
            $str = swap($s
        }
    }
}

/**
* Swap Characters at posit
* @param a string value
* @param i position 1
* @param j position 2
* @return swapped string
*/
function swap($a, $i, $j)
{
    $temp;
    $charArray = str_split
    $temp = $charArray[$i]
    $charArray[$i] = $char
    $charArray[$j] = $temp
    return implode($charAr
}

// Driver Code
$str = "ABC";
$n = strlen($str);
permute($str, 0, $n - 1);

// This code is contribute
?>
```

Output:

```
ABC
ACB
BAC
BCA
CBA
CAB
```

Algorithm Paradigm: Backtracking

Time Complexity: O(n*n!) Note that there are n! permutations and it requires O(n) time to print a permutation.

Note : The above solution prints duplicate permutations if there are repeating characters in input string. Please see below link for a solution that prints only distinct permutations even if there are duplicates in input.
Print all distinct permutations of a given string with duplicates.
Permutations of a given string using STL

Another approach:

C++

```
#include <bits/stdc++.h>
#include <string>
using namespace std;

void permute(string s , st
{
    if(s.length() == 0)
    {
        cout<<answer<<"  ",
        return;
    }
    for(int i=0 ; i<s.leng
    {
        char ch = s[i];
        string left_substr
        string right_subst
        string rest = left_
        permute(rest , ans
    }

}

int main()
{
    string s;
    string answer="";

    cout<<"Enter the strin
    cin>>s;

    cout<<"\nAll possible
    permute(s , answer);
    return 0;
}
```

Java

```
import java.util.*;

class GFG{

static void permute(String
{
    if (s.length() == 0)
    {
        System.out.print(a
        return;
    }

    for(int i = 0 ;i < s.l
    {
        char ch = s.charAt
        String left_substr
        String right_subst
        String rest = left_
        permute(rest, answ
    }
}

// Driver code
public static void main(St
{
    Scanner scan = new Sca

    String s;
    String answer="";

    System.out.print("Ente
    s = scan.next();

    System.out.print("\nAl
    permute(s, answer);
}
}

// This code is contribute
```

Python3

```
def permute(s, answer):
    if (len(s) == 0):
        print(answer, end =
        return

    for i in range(len(s))
        ch = s[i]
        left_substr = s[0:
        right_substr = s[i
        rest = left_substr
        permute(rest, answe

# Driver Code
answer = ""

s = input("Enter the strin

print("All possible string
permute(s, answer)

# This code is contributed
```

```
Output:
Enter the string : abc
All possible strings are :
```

Time Complexity: O(n*n!) The time complexity is the same as the above approach, i.e. there are n! permutations and it requires O(n) time to print a permutation.

Please write comments if you find the above codes/algorithms incorrect, or find other ways to solve the same problem.

Previous

Print reverse of a string using recursion

Next

Print all distinct permutations of a given string with duplicates

01 Generate all binary permutations such that there are more or equal 1's than 0's before every point in all permutations
17, Aug 14

02 Print all distinct permutations of a given string with duplicates
06, Jun 15

03 Print all the palindromic permutations of given string in alphabetic order
01, Jan 18

04 Print all lexicographical greater permutations of a given string
22, Jun 20

05 Print all palindrome permutation of a string
02, Mar 16

06 Print all permutation of a string in Java
16, Jan 19

07 Print all the permutation of a string without repetition using Collections i Java
03, Sep 19

08 Java Program to print distinct permutation of a string
05, Feb 19

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Medium

Easy Normal Medium Hard Expert

Improved By : Mithun Kumar, ArindamDuttagupta, rathbhupendra, nikhilchhipa9, imsushant12, srivastavaharshit848, adityapande88

Article Tags : Accolite, Amazon, Apple, Cisco, Citrix, MAQ Software, OYO Rooms, permutation, Samsung, Snapdeal, Walmart, Backtracking, Combinatorial, Greedy, Mathematical, Recursion, Strings

Practice Tags : Accolite, Amazon, OYO Rooms, Samsung, Snapdeal, Citrix, Walmart, MAQ Software, Cisco, Apple, Strings, Greedy, Mathematical, Recursion, permutation, Combinatorial, Backtracking

Improve Article

Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

GeeksforGeeks

5th Floor, A-118, Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org