

**CATEGORY ARCHIVES: STRINGS**

Questions related to strings

**Longest Palindromic Substring | Set 2**

October 17, 2012

Given a string, find the longest substring which is palindrome. For example, if the given string is "forgeeksskeegfor", the output should be "geeksskeeg".

[50 Comments](#) Category: [Strings](#)**An in-place algorithm for String Transformation**

October 15, 2012

Given a string, move all even positioned elements to end of string. While moving elements, keep the relative order of all even positioned and odd positioned elements same.

[95 Comments](#) Category: [Strings](#)**Longest Palindromic Substring | Set 1**

October 14, 2012

Given a string, find the longest substring which is palindrome. For example, if the given string is "forgeeksskeegfor", the output should be "geeksskeeg".

[86 Comments](#) Category: [Strings](#) Tags: [Dynamic Programming](#)**Print all permutations in sorted (lexicographic) order**

October 7, 2012

Given a string, print all permutations of it in sorted order. For example, if the input string is "ABC", then output should be "ABC, ACB, BAC, BCA, CAB, CBA".

[74 Comments](#) Category: [Strings](#) Tags: [MathematicalAlgo](#)**Lexicographic rank of a string**

October 6, 2012

Given a string, find its rank among all its permutations sorted lexicographically. For example, rank of "abc" is 1, rank of "acb" is 2, and rank of "cba" is 6.

44 Comments Category: [Strings](#) Tags: [MathematicalAlgo](#)

## Dynamic Programming | Set 17 (Palindrome Partitioning)

June 17, 2012

Given a string, a partitioning of the string is a palindrome partitioning if every substring of the partition is a palindrome.

83 Comments Category: [Strings](#) Tags: [Dynamic Programming](#)

## Pattern Searching | Set 7 (Boyer Moore Algorithm – Bad Character Heuristic)

Given a text `txt[0..n-1]` and a pattern `pat[0..m-1]`, write a function `search(char pat[], char txt[])` that prints all occurrences of `pat[]` in `txt[]`. You may assume that  $n > m$ .

May 26, 2012

27 Comments Category: [Strings](#) Tags: [Pattern Searching](#)

## Pattern Searching | Set 6 (Efficient Construction of Finite Automata)

May 2, 2012

In the previous post, we discussed Finite Automata based pattern searching algorithm. The FA (Finite Automata) construction method discussed in previous post takes  $O((m^3) * \text{NO\_OF\_CHARS})$  time. FA can be constructed in  $O(m * \text{NO\_OF\_CHARS})$  time.

22 Comments Category: [Strings](#) Tags: [Pattern Searching](#)

## Searching for Patterns | Set 5 (Finite Automata)

April 24, 2012

Given a text `txt[0..n-1]` and a pattern `pat[0..m-1]`, write a function `search(char pat[], char txt[])` that prints all occurrences of `pat[]` in `txt[]`. You may assume that  $n > m$ .

30 Comments Category: [Strings](#) Tags: [Pattern Searching](#)

## Check whether two strings are anagram of each other

April 12, 2012

Write a function to check whether two given strings are anagram of each other or not.

134 Comments Category: [Strings](#)

## Check whether a given string is an interleaving of two other given strings

Given three strings A, B and C. Write a function that checks whether C is an interleaving of A and B.

February 10, 2012

68 Comments Category: [Strings](#)

## Print all interleavings of given two strings

February 9, 2012

Given two strings str1 and str2, write a function that prints all interleavings of the given two strings. You may assume that all characters in both strings are different

[52 Comments](#) Category: [Strings](#)

## Print all permutations with repetition of characters

February 8, 2012

Given a string of length n, print all permutation of the given string. Repetition of characters is allowed. Print these permutations in lexicographically sorted order

[37 Comments](#) Category: [Strings](#)

## Length of the longest substring without repeating characters

December 2, 2011

Given a string, find the length of the longest substring without repeating characters. For example,

[167 Comments](#) Category: [Strings](#) Tags: [Dynamic Programming](#)

## Searching for Patterns | Set 4 (A Naive Pattern Searching Question)

July 27, 2011

Question: We have discussed Naive String matching algorithm here. Consider a situation where all characters of pattern are different.

[26 Comments](#) Category: [Strings](#) Tags: [Pattern Searching](#)

## Dynamic Programming | Set 5 (Edit Distance)

July 6, 2011

Given two strings str1 and str2 and below operations that can performed on str1. Find minimum number of edits (operations) required to convert 'str1' into 'str2'. Insert Remove Replace All of the above operations are of equal cost. Examples: Input: str1 = "geek", str2 = "gesek" Output: 1 We can convert str1 into str2 by... [Read More »](#)

[166 Comments](#) Category: [Strings](#) Tags: [Dynamic Programming](#)

## Searching for Patterns | Set 3 (Rabin-Karp Algorithm)

May 18, 2011

Given a text txt[0..n-1] and a pattern pat[0..m-1], write a function search(char pat[], char txt[]) that prints all occurrences of pat[] in txt[]. You may assume that n > m.

[55 Comments](#) Category: [Strings](#) Tags: [Pattern Searching](#)

## Searching for Patterns | Set 2 (KMP Algorithm)

April 3, 2011

Given a text txt[0..n-1] and a pattern pat[0..m-1], write a function search(char pat[], char txt[]) that prints all

occurrences of `pat[]` in `txt[]`. You may assume that  $n > m$ .

[138 Comments](#) Category: [Strings](#) Tags: [Pattern Searching](#)

## Searching for Patterns | Set 1 (Naive Pattern Searching)

April 1, 2011

Given a text `txt[0..n-1]` and a pattern `pat[0..m-1]`, write a function `search(char pat[], char txt[])` that prints all occurrences of `pat[]` in `txt[]`. You may assume that  $n > m$ .

[42 Comments](#) Category: [Strings](#) Tags: [Pattern Searching](#)

## Find the smallest window in a string containing all characters of another string

Given two strings `string1` and `string2`, find the smallest substring in `string1` containing all characters of `string2` efficiently.

March 9, 2011

[130 Comments](#) Category: [Strings](#)

[@geeksforgeeks](#), [Some rights reserved](#)

[Contact Us!](#)

[About Us!](#)

[Advertise with us!](#)