Day9

30 May 2022 16:58

Permutation of a String :-

Problem | Submissions | Solution | New Discuss

Permutations of a String

2 Difficulty: MEDIUM

Contributed By
Ankit Kharb | Level 3

Avg. time to solve
15 min

Success Rate
85%

Problem Statement Suggest Edit

You are given a string 'STR' consisting of lowercase English letters. Your task is to return all permutations of the given string in lexicographically increasing order.

String A is lexicographically less than string B, if either A is a prefix of B (and A ≠ B), or there exists such i (1 <= i <= min(|A|, |B|)), that A[i] < B[i], and for any j (1 <= j < i) A[i] = B[i]. Here |A| denotes the length of the string A.

For example :

```
If the string is "bca", then its permutations in lexicographically increasing
order are { "abc", "acb", "bac", "bca", "cab", "cba" }.
```

Note:

Sample Input 1:

```
3
abc
bc
c
```

Sample Output 1:

```
abc acb bac bca cab cba
bc cb
c
```

Explanation For Sample Input 1:

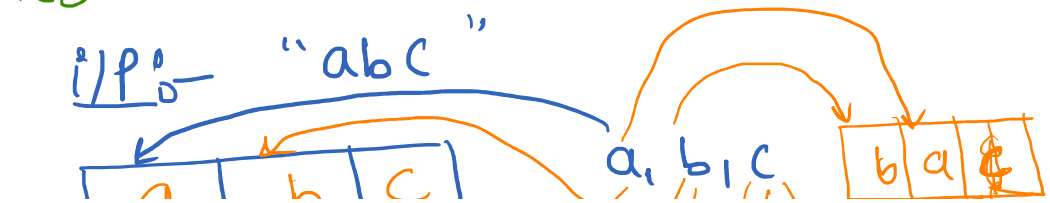
```
In the 1st test case, there are 6 permutations of the given string.
In the 2nd test case, there are 2 permutations of the given string.
In the 3rd test case, there is only 1 permutation of the given string.
```

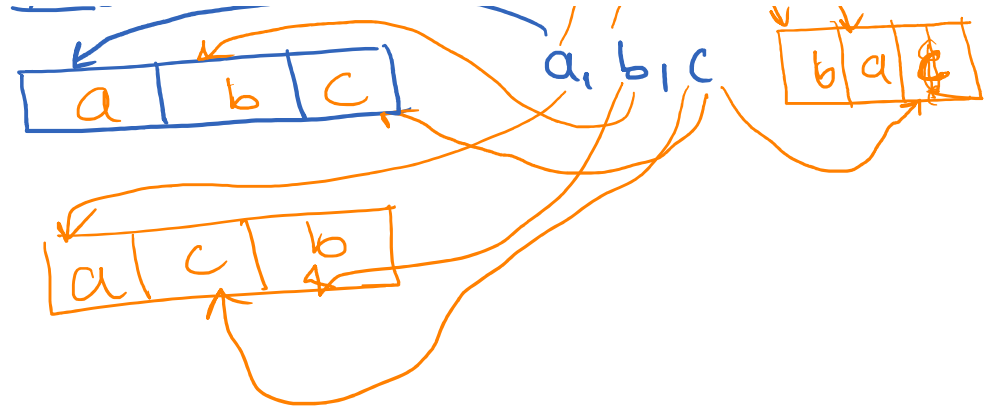
i/p :- "abc"

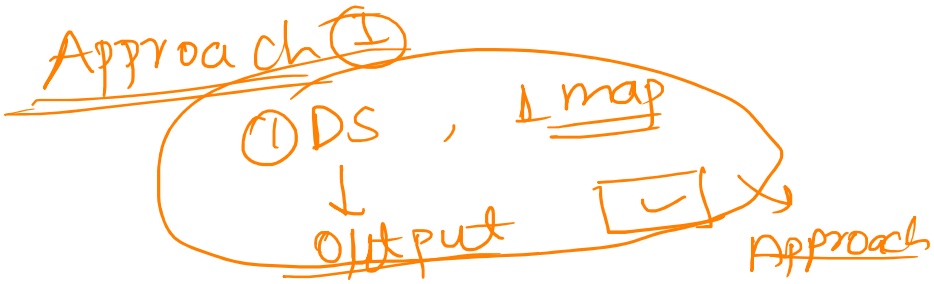
abc bac cab
acb bca cba

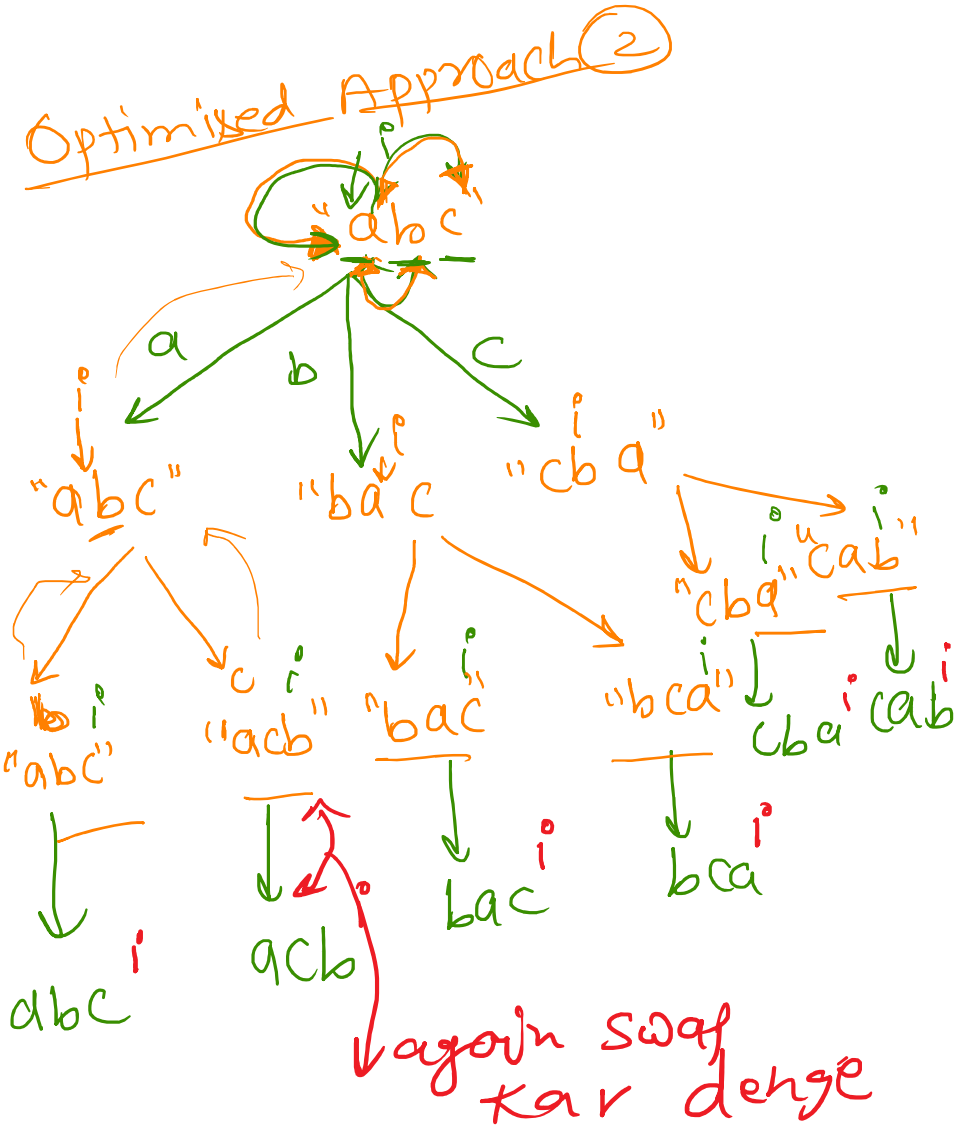
Vector<string>

i/p :- "abc"




a b c
a, b, c
b a
a c b


Approach 1
DS , map
output
Approach


Optimised Approach 2
"abc"
a
b
c
"abc"
"bac"
"cba"
"cba"
"cab"
"abc"
"acb"
"bac"
"bca"
cba
cab
abc
acb
bac
bca
again swap kar denge

and that is back tracking