Find All Subsequence of the string

Monday, November 8, 2021 7:59 AM

Method 1 (Pick and Don't Pick Concept)

From <https://www.geeksforgeeks.org/print-subsequences-string/>

Java

```
// Java program for the above approach
import java.util.*;
class GFG {

    // Declare a global list
    static List<String> al = new ArrayList<>();

    // Creating a public static Arraylist such that
    // we can store values
    // IF there is any question of returning the
    // we can directly return too// public static
    // ArrayList<String> al = new ArrayList<String>();
    public static void main(String[] args)
    {
        String s = "abcd";
        findsubsequences(s, ""); // Calling a function
        System.out.println(al);
    }

    private static void findsubsequences(String s,
                                         String ans)
    {
        if (s.length() == 0) {
            al.add(ans);
            return;
        }

        // We add adding 1st character in string
        findsubsequences(s.substring(1), ans + s.charAt(0));

        // Not adding first character of the string
        // because the concept of subsequence either
        // character will present or not
        findsubsequences(s.substring(1), ans);
    }
}
```

From <https://www.geeksforgeeks.org/print-subsequences-string/>

Method 2

Explanation :

```
Step 1: Iterate over the entire String

Step 2: Iterate from the end of string
        in order to generate different substring

        add the substring to the list
Step 3: Drop kth character from the substring obtained
        from above to generate different subsequence.

Step 4: if the subsequence is not in the list then recur
```

Below is the implementation of the approach.

From <https://www.geeksforgeeks.org/print-subsequences-string/>

Java

```java
// Java Program to print all subsequence of a
// given string.
import java.util.HashSet;

public class Subsequence {

    // Set to store all the subsequences
    static HashSet<String> st = new HashSet<>();

    // Function computes all the subsequence of an string
    static void subsequence(String str)
    {

        // Iterate over the entire string
        for (int i = 0; i < str.length(); i++) {

            // Iterate from the end of the string
            // to generate substrings
            for (int j = str.length(); j > i; j--) {
                String sub_str = str.substring(i, j);

                if (!st.contains(sub_str))
                    st.add(sub_str);
                // Drop kth character in the substring
                // and if its not in the set then recur
                for (int k = 1; k < sub_str.length() - 1;
                     k++) {
                    StringBuffer sb
                        = new StringBuffer(sub_str);

                    // Drop character from the string
                    sb.deleteCharAt(k);
                    if (!st.contains(sb))
                        ;
                    subsequence(sb.toString());
                }
            }
        }
    }

    // Driver code
    public static void main(String[] args)
    {
        String s = "aabc";
        subsequence(s);
        System.out.println(st);
    }
}
```

Output

```
aab aa aac bc b abc aabc ab ac a c
```

Method 3 :

One by one fix characters and recursively generates all subsets starting from them. After every recursive call, we remove last character so that the next permutation can be generated.

From <https://www.geeksforgeeks.org/print-subsequences-string/>

Java

```
// Java program to generate power set in
// lexicographic order.
class GFG {

    // str : Stores input string
    // n : Length of str.
    // curr : Stores current permutation
    // index : Index in current permutation, curr
    static void printSubSeqRec(String str, int n, int index,
                               String curr)
    {
        // base case
        if (index == n) {
            return;
        }
        if (curr != null && !curr.trim().isEmpty()) {
            System.out.println(curr);
        }
        for (int i = index + 1; i < n; i++) {
            curr += str.charAt(i);
            printSubSeqRec(str, n, i, curr);

            // backtracking
            curr = curr.substring(0, curr.length() - 1);
        }
    }

    // Generates power set in
    // lexicographic order.
    static void printSubSeq(String str)
    {
        int index = -1;
        String curr = "";

        printSubSeqRec(str, str.length(), index, curr);
    }

    // Driver code
    public static void main(String[] args)
    {
        String str = "cab";
        printSubSeq(str);
    }
}

// This code is contributed by PrinciRaj1992
```

Output

```
c
ca
cab
cb
a
ab
b
```

From <https://www.geeksforgeeks.org/print-subsequences-string/>