

输入字符串，输出字符串所有可能的子集并升序。

数据

字符串：ABCD
输出所有子集：

- 1) 返回一个包含所有子集的集合，按字典序升序排列。
- 2) 数据规模：1 ≤ A, B, C, D ≤ 26, 0 ≤ A, B, C, D ≤ 1000

输入描述

任意字符串字符串

输出描述

所有可能的子集并升序

用例

输入	ABCD
输出	["", "A", "B", "C", "D", "AB", "AC", "AD", "BC", "BD", "CD", "ABC", "ABD", "ACD", "BCD", "ABCD"]

题目解析

在数据上无限制，其实期望通过递归的解法实现

- 1. 任意字符串长度（递归深度）不超过1000000个元素中，期望返回1+2+3+4...，期望的解是一个递归，返回一个分子不可变，可以反复使用递归。
- 2. 任意字符串长度不超过1000000个元素，期望返回1+2+3+4...，期望的解是一个递归，返回一个分子不可变，可以反复使用递归。

解题思路如下：

定义一个函数negative，用于存储所有可能的子集并升序

定义一个递归函数negative，用于计算所有可能的子集并升序，初始化为false，返回一个包含所有可能的子集并升序

返回所有可能的子集并升序

- 1. 递归：递归函数negative，用于计算所有可能的子集并升序，初始化为false，返回一个包含所有可能的子集并升序
- 2. 递归：递归函数negative，用于计算所有可能的子集并升序，初始化为false，返回一个包含所有可能的子集并升序

数据规模：1 ≤ A, B, C, D ≤ 26, 0 ≤ A, B, C, D ≤ 1000

数据规模：1 ≤ A, B, C, D ≤ 26, 0 ≤ A, B, C, D ≤ 1000

2023.08.24

本篇为所有可能的子集并升序，返回Java或Python中升序排列。

Java实现

```
import java.util.*;

public class Solution {
    public List<String> negative(String s) {
        List<String> res = new ArrayList<>();
        if (s.length() == 0) {
            res.add("");
        }
        for (int i = 0; i < s.length(); i++) {
            String sub = s.substring(0, i);
            String sub2 = s.substring(i+1, s.length());
            List<String> res2 = negative(sub2);
            for (String str : res2) {
                res.add(sub + str);
            }
        }
        return res;
    }
}
```

Python实现

```
def negative(s):
    if s == '':
        return ['']
    res = []
    for i in range(len(s)):
        sub = s[:i]
        sub2 = s[i+1:]
        res2 = negative(sub2)
        for str in res2:
            res.append(sub + str)
    return res

s = 'ABCD'
res = negative(s)
print(res)
```

Python实现

```
def negative(s):
    if s == '':
        return ['']
    res = []
    for i in range(len(s)):
        sub = s[:i]
        sub2 = s[i+1:]
        res2 = negative(sub2)
        for str in res2:
            res.append(sub + str)
    return res

s = 'ABCD'
res = negative(s)
print(res)
```