

给定两个字符串，从字符串2中找出字符串1中的所有字符，去重并按照ASCII值从小到大排序。

输入字符串1：长度不超过1024

输入字符串2：长度不超过1000000

字符范围满足ASCII编码要求，按照ASCII的值由小到大排序

输入描述

bach  
bbacccecdg

输出描述

abc

备注

输入字符串1 为给定字符串bach，输入字符串2 bbacccecdg

从字符串2中找出字符串1的字符，去掉重复的字符，并且按照ASCII值从小到大排序，得到输出的结果为abc。

字符串1中的字符b在字符串2中找不到不能输出。

用例

输入	bach bbacccecdg
输出	abc
说明	无

题目解析

求交集问题，主要考虑Set的使用，以及数组的常用操作。

JavaScript算法源码

```
1 // 题目描述: 给定两个字符串s1和s2，求s1和s2的交集。
2 const readline = require("readline");
3
4 const rl = readline.createInterface({
5   input: process.stdin,
6   output: process.stdout,
7 });
8
9 const lines = [];
10 rl.on("line", (line) => {
11   lines.push(line);
12 });
13
14 if (lines.length == 2) {
15   const [str1, str2] = lines;
16   console.log(getIntersection(str1, str2));
17   lines.length = 0;
18 }
19
20 function getIntersection(str1, str2) {
21   const set1 = new Set(str1);
22   const set2 = new Set(str2);
23   return [...set1]
24     .filter((c) => set2.has(c))
25     .sort()
26     .join("");
27 }
28
29 // 测试用例
30 console.log(getIntersection("abc", "def"));
31 console.log(getIntersection("abc", "abc"));
32 console.log(getIntersection("abc", "ab"));
33 console.log(getIntersection("abc", "c"));
34 console.log(getIntersection("abc", ""));
35 console.log(getIntersection("", "abc"));
36 console.log(getIntersection("", ""));
37 console.log(getIntersection("a", "a"));
38 console.log(getIntersection("a", "aa"));
39 console.log(getIntersection("aa", "a"));
40 console.log(getIntersection("aa", "aa"));
41 console.log(getIntersection("a", "b"));
42 console.log(getIntersection("b", "a"));
43 console.log(getIntersection("ab", "ba"));
44 console.log(getIntersection("ab", "ab"));
45 console.log(getIntersection("ab", "baa"));
46 console.log(getIntersection("baa", "ab"));
47 console.log(getIntersection("baa", "baa"));
48 console.log(getIntersection("baa", "aaa"));
49 console.log(getIntersection("aaa", "baa"));
50 console.log(getIntersection("aaa", "aaa"));
51 console.log(getIntersection("aaa", "aaaa"));
52 console.log(getIntersection("aaaa", "aaa"));
53 console.log(getIntersection("aaaa", "aaaa"));
54 console.log(getIntersection("aaaa", "aaaaa"));
55 console.log(getIntersection("aaaaa", "aaaa"));
56 console.log(getIntersection("aaaaa", "aaaaa"));
57 console.log(getIntersection("aaaaa", "aaaaaa"));
58 console.log(getIntersection("aaaaaa", "aaaaa"));
59 console.log(getIntersection("aaaaaa", "aaaaaa"));
60 console.log(getIntersection("aaaaaa", "aaaaaaa"));
61 console.log(getIntersection("aaaaaaa", "aaaaaa"));
62 console.log(getIntersection("aaaaaaa", "aaaaaaa"));
63 console.log(getIntersection("aaaaaaa", "aaaaaaaa"));
64 console.log(getIntersection("aaaaaaaa", "aaaaaaa"));
65 console.log(getIntersection("aaaaaaaa", "aaaaaaaa"));
66 console.log(getIntersection("aaaaaaaa", "aaaaaaaaa"));
67 console.log(getIntersection("aaaaaaaaa", "aaaaaaaa"));
68 console.log(getIntersection("aaaaaaaaa", "aaaaaaaaa"));
69 console.log(getIntersection("aaaaaaaaa", "aaaaaaaaaa"));
70 console.log(getIntersection("aaaaaaaaaa", "aaaaaaaaa"));
71 console.log(getIntersection("aaaaaaaaaa", "aaaaaaaaaa"));
72 console.log(getIntersection("aaaaaaaaaa", "aaaaaaaaaaa"));
73 console.log(getIntersection("aaaaaaaaaaa", "aaaaaaaaaa"));
74 console.log(getIntersection("aaaaaaaaaaa", "aaaaaaaaaaa"));
75 console.log(getIntersection("aaaaaaaaaaa", "aaaaaaaaaaaa"));
76 console.log(getIntersection("aaaaaaaaaaaa", "aaaaaaaaaaa"));
77 console.log(getIntersection("aaaaaaaaaaaa", "aaaaaaaaaaaa"));
78 console.log(getIntersection("aaaaaaaaaaaa", "aaaaaaaaaaaaa"));
79 console.log(getIntersection("aaaaaaaaaaaaa", "aaaaaaaaaaaa"));
80 console.log(getIntersection("aaaaaaaaaaaaa", "aaaaaaaaaaaaa"));
81 console.log(getIntersection("aaaaaaaaaaaaa", "aaaaaaaaaaaaaa"));
82 console.log(getIntersection("aaaaaaaaaaaaaa", "aaaaaaaaaaaaa"));
83 console.log(getIntersection("aaaaaaaaaaaaaa", "aaaaaaaaaaaaaa"));
84 console.log(getIntersection("aaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
85 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaa"));
86 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
87 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
88 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
89 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
90 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
91 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
92 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
93 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
94 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
95 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
96 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
97 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
98 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
99 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
100 console.log(getIntersection("aaaaaaaaaaaaaaa", "aaaaaaaaaaaaaaa"));
```

Java算法源码

```
1 import java.util.Arrays;
2 import java.util.HashSet;
3 import java.util.Scanner;
4
5 public class Main {
6   public static void main(String[] args) {
7     Scanner sc = new Scanner(System.in);
8
9     String s1 = sc.next();
10    String s2 = sc.next();
11
12    System.out.println(getResult(s1, s2));
13  }
14
15  public static String getResult(String s1, String s2) {
16    HashSet set1 = new HashSet();
17    for (char c : s1.toCharArray()) set1.add(c);
18
19    HashSet set2 = new HashSet();
20    for (char c : s2.toCharArray()) set2.add(c);
21
22    Character[] characters = set1.stream().filter(c -> set2.contains(c)).toArray(Character[]::new);
23    Arrays.sort(characters);
24
25    StringBuilder sb = new StringBuilder();
26    for (Character c : characters) sb.append(c);
27
28    return sb.toString();
29  }
30 }
```

Python算法源码

```
1 # 题目描述
2 s1 = input()
3 s2 = input()
4
5 # 解题思路
6 def getResult():
7     set1 = set(s1)
8     set2 = set(s2)
9
10    set3 = set1.intersection(set2)
11
12    ans = list(set3)
13    ans.sort()
14
15    return "".join(ans)
16
17 # 测试用例
18 print(getResult())
```