

题目描述

小王在进行游戏大闯关，有一个关卡需要输入一个密码才能通过，密码获得的条件如下：

在一个密码本中，每一页都有一个由26个小写字母组成的若干位密码，每一页的密码不同，需要从这个密码本中寻找这样一个最长的密码，

从它的末尾开始依次去掉一位得到的新密码也在密码本中存在。

请输出符合要求的密码，如果有多个符合要求的密码，则返回字典序最大的密码。

若没有符合要求的密码，则返回空字符串。

输入描述

密码本由一个 **字符串数组** 组成，不同元素之间使用空格隔开，每一个元素代表密码本每一页的密码。

输出描述

一个字符串

用例

输入	h he hel hell hello
输出	hello
说明	无

输入	b ereddred bw bww bwwl bwwlm bwwln
输出	bwwln
说明	无

题目解析

本题和[华为机试 - 真正的密码_伏城之外的博客-CSDN博客](#)含义相同，解法类似。

本题存在一个疑问，那就是单字符算不算一个有效密码，比如用例

abcdef

本题用例没有相关例子，按照题目意思：

从它的末尾开始依次去掉一位得到的新密码也在密码本中存在

单字符删除末尾一位后就是空串，但是密码本中没有空串，因此是否可以认为不算有效密码呢？

但是也不能确定，因此这里我提供两种写法，一种不支持单字符密码，一种支持单字符密码

JavaScript算法源码

不支持单字符为有效密码

```
1  /* JavaScript Node ACM模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  rl.on("line", (line) => {
10    const arr = line.split(" ");
11
12    console.log(getResult(arr));
13  });
14
15  function getResult(arr) {
16    arr.sort((a, b) =>
```

JavaScript算法源码

不支持单字符为有效密码


```
1  /* JavaScript Node ACM模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  rl.on("line", (line) => {
10    const arr = line.split(" ");
11
12    console.log(getResult(arr));
13  });
14
15  function getResult(arr) {
16    arr.sort((a, b) =>
17      a.length != b.length ? a.length - b.length : a == b ? 0 : a > b ? 1 : -1
18    );
19
20    const set = new Set(arr);
21
22    while (arr.length) {
23      const str = arr.pop();
24
25      let end = str.length - 1;
26      while (set.has(str.substring(0, end))) {
27        if (end === 1) {
28          return str;
29        } else {
30          end--;
31        }
32      }
33    }
34
35    return "";
36  }
```

支持单字符为有效密码

```
1  /* JavaScript Node ACM模式 控制台输入获取 */
2  const readline = require("readline");
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout,
7  });
8
9  rl.on("line", (line) => {
10    const arr = line.split(" ");
11
12    console.log(getResult(arr));
13  });
14
15  function getResult(arr) {
16    arr.sort((a, b) =>
17      a.length != b.length ? a.length - b.length : a == b ? 0 : a > b ? 1 : -1
18    );
19
20    const set = new Set(arr);
21
22    let ans = "";
23
24    while (arr.length) {
25      const str = arr.pop();
26
27      let end = str.length - 1;
28
29      if (end === 0 && ans === "") {
30        ans = str;
31        continue;
32      }
33
34      while (set.has(str.substring(0, end))) {
35        if (end === 1) {
36          return str;
37        } else {
38          end--;
39        }
40      }
41    }
42
43    return ans;
44  }
```

Java算法源码

不支持单字符为有效密码

 伏城之外 [已关注](#)

 0  4  0  [专栏目录](#) [已订阅](#)

Java算法源码

不支持单字符为有效密码

```
1 import java.util.Arrays;
2 import java.util.HashSet;
3 import java.util.LinkedList;
4 import java.util.Scanner;
5
6 public class Main {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9
10        String[] arr = sc.nextLine().split(" ");
11        System.out.println(getResult(arr));
12    }
13
14    public static String getResult(String[] arr) {
15        Arrays.sort(arr, (a, b) -> a.length() != b.length() ? a.length() - b.length() : a.compareTo(b));
16
17        LinkedList<String> link = new LinkedList<>(Arrays.asList(arr));
18        HashSet<String> set = new HashSet<>(link);
19
20        while (link.size() > 0) {
21            String str = link.removeLast();
22            int end = str.length() - 1;
23
24            while (set.contains(str.substring(0, end))) {
25                if (end == 1) {
26                    return str;
27                } else {
28                    end--;
29                }
30            }
31        }
32
33        return "";
34    }
35 }
```

支持单字符为有效密码

```
1 import java.util.Arrays;
2 import java.util.HashSet;
3 import java.util.LinkedList;
4 import java.util.Scanner;
5
6 public class Main {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9
10        String[] arr = sc.nextLine().split(" ");
11        System.out.println(getResult(arr));
12    }
13
14    public static String getResult(String[] arr) {
15        Arrays.sort(arr, (a, b) -> a.length() != b.length() ? a.length() - b.length() : a.compareTo(b));
16
17        LinkedList<String> link = new LinkedList<>(Arrays.asList(arr));
18        HashSet<String> set = new HashSet<>(link);
19
20        String ans = "";
21
22        while (link.size() > 0) {
23            String str = link.removeLast();
24            int end = str.length() - 1;
25
26            if (end == 0 && "".equals(ans)) {
27                ans = str;
28                continue;
29            }
30
31            while (set.contains(str.substring(0, end))) {
32                if (end == 1) {
33                    return str;
34                } else {
35                    end--;
36                }
37            }
38        }
39
40        return ans;
41    }
42 }
```

Python算法源码

不支持单字符为有效密码

```
1 # 输入获取
2 arr = input().split()
```



伏城之外 已关注



0

4

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

专栏目录

已订阅

Python算法源码

不支持单字符为有效密码

```
1 # 输入获取
2 arr = input().split()
3
4
5 # 算法入口
6 def getResult(arr):
7     arr.sort(key=lambda s: (len(s), [ord(c) for c in s]))
8     setA = set(arr)
9
10    while len(arr) > 0:
11        s = arr.pop()
12        end = len(s) - 1
13
14        while s[:end] in setA:
15            if end == 1:
16                return s
17            else:
18                end -= 1
19
20    return ""
21
22
23 # 算法调用
24 print(getResult(arr))
```

支持单字符为有效密码

```
1 # 输入获取
2 arr = input().split()
3
4
5 # 算法入口
6 def getResult(arr):
7     arr.sort(key=lambda s: (len(s), [ord(c) for c in s]))
8     setA = set(arr)
9
10    ans = ""
11
12    while len(arr) > 0:
13        s = arr.pop()
14        end = len(s) - 1
15
16        if end == 0 and ans == "":
17            ans = s
18            continue
19
20        while s[:end] in setA:
21            if end == 1:
22                return s
23            else:
24                end -= 1
25
26    return ans
27
28
29 # 算法调用
30 print(getResult(arr))
```

复制