

题目描述

在一行中输入一个字符串数组，如果其中一个字符串的所有以索引0开头的子串在数组中都有，那么这个字符串就是潜在密码，

在所有潜在密码中最长的是真正的密码，如果有多个长度相同的真正的密码，那么取字典序最大的为唯一的真正的密码，求唯一的真正的密码。

输入描述

无

输出描述

无

用例

输入	h he hel hell hello o ok n ni nin ninj ninja
输出	ninja
说明	按要求，hello、ok、ninja都是潜在密码。 检查长度，hello、ninja是真正的密码。 检查字典序，ninja是唯一真正密码。

输入	a b c d f
输出	f
说明	按要求，a b c d f都是潜在密码。 检查长度，a b c d f是真正的密码。 检查字典序，f是唯一真正密码。

题目解析

我的解题思路如下：

将输入的所有字符串先按照长度升序，如果长度相同，则按照字典序升序。

这样最后一个字符串必然就是长度最长，字典序最大的，我们从最后一个字符串str开始：

接下来，我们不停地截取str的子串，即如下闭区间的子串：

- 0~str.length-2
- 0~str.length-3
- 0~str.length-4
- ....
- 0~2
- 0~1
- 0~0

然后将在输入的所有字符串中查找每次截取的子串是否存在，如果多存在则返回当str作为题解，如果有一个不存在，则直接中断查找，开始下一个str

JavaScript算法源码

```
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  });
```

## JavaScript算法源码

```
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    const set = new Set(arr);
17    arr.sort((a, b) =>
18      a.length !== b.length ? a.length - b.length : a == b ? 0 : a > b ? 1 : -1
19    );
20
21    outer: for (let i = arr.length - 1; i >= 0; i--) {
22      const str = arr[i];
23
24      for (let j = str.length - 1; j >= 1; j--) {
25        if (!set.has(str.slice(0, j))) continue outer;
26      }
27
28      return str;
29    }
30  }
```

## 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[] arr = sc.nextLine().split(" ");
10     System.out.println(getResult(arr));
11   }
12
13   public static String getResult(String[] arr) {
14     HashSet<String> set = new HashSet<>(Arrays.asList(arr));
15
16     Arrays.sort(arr, (a, b) -> a.length() != b.length() ? a.length() - b.length() : a.compareTo(b));
17
18     outer:
19     for (int i = arr.length - 1; i >= 0; i--) {
20       String str = arr[i];
21
22       for (int j = str.length() - 1; j >= 1; j--) {
23         if (!set.contains(str.substring(0, j))) continue outer;
24       }
25
26       return str;
27     }
28
29     return "";
30   }
31 }
```

## Python算法源码

```
1  # 输入获取
2  arr = input().split()
3
4
5  # 算法入口
6  def getResult(arr):
7    setArr = set(arr)
8
9    arr.sort(key=lambda s: (len(s), [ord(c) for c in s]))
10
11    for i in range(len(arr) - 1, -1, -1):
12      s = arr[i]
13      flag = True
14
15      for j in range(len(s) - 1, 0, -1):
16        if s[:j] not in setArr:
17          flag = False
18          break
19
20      if flag:
21        return s
22
```

## 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[] arr = sc.nextLine().split(" ");
10        System.out.println(getResult(arr));
11    }
12
13    public static String getResult(String[] arr) {
14        HashSet<String> set = new HashSet<>(Arrays.asList(arr));
15
16        Arrays.sort(arr, (a, b) -> a.length() != b.length() ? a.length() - b.length() : a.compareTo(b));
17
18        outer:
19        for (int i = arr.length - 1; i >= 0; i--) {
20            String str = arr[i];
21
22            for (int j = str.length() - 1; j >= 1; j--) {
23                if (!set.contains(str.substring(0, j))) continue outer;
24            }
25
26            return str;
27        }
28
29        return "";
30    }
31 }
```

## Python算法源码

```
1 # 输入获取
2 arr = input().split()
3
4
5 # 算法入口
6 def getResult(arr):
7     setArr = set(arr)
8
9     arr.sort(key=lambda s: (len(s), [ord(c) for c in s]))
10
11     for i in range(len(arr) - 1, -1, -1):
12         s = arr[i]
13         flag = True
14
15         for j in range(len(s) - 1, 0, -1):
16             if s[:j] not in setArr:
17                 flag = False
18                 break
19
20         if flag:
21             return s
22
23     return ""
24
25
26 # 算法调用
27 print(getResult(arr))
```