题目描述

给定一个乱序的数组，删除所有的重复元素，使得每个元素只出现一次，并且按照出现的次数从高到低进行排序，相同出现次数按照第一次出现顺序进行先后排序。

输入描述

一个数组

输出描述

去重排序后的数组

用例

输入	1,3,3,3,2,4,4,4,5
输出	3,4,1,2,5
备注	数组大小不超过100 数组元素值大小不超过100。

题目解析

简单的排序问题。

JavaScript算法源码

```
/* JavaScript Node ACM模式 控制台输入获取 */
const readline = require("readline");

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout,
});

rl.on("line", (line) => {
  const arr = line.split(",");
  console.log(getResult(arr));
});

/* 算法逻辑 */
function getResult(arr) {
  const count = {};
  const first = {};

  for (let i = 0; i < arr.length; i++) {
    const s = arr[i];

    if (!count[s]) count[s] = 0;
    count[s]++;

    if (!first[s]) first[s] = i;
  }

  return [...Object.keys(first)]
    .sort((a, b) =>
      count[a] != count[b] ? count[b] - count[a] : first[a] - first[b]
    )
    .join(",");
}
```

Java算法源码

```
import java.util.HashMap;
import java.util.Scanner;
import java.util.StringJoiner;

public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    String[] arr = sc.nextLine().split(",");
    System.out.println(getResult(arr));
  }

  public static String getResult(String[] arr) {
    HashMap<String, Integer> count = new HashMap<>();
    HashMap<String, Integer> first = new HashMap<>();

    for (int i = 0; i < arr.length; i++) {
      String s = arr[i];
      count.put(s, count.getOrDefault(s, 0) + 1);
      first.putIfAbsent(s, i);
    }

    StringJoiner sj = new StringJoiner(",");

    first.keySet().stream()
        .sorted(
            (a, b) -> {
              int countA = count.get(a);
              int countB = count.get(b);

              if (countA != countB) {
                return countB - countA;
              } else {
                int firstA = first.get(a);
                int firstB = first.get(b);
                return firstA - firstB;
              }
            })
        .forEach(s -> sj.add(s));

    return sj.toString();
  }
}
```

Python算法源码

```
# 输入获取
arr = input().split(",")


# 算法入口
def getResult(arr):
    count = {}
    first = {}

    for i in range(len(arr)):
        s = arr[i]

        if count.get(s) is None:
            count[s] = 0
        count[s] += 1

        if first.get(s) is None:
            first[s] = i

    tmp = list(first.keys())
    tmp.sort(key=lambda x: (-count[x], first[x]))
    return ",".join(tmp)


# 算法调用
```