

题目描述

华为商城举办了一个促销活动，如果某顾客是某一秒内最早时刻下单的顾客（可能是多个人），则可以获取免单。
请你编程计算有多少顾客可以获取免单。

输入描述

输入为 n 行数据，每一行表示一位顾客的下单时间

以（年-月-日时-分-秒.毫秒） yyyy-MM-ddHH:mm:ss.fff 形式给出。

- $0 < n < 50000$
- $2000 < yyyy < 2020$
- $0 < MM <= 12$
- $0 < dd <= 28$
- $0 <= HH <= 23$
- $0 <= mm <= 59$
- $0 <= ss <= 59$
- $0 <= ffff <= 999$

所有输入保证合法。

输出描述

输出一个整数，表示有多少顾客可以获取免单。

用例

输入	3 2019-01-01 00:00:00.001 2019-01-01 00:00:00.002 2019-01-01 00:00:00.003
输出	1
说明	样例 1 中，三个订单都是同一秒内下单，只有第一个订单最早下单，可以免单。

输入	3 2019-01-01 08:59:00.123 2019-01-01 08:59:00.123 2018-12-28 10:08:00.999
输出	3
说明	样例 2 中，前两个订单是同一秒内同一时刻（也是最早）下单，都可免单，第三个订单是当前秒内唯一——个订单（也是最早），也可免单。

输入	5 2019-01-01 00:00:00.004 2019-01-01 00:00:00.004 2019-01-01 00:00:01.006 2019-01-01 00:00:01.006 2019-01-01 00:00:01.005
输出	3
说明	样例 3 中，前两个订单是同一秒内同一时刻（也是最早）下单，第三第四个订单不是当前秒内最早下单，不可免单，第五个订单可以免单。

题目解析

考察 [数组排序](#) 以及字符串操作。

JavaScript算法源码

```
1  /* JavaScript Node ACN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  const lines = [];
10 let n;
11 rl.on("line", (line) => {
12   lines.push(line);
13 })
```

```

14     if (lines.length == 1) {
15         n = lines[0] - 0;
16     }
17
18     if (n && lines.length == n + 1) {
19         lines.shift();
20         console.log(getResult(lines));
21         lines.length = 0;
22     }
23 });
24
25 function getResult(arr) {
26     arr.sort().reverse();
27
28     const stack = [arr.pop()];
29
30     while (arr.length) {
31         const time = arr.pop();
32         const top = stack.at(-1);
33         if (top === time || top.substring(0, 19) !== time.substring(0, 19)) {
34             stack.push(time);
35         }
36     }
37
38     return stack.length;
39 }

```

Java算法源码

```

1 import java.util.Arrays;
2 import java.util.LinkedList;
3 import java.util.Scanner;
4
5 public class Main {
6     // 输入获取
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9
10        int n = Integer.parseInt(sc.nextLine());
11
12        String[] arr = new String[n];
13        for (int i = 0; i < n; i++) {
14            arr[i] = sc.nextLine();
15        }
16
17        System.out.println(getResult(arr));
18    }
19 }

```

```

20 // 算法入口
21 public static int getResult(String[] arr) {
22     Arrays.sort(arr);
23
24     LinkedList<String> stack = new LinkedList<>();
25     stack.add(arr[0]);
26
27     int i = 1;
28     while (i < arr.length) {
29         String time = arr[i++];
30         String top = stack.getLast();
31
32         if (top.equals(time) || !top.substring(0, 19).equals(time.substring(0, 19))) {
33             stack.add(time);
34         }
35     }
36
37     return stack.size();
38 }
39 }

```

Python算法源码

```

1 # 输入获取
2 n = int(input())
3 arr = [input() for _ in range(n)]
4
5
6 # 算法入口
7 def getResult():
8     arr.sort(reverse=True)
9     stack = [arr.pop()]
10
11     while len(arr) > 0:
12         time = arr.pop()
13         top = stack[-1]
14
15         if top == time or top[:19] != time[:19]:
16             stack.append(time)
17
18     return len(stack)
19
20
21 # 算法调用
22 print(getResult())

```