

{% note info %} **摘要** Title: 1913. 公平摄影 Tag: 枚举、前缀和、哈希表 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by: NEFU AB-IN

[Link](#)

@TOC

## 1913. 公平摄影

### • 题意

农夫约翰的  $N$  头奶牛站在一维长围栏的不同位置。第  $i$  头牛位于位置  $x_i$ ，其所属品种为  $b_i$ （根西岛牛或荷斯坦牛）。所有奶牛的位置各不相同。约翰想给一段连续区间内的奶牛拍摄一张照片，用来在乡村集市上展览。但是我们希望他所有品种的奶牛都能在照片中得到公平的展示。因此，他希望确保无论照片中出現哪些品种的奶牛，每种品种的奶牛在照片中的数量都必须相等。例如，一张照片中只包含荷斯坦牛是可以的，包含荷斯坦牛和根西岛牛各 27 头也没问题，但是包含 10 头荷斯坦牛和 9 头根西岛牛则不可以。请确定，约翰可以拍下的满足以上条件的照片的最大尺寸。照片的尺寸是指照片中奶牛最大和最小位置之间的差。约翰最终可能只拍下一头奶牛，这种情况下，照片尺寸为 0。

### • 思路

题目可以转换为两种模型

- **求两个种类数量相等的连续最长区间**
  - 可以想到用**前缀和**来做，两个种类，一个为  $1$ ，一个为  $-1$
  - 对种类进行前缀和的维护，记录每个前缀和最早出现的下标，当这个下标被标记过了，说明前面有这个值了，说明存在一段区间和为 0，说明存在一段区间两个种类数量相等
  - 记录每个前缀和最早出现的下标，可以用**哈希表**来实现
- **求同一种类的连续最长区间**
  - 类似于**双指针**，两个指针  $last$  和  $i$ ， $i$  一直递增
  - 设定一个  $last$  变量用来标记新的种类的起始下标
  - 当  $last$  刚开始，或者当前种类与前一个种类不相等时，说明  $last$  需要设定为新值

`lst[i].x`

此题还有一个难点，就是 **前缀和和下标相减的下标不对应** 比如答案为  $x_i - x_j$ ，而此时的前缀和为  $S_i - S_{j-1}$  所以我们可以设  $S'_j = S_{j-1}$ ，即是不包含  $lst[i].op$  的前缀和，即左闭右开的前缀和这样坐标就统一了

### • 代码

...

Author: NEFU AB-IN

Date: 2022-01-21 18:24:16

```
FilePath: \ACM\Acwing\1913.py
LastEditTime: 2022-01-21 21:34:49
'''

from collections import Counter

class Node(object):
    def __init__(self, x, op):
        self.x = x
        self.op = op

    def __lt__(self, a):
        return self.x < a.x

    def __repr__(self):
        return f"[{self.x}, {self.op}]"

N = int(1e5 + 10)
d = Counter()
lst = []

if __name__ == "__main__":
    n = int(input())
    for i in range(n):
        x, b = input().split()
        x = int(x)
        op = 1 if b == 'G' else -1
        lst.append(Node(x, op))
    lst.sort()

    # 求连续最长的相同字母子串
    last, res, sum = 0, 0, 0
    for i in range(n):
        if i == 0 or lst[i].op != lst[i - 1].op:
            last = lst[i].x
            res = max(res, lst[i].x - last)

        for i in range(n):
            # d用来存si'的前缀和的最早下标x, 即左闭右开 (即不包含lst[i].x的i的前缀和)
            # 如果没标记过, 就标记上, 这样保证是最早出现的
            if sum not in d:
                d[sum] = lst[i].x
            sum += lst[i].op
            if sum in d:
                res = max(res, lst[i].x - d[sum])
    print(res)
```