

HW#8

经济学院 刘安澜 1700015495

723A. The New Year: Meeting Friends

implementation, math, sorting, 800,

<https://codeforces.com/problemset/problem/723/A>

解题思路：题目的样例非常有迷惑性，很容易让人直接想到用三个坐标的平均数来做；但是这样并不一定达到最优，所以还应该遍历解决问题。所以首先排序，在坐标最小值和最大值之间的区间上遍历一遍距离就可以。

```
a,b,c=sorted(map(int,(input().split())))
m=b+c-2*a
for i in range(a,c+1):
    m=min(m,abs(a-i)+abs(b-i)+abs(c-i))
print(m)
```

705A. Hulk

implementation, 800,

<https://codeforces.com/problemset/problem/705/A>

解题思路：这题是很久以前刷的，当时尝试了几种写法都出现混乱，所以最后偷懒就按照奇偶数分类考虑，批量输出中间的 that I love that I hate，最后再按照奇偶分别添加结尾即可。

```
n=int(input())
print('I hate'+ ' that I love that I hate'*(n//2)+' it' if n%2==1
else 'I hate'+ ' that I love that I hate'*((n-1)//2)+' that I
love'+ ' it')
```

200B. Drinks

implementation, math, 800,

<https://codeforces.com/problemset/problem/200/B>

解题思路：就是对于每杯饮料的浓度取平均值即可。

```
n=int(input())
l=list(map(int,input().split()))
s=0
for i in range(n):
    s+=l[i]
print(s/n)
```

492B. Vanya and Lanterns

binary search/implementation/math/sortings, 1200,

<https://codeforces.com/problemset/problem/492/B>

解题思路：除了需要找到路灯之间的最大距离的一半之外，还需要考虑路灯距离路边缘的最大距离；二者的最大值为路灯照明距离的最小值。

```
n,l=map(int,input().split())
m=list(sorted({int(x) for x in input().split()}))
m2=max(m[0],l-m[-1])
for i in range(len(m)-1):
    m2=max(m2,(m[i+1]-m[i])/2)
print(m2)
```

//选做：

在连续 3 周选做题陷入超时的困扰之后，本周在写程序时特别注意到了循环的简洁性，两道选做题都是一次 AC，非常有成就感。

466C. Number of Ways

binary search/brute force/data structures/dp/two pointers, 1700,
<https://codeforces.com/problemset/problem/466/C>

解题思路：

第一次看到本题让我一下想到了 arrival time 的概念；应用这个思想，只需要找到和为 $1/3$ 总和（下称 1 类点）与和为 $2/3$ 总和（下称 2 类点）的点，符合要求的组合需要让 1 类点出现在 2 类点前面。对于这种情况，我们需要将 1 类和 2 类点分别计数，只要对于每一个 2 类点都加上它前面 1 类点的个数就可以得到答案；运算复杂度为 $O(n)$ 。另外，对于和为 0 的情况需要额外考量，将中间所有和为 0 的点（1 类点个数-1）两两组合即可。

解题代码：

```
n=int(input())
l=[int(x) for x in input().split()]
m=0
s=0
k=sum(l)
if k%3!=0:
    print(0)
else:
    j=0
    for i in range(n):
        s+=l[i]
        if s==k//3:
            j+=1
        if s==(k//3)*2:
            m+=j
    print((j-1)*(j-2)//2 if k==0 else m)
```

1443C. The Delivery Dilemma

binary search/greedy/sortings, 1400,
<https://codeforces.com/problemset/problem/1443/C>

解题思路：

总用时不会大于外卖时间的最大值。所以建立二维数组方便形成对应关系，根据外卖时间倒序排列数组，然后从外卖时间最长的改为自取，把相应的自取时间加总，与剩余外卖时间的最大值取最大值为新的最优时间，将列表全部遍历一遍打擂台得到最优。另外，循环到最后一个时只剩下全部自取时间的加总，需要额外多写一步。

可能的优化：如果加总的自取时间大于现有外卖时间最大值就可以退出循环，减少循环次数。

另外，本题的输出比较多，根据同学的提示一步输出就不会超时。

解题代码：

```
n=int(input())
output=[]
for i in range(n):
    m=int(input())
    a=[int(x) for x in input().split()]
    b=[int(x) for x in input().split()]
    l=([0]*2) for i in range(m)]
    for j in range(m):
        l[j][0]=a[j]
        l[j][1]=b[j]
    l.sort(key=lambda x:x[0],reverse=True)
    t=l[0][0]
    s=0
    for j in range(m-1):
        s+=l[j][1]
        t=min(t,max(s,l[j+1][0]))
    output.append(str(min(t,s+l[m-1][1])))
print('\n'.join(output))
```
