

HW#6

经济学院 刘安澜 1700015495

266A. Stones on the Table

implementation, 800, x65000+,

<http://codeforces.com/problemset/problem/266/A>

解题思路：遍历列表数出相邻元素相同的个数，即为答案。

```
n=int(input())
l=input()
m=0
for i in range(n-1):
    if l[i]==l[i+1]:
        m+=1
print(m)
```

96A. Football

implementation/strings, 900, x65000+,

<http://codeforces.com/problemset/problem/96/A>

解题思路：如果字符串中没有'1111111'和'0000000'就可以判定不是“危险时刻”。

```
l=input()
if l.find('1111111')== -1 and l.find('0000000')== -1:
    print('NO')
else:
    print('YES')
```

615A. Bulbs

implementation, 800, <http://codeforces.com/contest/615/problem/A>

解题思路：循环两遍；在子行中再次循环将剩余的元素添加进列表，最后用 set 函数去除重复元素并和总的灯数比较大小即可。

```
n,m=map(int,input().split())
l=[]
for i in range(n):
    k=[int(x) for x in input().split()]
    for j in range(k[0]):
        l.append(k[j+1])
print('YES' if set(l)==set(range(1,m+1)) else 'NO')
```

236A. Boy or Girl

brute force, implementation, strings, 800,

<https://codeforces.com/problemset/problem/236/A>

解题思路：找不同元素的个数只需要运用集合，最后比较集合的长度是否是偶数即可。

```
print('CHAT WITH HER!' if len(set(input()))%2==0 else 'IGNORE HIM!')
```

69A. Young Physicist

implementation, math, 1000,

<https://codeforces.com/problemset/problem/69/A>

解题思路：建立三个计数变量；遍历输入行，并循环更新计数变量，如果三个计数变量都为0那么输出YES。

```
n=int(input())
x=0
y=0
z=0
for i in range(n):
    x,y,z=map(int,input().split())
    x+=x
    y+=y
    z+=z
print('YES' if x==0 and y==0 and z==0 else 'NO')
```

//选做：

1000B. Light It Up

greedy, 1500, <https://codeforces.com/problemset/problem/1000/B>

解题思路：

这道题目如果暴力循环做就会超时；所以解决超时问题的关键是找到遍历时上一次和下一次时间之间的数学关系，而不是再次列表计算（复杂度 $O(n^2)$ ）

构造截止到某时刻的开灯总时长列表，由于只可能在关灯后的1秒插入为最优解，所以这样就方便我们构造插入点后的时间的数学公式；在列表中将关灯后且间隔区间大于1的时刻全部遍历一遍即可。

这道题改了很久超时的原因，一方面是没有想清楚要在关灯后插入是最优解，还有一方面是构造的列表是时间间隔的列表，每次循环时计算时长都需要再切分调用，复杂度就非常高了。

解题代码：

```
n,m=map(int,input().split())
l=[0]+[int(x) for x in input().split()]+[m]
b=[0]*(n+1)
b[0]=l[1]-l[0]
for i in range(1,n+1):
    b[i]=b[i-1]+(l[i+1]-l[i])*((i+1)%2)
```

```

a=b[-1]
for i in range(1,n+1,2):
    if l[i+1]-l[i]>1:
        a=max(a,b[i-1]+m-b[-1]-l[i]+b[i-1]-1)
print(a)

```

此外我还参考了老师在群中提供的另一个同学的思路，利用添加断点时对于开关灯时间的反射构造 $b(n)=n$ 时刻前开灯总时长- n 时刻前关灯总时长；如果在 n 时刻添加断点，那么此时 $b(n)' = 2b(n)-b(m)$ ，由于 $b(m)$ 是定值，所以使添加断点后开灯时间时长最长即等价于寻找 $b(n)$ 最大的非节点。

很容易得知让 $b(n)$ 最大，添加断点必须在关灯时刻之后的下一时刻，此时 $b(n+1)=b(n)-1$ ；所以只需要根据节点算每一个节点上的 $b(n)$ ，并找到最大的 $b(n)$ ， $n+1$ 就是我们找的断点，根据数学公式可推知总开灯时长。如果不添加断点，那么 $b(n)$ 最大值出现在末尾，最后添加一步取最大值输出即可。

解题代码：

```

n,m=map(int,input().split())
l=[int(x) for x in input().split()]
j=0
b=[0]*(n+1)
b[0]=l[0]
for i in range(0,n-1):
    b[i+1]=b[i]+(-1+2*(i%2))*(l[i+1]-l[i])
b[-1]=b[n-1]+(1-2*(n%2))*(m-l[-1])
print(max((b[-1]+m)//2,(m+2*max(b)-b[-1]-1)//2))

```

总结：数据量大、难度大的题目如果超时，需要尽量找每个变量之间的数学关系来节省运算时间；本题构造 $b(n)$ 确实是非常巧的方法，但是更重要的还是通过寻找规律+选择最优的存储数据+推导公式来减少循环的复杂度。