

三月月赛题目讨论

➤ 问题1：果老师的计算器

考察点：模拟

倒着模拟一遍即可。

➤ 问题1：果老师的计算器

参考代码：

```
for(int i = N; i >= 1; i--) {  
    int opt = a[i], t = b[i];  
    if(opt == 1) {  
        X -= t;  
    } else if(opt == 2) {  
        X += t;  
    } else if(opt == 3) {  
        X /= t;  
    } else {  
        X *= t;  
    }  
}  
cout << X;
```

➤ 问题2：果老师的砝码

考察点：观察规律

每个砝码可以放左边，不放和放右边，很容易联想到三进制，所以按照3进制排列即可，选择砝码如下：1,3,9,27,...，选m个数最多能称出的重量为： $3^0+3^1+3^2+3^3+\dots+3^{(m-1)}$ 。

所以我们就按这个式子求出第一个大于n的数是3的几次方就好了，因为数据范围很大，需要用到高精度，因为对于只需要1个数的只有1，需要2个数的最大为4，需要3个数的最大为13，需要4个数的最大为40，可以得到 $1+3=4$ ， $4+9=13$ ， $13+27=40$ ，所以就有了 $m = m * 3 + 1$ 这个式子，当然这道题也可以用大数去写。

`long double` 表示的范围是 $-1.2 \times 10^{-4932} \sim 1.2 \times 10^{4932}$

➤ 问题2：果老师的砒码

参考代码：

```
int main() {  
    int s = 1;  
    long double n, sum = 1, x = 1;  
    cin >> n;  
    while(sum < n) {  
        x *= 3;  
        sum += x;  
        s++;  
    }  
    cout << s << endl;  
    return 0;  
}
```

➤ 问题3：项链染色

考察点：贪心

优先选择喜爱程度大的颜色进行染色

➤ 问题3：项链染色

参考代码：

```
struct se{long long a,b;};
ooo(se a,se b){return a.b>b.b;}
int n,m;se a[100005];long long t;
int main()
{
    cin>>n>>m;
    for(int i=1;i<=m;i++)cin>>a[i].a;
    for(int i=1;i<=m;i++)cin>>a[i].b;
    sort(a+1,a+1+m,ooo);
    for(int i=1;i<=m;i++)
    {
        if(n<a[i].a){cout<<t+a[i].b*n;return 0;}
        t+=a[i].b*a[i].a;n-=a[i].a;
    }
    cout<<t;
}
```

➤ 问题4：训练

考察点：贪心

a - 刷掉本题要花费的时间， **b** - 本题每分钟会带来的疲倦值

题贪心的方法是：优先做 a_i / b_i 小的题目。

直接理解 a_i / b_i 小，首先分母 b_i 要大，就是本题每分钟带来的疲劳值较大（因此该题就需要优先做），其次分子 a_i 要小，就是本题花费的时间尽量少（这样后面的题的疲劳 * 本题花费的时间就尽量小）。

那么从排序的角度来看，有2个题目 x 和 y 。

当 x 在前，产生的疲劳值为 $a_x * b_y$

当 y 在前，产生的疲劳值为 $a_y * b_x$

那么排序时令 x 排在前面的标准就是 $a_x * b_y < a_y * b_x$ ，即 $a_x / b_x < a_y / b_y$

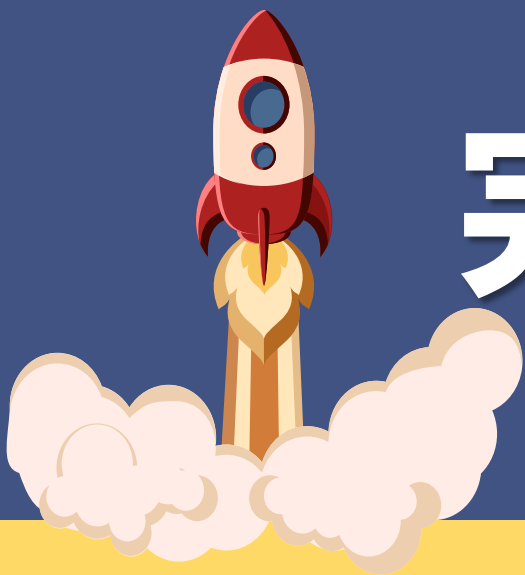
对排序后序列的任意两个相邻题目，都是如此，即 a_i / b_i 小的在前。

➤ 问题4：训练

参考代码：

```
struct work
{
    ll a, b;
    double x;
}W[MAXN];
bool cmp(work w1, work w2)
{
    return w1.x > w2.x;
}
```

```
int main()
{
    int n;
    cin >> n;
    ll ans = 0;
    ll sum = 0;
    for(int i=0; i<n; i++)
    {
        cin >> W[i].a >> W[i].b;
        W[i].x = 1.0 * W[i].b / W[i].a;
        sum += W[i].b;
    }
    sort(W, W+n, cmp);
    for(int i=0; i<n; i++)
    {
        sum -= W[i].b;
        ans += sum * W[i].a;
    }
    cout << ans << endl;
    return 0;
}
```



完！

以梦为码 心之所往

