解题报告

A: 雷神之路

题意

```
  初始位置在0，走到n(n<=10^18)，每次可以走1、2、3步，但是路上有m(m<=500)个地雷，有地雷的地方不可以走。问走到n有多少种方法?
```

分析

```
   每次可以走1、2、3步，以dp[i]表示走到i点有多少种方法，那么方程为
dp[i]=dp[i-1]+dp[i-2]+dp[i-3]，有n的范围可以知道要用矩阵快速幂加速。有地雷就不能直接算了，但是地雷与地雷之间还是可以用矩阵快速幂去算，所以就分块去算就好了。注意地雷之间的距离很近的时候，要特判。
```

思考

```
  初始条件设立不方便，因为从零开始，可以把所有坐标都加2，这样就方便多了。后面也要注意到坐标始处理过的了。
```

B: Snowdrop修长廊

题意

```
  有n个点需要覆盖，一次覆盖可以覆盖x点到y点，代价是w+(y-x)^2，如果x==y，代价就是w。问覆盖的最小代价是多少。
```

分析

```
  状态方程很好想，dp[i]=min(dp[j]+w+(x[i]-x[j+1])^2, j<i)。但是状态转移是二维的，n太大，铁定会超时。这时就要考虑下斜率优化，假设k<j<i，如果在j的时候决策要比在k的时候决策好，那么也是就是dp[j]+w+(x[i]-x[j+1])^2<dp[k]+w+(x[i]-x[k+1])^2。两边移项一下，得到：(dp[j]+x[j+1]^2-(dp[k]+x[k+1]^2))/(2*(x[j+1]-x[k+1]))<x[i]。我们把dp[j]-num[j]^2看做是yj，把2*num[j]看成是xj,这样就构造出了斜率表达式，据此就可以把一些永远不能成为最优解的点删掉从而降低复杂度到O(n).
```

思考

```
  注意是点的覆盖，不是区间的覆盖。
```

C: TaoSama与煎饼

题意

```
  初始位置在1点，要走到N点，有四种卡片，分别可以走1、2、3、4步，走到一个地方时，获得那个地方的值。保证所给的卡片可以走到n，同时每种卡片的数量不超过40.
```

分析

```
  因为说了每种卡片的数量不超过40，数量那么小是个暗示。那么就可以以每种卡片的数量为状态，40^4的空间也不会超。dp[x,y,k,z] 表示四种卡片分别取x、y、z、k达到位置1+x+y*2+k*3+z*4的最大数目。这个状态分别由dp[x-1,y,k,z]、dp[x,y-1,k,z]、dp[x,y,k-1,z]、dp[x,y,k,z-1]转移而来。
```

思考

```
  注意初始状态就好了。
```

E: Goozy的积木

题意

```
  给出n块积木和告高度，问是否能搭出两个高度相同的塔?
```

分析

```
  用dp[i][j]表示用前n快积木搭出高度差为j的双塔中较矮的塔高度。
  dp[i][j]=max
  {
    dp[i-1][j-a[i]] //搭到较高的塔上
    dp[i-1][j+a[i]]+a[i] // 搭到较矮的塔上
    dp[i-1][a[i]-j]+a[i]-j // 搭到较矮的塔变为较高的塔
    dp[i-1][j]  //  不选
  }
  由于n很大，要用滚动数组优化。
```

思考

```
注意初始状态就好了。
```

F: 先锋看烟花

题意

有n个区域1～n，每个区域之间距离为1，有m个烟火要放，给定放的地点a[i]、时间t[i] ,如果你当时在区域x，那么你可以获得b[i]-|a[i]-x|的happiness。你每个单位时间可以移动不超过d个单位距离，求获取到的最大的happiness值。

分析

设dp[i][j]为到放第i个烟花的时候站在j的位置可以获得的最大happiness。那么我们可以很容易写出转移方程：dp[i][j]=max(dp[i-1][k]+b[i]-|a[i]-j|，其中max(1,j-t*d)<=k<=min(n,j+t*d)。因为计算式中b[i]跟决策没关系，然后反过来求|a[i]-x|的和的最小值即可。由于是求一段区间的最小值，我们可以想到用单调队列维护，维护一个单调升的队列。以每个t[i]时刻为基准，枚举每个观赏地点，计算从t[i-1]到t[i]可以移动的格数从而得出可以移动的范围，在这个范围里面找最小值加过去就行了。因为是枚举观赏地点，所以可以发现移动的范围相当于一个滑动窗口，随着枚举向右移动。

思考

注意问题的转化。

H: 又见背包

题意

经典的多重背包问题。

分析

本以为二进制优化可以过的，结果被卡了。用传统的多重背包的单调队列优化就可以了。

思考

还是要计算一下复杂度再去决定算法。

I: Mingo's Game

题意

题意比较复杂，其实就是前不久做的一道CF题目。
要把1～n分成k组，每组内的数必须连续，组与组不相交且每个数必须属于一个组，并且任意i有一个参数ti。走到下一个节点的公式给出，求从1走到n的最小期望时间。

分析

先推期望公式。设sum[i]=∑ti，rev[i]=∑1/ti，那么从1走到i的期望exc[i]=exc[i-1]+sum[i]ti。然后观察得到从l走到r的期望公式exc[l][r]＝exc[r]-exc[l-1]-sum[l-1]*(rev[r]-rev[l-1])。然后设dp[i][j]表示以前i个数成j段的最小费用。转移：dp[i][j]=min(dp[l][j-1]+exc[l+1][i])。发现这是个O(n*n*k)的转移。推出斜率公式：

slope(j,k)=(y(j,p)-y(k,p))/(sum[j]-sum[k]);

其中y(x,p)=dp[x][p]-exc[x-1]+sum[x-1]*rev[x-1]。

并且推出结论若k<j<i且g(j,k)<rev[i]，那么对于i来说，j要优于k。由于rev[i]随i递增而递增，当求解l的时候，只需要维护单调队列的top最优就可以了。

思考

确实很难想，特别是斜率的公式。

L: 来签个到吧

题意

给出一些数的集合，对任意球x、y，使集合内|x-y|存在，问需要加入多少个球。之后，有放回的取球，问取球的次数的期望是多少。

分析

首先对第一个问题，不难发现最终球的个数是最大值处以这些数公共的gcd值。得出球的个数之后，求期望就很简单了，设个数为m，ans＝∑m/i,1<=i<=m.

思考

注意球上的数可能是0，0存在的时候是计算gcd值的时候是会被忽略掉的。