126

带权平均数之和

令$sum[i]=\sum_{j=1}^{i}a[j]$

令$b[i]=\sum_{j=1}^{i}w[j]$

$f(l,r)=w[r-l+1]*(sum[r]-sum[l-1])$

考虑$sum[r]$的贡献系数，就是以r为右端点的所有区间的长度的w值

以r为右端点的所有区间的长度是$1..r$的，所以$sum[r]$的贡献系数是$b[r]$

再考虑$-sum[l-1]$的贡献系数，也就是以l为左端点的所有区间的长度的w值

所以$-sum[l-1]$的贡献系数是$b[n-l+1]$

所以答案就是$\sum_{i=1}^{n}sum[i]*(b[i]-b[n-i])$

129

2 Climb

2.1 $O(n^2)$

显然，除了最后一天外，药丸按照 A_i-B_i 降序排列使用最优，那么我们不妨枚举哪一颗药丸最后一天吃，把剩下的按照 A_i-B_i 降序排列，枚举使用前几个即可。

2.2 Solution

我们不妨还是枚举每个药 k，假设 k 最后一天喝，那么我们可以在剩下的药丸里二分得到我们还需要吃哪些药丸。通过 RMQ 区间最小值我们可以方便的判断拿掉 k 之后 A_i-B_i 的前缀和是否会在某个时刻小于等于 C_i，细节见代码。

106

2 石头剪刀布

考虑一个最简单的问题，如果给出一个序列，怎么求它的 w 值呢？
一个最直接的做法是记 dp[i]表示 i 这个位置结尾最长的长度，转移为 max(dp[j]+1) 其中 j 能够战胜 i，然后这个 w 值就是 dp 值的最小值，总的时间复杂度为 O(n^2)，结合枚举能获得 20 分。

我们换个想法，记一下做到位置 i，结尾为剪刀石头布的的最长序列是多长，分别记作 p0, p1, p2。那么比如 i+1 这个位置是剪刀，我们就用 max(p1+1, p0)去更新一下 p0 这个值，也就是说如果选了这个剪刀，那么上一个位置必须是石头，所以最长序列的长度为 p1+1，否则是 p0。最后就是 max(p0, p1, p2)，时间复杂度 O(n)，结合枚举还是只能获得 20 分。

于是我们可以设计一个四维的状态，dp[i][p0][p1][p2]表示做到第 i 个位置，结尾为剪刀，石头，布的最长序列分别为 p0, p1, p2 的方案数。如果 i+1 这个位置我们选了剪刀，那么就转移到 dp[i+1][max(p0,p1+1)][p1][p2]这个地方。最后我们把 dp[n][p1][p2][p3]加到答案 answer[max(p1,p2,p3)]里面，表示有这么多个序列的 w 值为 max(p1,p2,p3)。时间复杂度为 O(n^4)，可以获得 40 分。

通过观察我们会发现这个 dp 的有用状态非常少，具体的我们有|p0-p1|<=2, |p0-p2|<=2，对于一个长度为 p0 的以剪刀结尾的序列，那么我们把最后一项去掉就能得到一个以石头结尾的序列长度，所以有 p1>=p0-1，同理有 p2>=p1-1, p0>=p2-1，结合这三式就能得到这三项的差不会超过 2。

所以我们可以把上面的状态改写成 dp[i][p0][d1][d2]其中 d1, d2 为-2 到 2 之间的数字，表示 p1-p0 和 p2-p0 的值，然后用上面说的方法转移和统计答案即可。

时间复杂度为 O(n^2)，其中可能有一个较大的常数，可以获得 100 分。如果你实现常数很大或者在 40 分算法的基础上直接使用 map 优化或者你只发现了可以压缩一维，那么可以获得 70 分。

396

T3

显然问题就是判定一组同余方程有没有解。
说一个不用 CRT 的方法，一个 $\bmod n$ 的约束可以转化成对所有 $\bmod d(d|n)$ 的约束，之后就只需要判断约束之间有没有冲突就行了，$O(n\log n)$ 。
把这个做法改一改可以线性，由于质因子分解是线性的，因此我们只需要统计所有 $\bmod p^k$ 的结果就行了。