T1 小L的游戏(game)

枚举先打死哪只小怪兽，取最优解。以下为了说明方便，不妨先打死 1 号小怪兽。

设 $f(p)$ 表示最小的 x 使得 $\sum_{i=1}^{x} i \geq p$ 。因为一定可以打出恰好 hp_1 点攻击 $(\star)$ 击杀第一只小怪兽，所以两只怪兽的最优死亡的时间分别是 $f(hp_1)$ 和 $f(hp_1+hp_2)$ 。这样就可以做到最小化伤害，为 $f(hp_1)*atk_1+f(hp_1+hp_2)*atk_2$。

对于$(\star)$的正确性：我们可以发现，在击杀第一只怪兽的最后一击 $f(hp_1)$ 中，被浪费的攻击力严格小于 $f(hp_1)$ 所以把被浪费的攻击力的那一天打另一只即可。

T2 小L的楼梯(stair)

设DP状态 $dp[i]$ 表示走到第 i 节台阶的方案数，则转移为：

$$dp[i] = \sum_{j=\max(1,i-k)}^{i-1} dp[j].$$

一种 $O(nk)$ 的做法是从小到大枚举 i 然后再枚举 j 求和。可以使用前缀和优化求和的过程做到 $O(n)$ 。

T3 小L的旅行(travel)

首先，我们要看到问题实际上在说：规定序列的下一位须与当前位在图上相邻，求最长自由度上升序列。于是我们就得到了DP状态：$dp[pos]$表示从pos号景点出发的最长自由度上升序列的长度。这里，我们用f来表示自由度，Neighbor表示由道路直接相连的点的集合，则转移为：

$$dp[pos] = \max_{nxt \in Neighbor(pos), f[nxt] > f[pos]} dp[nxt] + 1.$$

可以按照自由度排序进行转移，也可以直接记忆化搜索，会更加方便好写。由于每一条边只会被两个点访问到，所以时间复杂度是$O(m)$的。

T4 小L的flappy(bird)

DP状态为 $dp[i][0/1]$ 表示选第 i 个，且第 i 个比第 $i-1$ 个大/小。朴素的转移为

$$dp[i][0] = \max_{j<i,h[j]<h[i]} dp[j][1] + 1(\star)$$
$$dp[i][1] = \max_{j<i,h[j]>h[i]} dp[j][0] + 1$$

直接枚举复杂度为 $O(n^2)$ ，可以获得80%的分数。

发现这个形式就是拦截导弹，进而观察是否满足拦截导弹的性质。发现有些不同，并不能保证 $a[i]$ 的性质。但是按照该思路思考，对($\star$)来说，如果 $dp[j][1]$ 的值更低但是 $h[j]$ 更高的话，一定不优。所以在 h 单调增的情况下，dp 值应当单调增。所以转移的时候可以用二分找到使得 h 比当前高度小的最大的 dp 值进行转移。考虑维护这样的序列，修改时：

1. 如果当前结果比栈顶结果严格更优，则弹栈。并且重复步骤1直到不满足。
2. 如果当前结果比栈顶结果严格更劣，则不加入当前结果，并且结束该过程。
3. 此时要么当前 h 和 dp 值都比栈顶高，要么h 和 dp 值都比栈顶低。对于前者，我们将当前结果加入栈中；对于后者，考虑栈顶元素 i' 的 dp 转移是来自 j' 的，显然我们可以从 j' 转移到当前的 i 使得 dp 值更高，与之前我们的转移相矛盾，所以不会出现这样的情况。