

2018 NOIP 普及组热身赛 Day1 题解

列车抢票

比较简单的模拟题，实现以 if-else 为主，但是考虑到硬卧 b 和订单总数 $\sum x_i$ 的数据量较大，所以如果对于硬卧代硬座的情况，一个一个硬卧换三张硬座来处理，那么复杂度将会主要依赖硬卧数量和硬座人数 $O\left(\min\left(b, \frac{\sum x_i}{3}\right)\right)$ 。

因此对每个订单，直接计算需要多少票，以及多少硬卧转换硬座来满足需求，才能将复杂度控制为 $O(n)$ 。

水池灌水

一道非常典型的 BFS 题，如果 BFS 来做的话，复杂度会是 $O(n * m)$ 的，于是能够轻松通过。

如果对于 BFS 并不熟悉，那么直接暴力多次扫描，至多要扫描 $n + m$ 轮，因此复杂度会提高到 $O((n + m) * n * m)$ ，在最坏情况下，最后的数据范围是可能超时的，不过本题数据随机生成，所以并没有特别卡的数据，写的好看一点应该也能通过。

挑选蛋糕

一道动态规划题。

定义 $dp_{i,t}$ 表示以第 i 个蛋糕为结尾的 t 元组的最小花费。即已经挑选了 t 个蛋糕，其中最后一个蛋糕是第 i 个蛋糕的最小花费。

那么状态转移就是：
$$dp_{i,t} = \min_{j < i \text{ \& \& } a_j < a_i} dp_{j,t-1} + b_i$$

由于只要求到三个蛋糕，所以 t 只需要循环到 3。

对于 $t=1$ 的情况，直接令 $dp_{i,1} = b_i$ 即可。

复杂度 $O(3n^2)$ 。

其他方法还有线段树/树状数组求区间最值来求解，复杂度可以达到 $O(t * n \log n)$ 要开 long long。

宝石排列

可以简单通过求排列后对每个排列计算魔法值和的方式，拿到 80 分。

考虑到摆放宝石仅与已摆放的宝石集合和最后一颗摆放的宝石有关，因此可以用状压 DP 来求解。

已摆放的宝石集合用二进制数 $state$ 表示，其中二进制位为 1 表示已取过该宝石，否则为未取过。

则 dp 状态为 $dp_{state,i}$ 表示已取过的宝石集合为 $state$ ，最后一个宝石是第 i 个宝石的最大魔法值，其中 $(state \& (1 \ll i)) \neq 0$ 。

假设对于 $dp_{state,i}$ ，在 $state$ 状态中去掉第 i 个宝石的状态 $state' = state - (1 \ll i)$ ，则状态转移为：
$$dp_{state,i} = \max_{(state' \& (1 \ll j)) \neq 0} (dp_{state',j} + a_{j,i})$$

因此，一共有 $2^n * n$ 种状态，每个状态需要遍历对应的 j ，所以总复杂度为 $O(n^2 2^n)$ 。要开 long long。