云斗 CSP-S 模拟赛题解

1 修复括号串

如何判断一个字符串是否是括号串呢，首先左右括号数量必须相同，从左到右依次枚举每个字符，左括号入栈，右括号弹栈顶，如果在栈空的适合遇到右括号就非法了。

或者将左括号视作 $+1$ ，右括号视作 -1 ，要求序列所有前缀的和不小于 0 ，且整个序列的和恰好为 0 。

做法一

枚举所有修复后的字符串，判断是否是合法括号串，计算修改的字符数量。

时间复杂度 $O(n2^n)$ ，期望得分 30 分。

做法二

从贪心的角度思考，每次会贪心的将最左侧的右括号变成左括号，最右侧的左括号变成右括号。

先操作若干次（按照贪心策略）将左右括号数量变到一致，然后二分修改多少对括号。

时间复杂度 $O(nlogn)$ ，期望得分 100 分。

做法三

在用栈模拟匹配括号串的适合，如果遇到右括号无法弹栈顶，就将其修改为左括号加入栈，最后将栈内剩余左括号的一半修改为右括号。

时间复杂度 $O(n)$ ，期望得分 100 分。

2 序列

考虑 $n \leq 5000$，可以直接记录前一个数，由于可能的数的状态和是 $\mathcal{O}(n^2)$ 的，因此可以通过。（这个状态数是怎么分析出来的：对于 a_i，他的取值由最大的 $j < i$，$a_j = a_{n-j+1}$ 决定，因此每个位置的状态都是 $\mathcal{O}(n)$ 的）。

对于答案 ≤ 2，考虑答案是 0 还是 1，如果都不是，那么答案就是 2，对于答案为 1 的情况，枚举修改哪个数即可。

正难则反，考虑最多保留几个数。

考虑对于 $1 \leq i < j \leq \lceil N/2 \rceil$，若 a_i 和 a_j 都能保留，则 $a_j - a_i \geq j - i$，即 $a_j - j \geq a_i - i$。因此将每个数减去下标之后（后一半的数减去 $n - i$）就变成类似于最长不下降子序列的问题。

从小到大考虑每个 i，对于 a_i 和 a_{n-i+1}，若二者不等则最多保留 1 个，二者独立计算即可；若二者相等，则要么保留 0 个要么保留 2 个，dp 时同时考虑。

使用树状数组/线段树可以做到 $\mathcal{O}(n \log n)$。

3 海棠花落题解

考虑实际上是为树上每一个点赋一个点权 b_i，要求相邻的点权不能一样，最小化 $\sum b_i \times a_i$

subtask 1 $n \leq 10, a_i \leq 100$

这里注意到事实上我们不会使用比 n 大的点权，这一档分数可以你直接搜索并加入一定的剪枝能够通过这一题

subtask 2 $n \leq 4 \times 10^3, a_i \leq 10^8$

这里我们考虑设置 $f_{i,j}$ 表示在树上，以 i 为根的子树，已经完成了赋值，同时给 i 这个点赋的权为 j 的最小 $\sum b_x \times a_x$ 和

我们能够写出转移式

$f_{i,j} = \sum_{v \in son_i} \min_{k \neq j} f_{v,k} + j \times a_i$

这里 son_i 指 i 的儿子集合，而对于 $\min_{k \neq j} f_{v,k}$ 你只需要记录 $f_{v,....}$ 的最小值和次小值就能一次 $O(1)$ 的转移，故而如果考虑到上一档中的 $b_i \leq n$ ，我们获得了一个 $O(n^2)$ 的算法，能够通过该档部分分

subtask 3 $n \leq 3 \times 10^5, a_i \leq 10^8$

这里我们考虑到 $b_i \leq n$ 实际上一个非常平凡的界，这里我们证明一个更好的界 $b_i \leq \log n + 1$

考虑与 i 相邻的点集 S，则我们这个点的 b_i 为 $mex\{b_j, j \in S\}$ ，即最小的没有现在 $\{b_j\}$ 的正整数. 这里可以想象，如果选择非该值，一定不如选择该值来得更优

考虑一种 $b_i = T$ 的情况，以 i 为根，则存在 i 的 $T-1$ 个儿子 $s_1, ...s_{T-1}$，满足 $b_{s_j} = j$，即 i 的所有儿子必须至少包含 $[1, T-1]$ 的所有整数

考虑设 f_T 表示存在 $b_i = T$ 的树的最小大小，根据上述分析我们知道
$f_T = 1 + \sum_{j=1}^{T-1} f_j, f_1 = 1$
解得 $f_T = 2^{T-1}$，所以 $b_i \le \log n + 1$
有了这个限制之后我们可以设计 $O(n \log n)$ 的做法，可以通过该档部分分

subtask 4 $n \le 10^6, a_i \le 10^8$ 每个点的度数小于等于 5

这一档你可以注意到每个点的 $b_i \le 5$
同时这一档分是给一些奇怪的依赖于儿子数量不会太多的做法做的

subtask 5 $n \le 10^6, a_i \le 10^8$

但是我们有完全不需要基于上界的做法。
我们思考 $\min_{k \ne j} f_{v,k}$ 这个函数，实际上的值是一个分类函数。
我们设 s，$f_{v,s} = \min f_{v,k}$
同时设 s',$f_{v,s'} = \min_{k \ne s} f_{v,s'}$
即 v 的最小值和次小值点。
当 j 为 s 时，其值为 $f_{v,s'}$
当 j 不为 s 时，其值为 $f_{v,s}$
于是我们发现实际上 $\sum_{v \in son_i} \min_{k \ne j} f_{v,k}$ 只有至多 $|son_i| + 1$ 种取值，即我们取任一子树的最小值对应的点，和不取任何一个所对应的值，这里我们称儿子的最小值的点的集合为 S

那么我们发现我们实际上并不需要维护所有点，我们只需要维护一个点的最小值和次小值位置即可，这是我们在第一部分就已考虑过的事情

那么我们接着来考虑后面这个 $j \times a_i$ 函数，其为一个递增函数

我们考虑我们取任一子树的最小值对应的点，$j \in S$ 是固定无法修改的，我们可以快速 $O(1)$ 计算得出答案

当我们不取任一子树的最小值对应的点时，由第二部分我们应取 $\text{mex}(S)$ 来作为 j 进行计算

但是这里注意到我们实际上还需要维护次小值的值，所以我们除了计算 S,$\text{mex}(S)$ 对应的点值，实际上我们还需要计算 $\text{mex}(S \cup \{\text{mex}(S)\})$ 处的值

那么我们现在对于一个点只需要计算 $O(size(son_i) + 2)$ ，我们获得了 $O(n)$ 的做法
std 中的实现使用了常数较大的做法，直接取了 $v \in S$ 计算 $v, v+1, v+2$ 处点值
容易证明 $S \cup \text{mex}(S) \cup \text{mex}(S \cup \{\text{mex}(S)\}) \in V = \{v, v+1, v+2\}(v \in S)$

4 停车场

做法一

枚举每一个空地，假设这个空地上添加了一辆车，枚举每个车，搜索的方法判断是否可以到达出口。

时间复杂度 $O(\sum(nm)^3)$ ，期望得分 20 分。

做法二

在做法一的基础上，建超级源点（想所有出口连边），搜索超级源点所在连通块，判断每个车是否在连通块内。

时间复杂度 $O(\sum(nm)^2)$ ，期望得分 30 分。

做法三

考虑将空地和空地连边建图，再建超级源点，每辆车第一步会移动到图上（最多可能 4 个不同的节点），然后走向源点，考虑哪些点是这辆车走向源点的必经点。

在边双连通分量内没有必经点，考虑建圆方树。

每辆车第一步可以移动到的节点在圆方树上对应的节点集合记作 S 。

计算 S 集合种所有点在圆方树上的 LCA ，要想到源点，必须经过 LCA ，于是这条路径上的每个圆点都是必经点。

计算每辆车的必经点的交集即是所有不能放车的位置（某个现有车辆的必经点），可以在 LCA 处打上标记，再从树上线性求出交集大小。

时间复杂度 $O(n^2logn)$ ，期望得分 100 分。