# 神灵庙(desire)

## 40 分

当树结构已确定时，肯定是深度大的节点安排小的 $a_i$。

先把 $a_i$ 排序，然后按照深度来做 dp，设 $f_{i,j,k,l}$ 为当前做到深度 $i$，深度为 $i$ 的节点有 $j$ 个，深度为 $i+1$ 的节点有 $k$ 个，已经放了 $l$ 个叶子。时间 $\mathcal{O}(n^4)$。

## 100 分

总的花费是 $\sum dep \times val$，因为我们是按照深度往下做，可以考虑把代价分到每一层，即每次深度加一是都加上还没安排的 $a_i$ 的总花费。

这样的话状态就不必带上深度了，$f_{i,j,k}$ 表示当前深度的节点有 $i$ 个，下一层的节点有 $j$ 个，叶子已经放了 $k$ 个的最小花费。时间 $\mathcal{O}(n^3)$。

注意到有 $\frac{1}{4}$ 或 $\frac{1}{8}$ 的常数，所以可以通过。 空间的话滚动数组即可。

# 断句(dealing)

## 子任务 1

AC 自动机 + dp 即可。

## 子任务 2

现在 $|S|$ 较小，考虑在 AC 自动机上的每个状态都处理出一个 $last_u$，表示从 $u$ 不停地跳 $fail$，遇到的第一个代表 $S$ 中的串的状态。dp 转移时每次跳 $last$ 即可，因为只有这些状态才对 dp 转移有用，而这些状态最多 $|S|$ 个，所以时间为 $\mathcal{O}(|S||T|)$。

## 子任务 3

如果你写了子任务 2 的做法，你会发现你直接通过了本题。这是因为子任务 2 的做法对于所有数据的时间复杂度其实都是根号的：

因为 $\sum_{s\in S}|s| \le 2\times 10^5$，所以长度大于 $\sqrt{200000}$ 的串最多 $\sqrt{200000}$ 个，从任意状态开始跳 $last$，也最多跳到根号个这样的串代表的节点；而对于长度小于等于根号的串，也只会跳到根号个，这是因为每次跳到的状态对应的串长都是不一样的，而这部分串的串长只有一到根号这根号种。

# 绀珠传（lunatic)

## 子任务 2，3

如果存在一组为交为空，这时的策略肯定是长度前 $k-1$ 大单独成组，剩下的都放一组。下面我们讨论没有交为空的组的情况：

对于一条线段 $i$，如果存在另一条被 $i$ 包含的线段 $j$，那么线段 $i$ 在最优方案中要么单独作为一组，要么和 $j$ 一个组。

所以可以把这样的 $i$ 都先去掉。现在我们剩下的线段都互不包含。

统计答案时，先枚举一个 $t$，剩下的线段组成了 $t$ 个组，则 $i$ 这样的线段有 $k-t$ 个单独成组（肯定用长度前 $k-t$ 大的）。

所以只需对于所有的 $t$ 都求出剩下的线段的贡献即可。

注意到到剩下的线段按照端点排序后，最优情况下每一组必然是一个区间，一个区间 $[x,y]$ 的贡献即为 $R_x-L_y$。

直接实现可以视情况拿到子任务 2 或 3 的分数。

## 子任务 4

当 $t=1$ 时，所有剩下的线段都是一组，贡献为 $R_1-L_m$（假设剩下 $m$ 条线段）。$t$ 增大时，可以理解为我们选择了若干个断点，每个断点 $p$ 的对贡献的增量是 $R_{p+1}-L_p$，把这个值从大到小排序选前 $t-1$ 大即可。

# 天空璋（season）

## 子任务 2

对于一个修改，使用前缀和，在矩阵中的 $\mathcal{O}(n)$ 行都修改一下。最后得到整个 $A$，再做最小生成树。

## 子任务 3

维护一个历史版本和，就不用每行都修改了。还是得到 $A$，再做最小生成树。

## 子任务 4

现在的问题是当 $n$ 较大时，不能得到完整的 $A$ 再做最小生成树。

所以考虑使用 Boruvka 算法。

难点在于每条边的边权是 $A_{i,j}+A_{j,i}$，导致扫描线怎么都不好扫。

考虑添加 $m$ 个新的操作，新加的第 $i$ 个操作为原来的第 $i$ 个操作沿主对角线反转得来，加的数也和原第 $i$ 个操作一样。这时我们惊奇地发现，新的 $A'_{i,j}=A'_{j,i}=A_{i,j}+A_{j,i}$。

现在我们就只需实现一个矩形加，每行查询颜色与 $c_i$ 不同的最大值了。

扫描线即可，颜色与 $c_i$ 不同的限制只用维护最大值和次大值就可以解决。