T2.仍旧是步履匆匆然而人群熙攘又有谁忘记了谁的好 题解

考点: 笛卡尔树/分治,树形DP

~~数据水，题目经典，思路自然，码量小。~~

送温暖。

简要题意: 使$\sum_{i=1}^{m}(m \cdot a_{b_i}) - \sum_{i=1}^{m}\sum_{j=1}^{m} f(\min(b_i, b_j), \max(b_i, b_j))$ 最大化

1.$n \le 10, m \le 10$

二进制枚举选了哪些数组成子序列，然后依照题意暴力计算。

期望得分 5pts 。

2. $n \le 22, m \le 16$

把算法一简单优化一下，预处理出有用的二进制状态，可以通过ST或者直接预处理区间最小值的方式做到每种状态 $O(m^2)$ 计算贡献。

复杂度 $O(\binom{n}{m}m^2)$

期望得分 20pts 。

3. $n \le 50, m \le 50$

~~随便设的一档分，留给可能的复杂度不够优秀的做法。~~

4. $m \le 2$

枚举一下选了哪两个数算一下贡献就行了。

结合算法2可以拿到25pts 。

5. 保证 a_i 单调

先判一下是递增还是递减。

以递增为例，那么一个区间里的数有贡献的必然是最左边的，那么对于所有的 a_{b_i} ，它贡献的系数是确定的，且和具体选了什么数无关。那么就直接设 $dp_{i,j}$ 表示当前考虑到第 i 个，选了 j 个 dp 就行了 。

不过这个部分分和正解没什么关系。

复杂度 $O(nm)$

结合算法2、算法4可以拿到35pts。

6. $n \le 300, m \le 300$

观察式子。

考虑当 $i = j$ 时,$f(\min(b_i, b_j), \max(b_i, b_j)) = a_{b_i}$,可以消掉一个 a_{b_i} 。

那么要求的实际上就是 $\sum_{i=1}^{m}((m-1) \cdot a_{b_i}) - \sum_{i=1}^{m}\sum_{j=1, j \ne i}^{m} f(\min(b_i, b_j), \max(b_i, b_j))$ 。

即 $\sum\limits_{i<j} a_{b_i} + a_{b_j} - 2 \cdot f(b_i, b_j)$ 。

将 b_i 看成点， a_{b_i} 看成权值，那么这个东西和树上距离 $val_i + val_j - 2 \cdot val_{lca(i,j)}$ 的形式是很像的。

具体的，建立笛卡尔树(也就是按值对区间分治)，找到一个区间的最小值，视其为分界分成左右两个区间，分别递归处理，构造出一个树形结构，使得 a_i 大的点在下，a_i 小的点在上。对于一对父亲 fa 和儿子 son ，令边 $fa \to son$ 的边权为 $a_{son} - a_{fa}$ ，那么每对 (b_i, b_j) 的贡献都对应了树上某一条路径的长度。那么问题就变为了: 在树上取 m 个点，使得其两两间路径长度和最大。

设 $dp_{i,j}$ 表示当前以 i 为根的子树内选了 j 个点的最大值，直接树上背包合并，枚举儿子内选了 x 个点，那么设根到儿子的边权为 w ，其贡献就是 $x \cdot (m-x) \cdot w$ 。直接树上 DP 就可以了。

如果做背包的时候枚举范围是 $[0, m]$ ，那么复杂度是 $O(nm^2)$ 。

结合算法4、5期望得分65pts 。

7. $n, m, a_i \le 1500$,且 a_i 随机生成

显然算法6的复杂度是非常不满的，又因为数据是随的，随便卡一卡就可以拿到这一档分数。

结合算法4、5期望得分70pts。

8. $n \le 4000, m \le 4000$

考虑算法6的做法。

发现对于以 x 为根的子树，其子树内选的点的个数不可能超过 $size_x$ 。那么在枚举选了多少个的时候可以直接用子树大小作为上界，考虑完一个子树后再将该大小合并到根。具体细节参考 std 。

这样卡好上界后复杂度就是 $O(nm)$ 的了。

期望得分100pts 。

对于卡上界后复杂度的简单证明:

等价于有一个 n 个点的图，每次将点集 S 与点集 T 之间的点连边，直到将所有点都合并。

每次合并中 S 与 T 无交集，没有连出重边。

所以边数最大为 $O(n^2)$ 级别，复杂度 $O(n^2)$ 级别。

证毕。