【题解】牛客 CSP-S 提高组赛前集训营 6

T1 集合统计

题意

定义一个集合 S的 f 函数为 $f(S) = \max\{a\} - \min\{a\}(a \in S)$。给定一个集合 S，求该集合所有非空子集的 f函数之和。

题解

解法一（$n \le 20$）

直接枚举所有子集判断即可。复杂度 $O(2^n \times n)$。

解法二（题解做法）

考虑每个数作为答案的贡献。

设小于 a_i的数共有 j个，那么 a_i 作为最大值的子集共有 2^j 个，对答案累加上$2^j \times a_i$ 即可。a_i作为最小值同理。

找到小于 a_i的数和大于a_i的数可以直接排序。

复杂度 $O(n\log n)$

T2 下雨天

题意

有 n个池塘，第 i个池塘容量为 i，初始时水量为 0。接下来 q 天会发生水量变化：设第 k个水池当前水量为 u，经历了某天，水量如果增加 v，那么该水池在这天过后的水量为 $\min(u+v,k)$；若 v是个负数（水量减少），那么水量为 $\max(0,u+v)$。求每天所有水池的总水量。

题解

解法一（$\mathbf{n\cdot q<10^7}$）

注意到 $n\cdot q<10\text{\^{}}7$的数据有 30 分，可以直接暴力获得。

出人意料的是第 5 个点 $n\cdot q=10^9$，测试中某些暴力也能跑过，这样暴力可以获得 35 分。

解法二（数据随机）

接下来需要稍微认真一些地分析问题。

首先注意到小的池塘的水量一定小于等于大的池塘。

然后注意到大小相邻的两个池塘水量要么相等，要么水量相差 1。

或者说，一段相邻的关系要么呈斜率为 111 的斜线，要么呈斜率为 0 的直线。

注意到至少有 60% 的数据随机，观察后发现随机情况下，斜线和直线的数目较小，可以用预先编号分段后利用线段树或者其他方法处理。

这样大概可以获得 75 分左右，会被一些精心构造的数据卡掉。

解法三（题解做法）

可以发现，相邻关系不同的数目是 $O(q)$ 的，每次最多生成两个新的，或至多删除所有旧的。

利用一个栈维护即可获得满分。

均摊时间复杂度 $O(q)$，当然也可以用 set 之类的东西维护。

想起来总觉得很难写，其实写了之后发现没有那么困难。

T3 树上队列

题意

一棵 n 个点的树，求多少种 DFS 序满足任意相邻 m个节点的导出子图连通。

求所有 $m=2,3,\cdots,\lfloor\frac{n}{2}\rfloor$都求一遍答案。

题解

解法一（n≤12）

暴搜每种合法的 DFS 序，检验是否合法。注意一下实现应该可以通过 12 以内的分数。

解法二（树退化成链）

我们发现不论 m取何值，只有从两端中的一端走到另一端合法，也就是所有的答案都是 2。因此输出 $\lfloor\frac{n}{2}\rfloor - 1$ 个 2 即可。

解法三（n≤3 000）

有一个非常重要的性质隐藏在题目里：$2m \le n$。

那么我们因此有一个结论：在开始时，队列一定踞在原树的某一棵子树里，且去掉队列所占有的这些节点后，仍然是一个连通图。

这是因为既然 $2m \le n$，就意味着队列开始时占据的 m个节点在移动过程中必须移开。而一旦这些节点将原树分成两个子树，那么当队列全部移开时，就会导致顾此失彼，这些子树没有边将它们合并起来。也就是说这样就会导致队列内元素不连通，这是一种不合法的方案。

形式化地说，就是：

令 b_u 表示 u 是否在初始队列中，即 $bu = [u \in$ 初始队列$]$，E 为边集。那么一个合法方案的初始队列就一定有且仅有一个 $(u,v) \in E$，使得 $b\_u \neq b\_v$。

根据对称性，可以发现最终队列一定也是踞在一个子树中的。

我们先介绍一个 $O(n^2)$ 的做法：考虑枚举队列所在子树的根节点 u。一个节点 u可以唯一确定一个初始时的队列（它确定了队列的大小、所占格子，但无法确定顺序，这是我们之后需要统计的）。这是因为我们假设 u 的父亲是 v，则只有 v 这一条不属于该队列。如果我们将 u 设为全树的根，那么 v 的大小就是 $n-m$。由于 $2m \leq n$，因此 $n-m$一定是 u 所有子树中唯一一个最大的。

于是我们就可以确定哪些节点是初始队列中的节点，哪些节点不是队列中的节点。

接下来我们讨论情况是否合法。

情况合法的充要条件是，在 v子树的所有节点中，儿子节点个数大于 1 的节点的子树大小不超过 m。这个是很好理解的，因为队列遇到分叉地方必须要分开，而这个子树的根节点是无法移开的（不然就把这个叉分开了），因此所有儿子节点个数大于 1（有分叉）的子树大小在最终状态中肯定是被队列占满的，否则不合法。

在合法的状态中，我们需要统计方案数。

根据乘法原理我们发现，方案数就是v子树的拓扑序数量乘上 u子树除去 v 子树后的拓扑序数量。拓扑序的数量可以通过预处理组合数，然后依次合并子树转移。

时间复杂度 $O(n^2)$。

解法四（题解做法）

~~不管你是什么方法得到的反正我是打表发现的~~一个结论：对于答案序列 $ans_2, ans_3, \cdots, ans_{\lfloor\frac{n}{2}\rfloor}$，一定存在一个划分点 k，满足对于 $ans_2, ans_3, \cdots, ans_{k-1}$ 的答案都是 0，$ans_k, ans_{k+1}, \cdots, ans_{\lfloor\frac{n}{2}\rfloor}$的答案都为一个相同的值。

这个结论的正确性是很容易理解的。根据上面的 n^2 做法我们发现，队列的路线一定是从一个子树出发，然后经过一条长长的链，最后落到一棵子树中。再根据已经提及的合法条件可知，随着队列大小的扩大，知道扩大到这两端较大的子树的大小后，就能够填满这两棵子树，开始出现合法方案，而且在这之后随着队列大小的增加，新的元素可以被挤到中间那一条链上去，而这样是不会影响总方案数的。

因此我们试图在原图上找到这样一个类似于流星锤的形状：两棵子树 u, v，满足较大的子树大小尽可能小，以及中间的一条长链，这条长链上有所有不在 u, v 子树内的点。注意这里的长链指的是边。

想到子树大小尽可能小，我们自然地想到了树的重心（PS：比赛时，选手 Freopen 提交了一个这么找重心的方法——先随便以一个点为根，然后找到树上唯一一个子树大小为 $\frac{n+1}{2}$ 的点，即为树的重心。在这道题上，这种方法是正确的。因为如果不存在这样一个点，那么答案一定全部都是 0）。我们可以找出树的重心 x，然后找到重心的最大子树 y，可以发现一旦存在答案，那么 $x \leftrightarrow y$的这条边一定是“流星锤”中间这条链的一部分。然后把 $x \leftrightarrow y$这条边不断往外延伸，就可以找到“流星锤”两端的子树节点 u, v。然后这个划分点 k 就等于

u, v 子树大小的较大值，答案就是 u子树的拓扑序数量乘上 vvv 子树的拓扑序数量的两倍。

预处理阶乘的逆元 $O(n)$，找重心 $O(n)$，动态规划求拓扑序 $O(n)$，输出答案 $O(n)$，因此总复杂度$O(n)$。

解法五（验题人做法）

对解法三的 DP 进行换根优化。注意细节。

时间复杂度也是 $O(n)$。