2023「钉耙编程」中国大学生算法设计超级联赛 (9) 题解

2023 年 8 月 15 日

A Capoo on tree

假设没有修改操作,那么我们只需要建立可持久化线段树,将权值从大到小依次插入,这样就能对于任意权值 v, 得到包含所有点权 $a_i \geq v$ 的点的线段树, 可以通过这个线段树维护区间最长连续子段的长度。我们将原树做一个树链剖分, 询问时先找到目标答案最早出现在哪段链上，然后在链对应的线段树上二分寻找合法的位置即可，时间复杂度为 $O(n\log(n))$。

对于修改操作，容易发现，将所有点权为 x 的点权值 $+1$，在可持久化线段树上所产生的变化为将 $x+1$ 版本的线段树节点替换为 x 版本的节点，即此时 $x+1$ 版本的线段树包含了原本 $x \sim n$ 的节点信息。同理，点权 -1 相当于把 x 版本的线段树节点替换为 $x+1$ 版本的节点，即此时 x 版本的线段树只能包含原本 $x+1 \sim n$ 的节点。因此对于每个修改，我们只需要做一个可持久化线段树上的区间节点替换操作，配合树剖即可完成信息变化的维护，时间复杂度为 $O(n\log^2(n))$。

B shortest path

对不起对不起，出题人和传题人和 hduoj 三者之间配合得都不是很好，导致一开始传了个样例，这里传题人背大锅了。

倒过来考虑。容易发现，连续进行超过两次减一再除二是不优的: 假设 x 是偶数，$(x-1-1)/2 = x/2-1$，而后者比前者少一次操作。同理，连续进行超过三次减一再除三也是不优的。因此，每次除二/除三操作，只会在最接近当前数的偶数/三的倍数进行。

通过上面的分析，我们可以初步得到一个简单的暴力搜索算法:

```
ll solve(ll n){
    if(n<N) return f[n];
    return min(
        solve(n/2)+n%2+1,
        solve(n/3)+n%3+1
    );
}
```

直接这样暴力地搜索显然会 TLE，但是在此基础上加上记忆化即可通过。

但是为什么这样的复杂度是正确的呢? 我们进一步分析上述操作，容易发现，对于输入 n，会搜索到的所有点均可表示为 $\lfloor n/(2^a * 3^b) \rfloor$，此时 a,b 均有最多 $\log(n)$ 种不同的取值，

因此总共会搜索到的值的数量不超过 $O(\log^2(n))$ 的，若记忆化使用的是 Hash map，则单次询问复杂度即为 $O(\log^2(n))$，若是普通 map 则为 $O(\log^3(n))$，不太能通过此题。

C Reasoning

项 (term) 可以看作一阶逻辑体系中的数值/函数，公式 (formula) 则是一阶逻辑中的所有合法公式。某个变元 x 在公式中自由出现，也就是说存在一处 x 的出现，使得没有全称量词 $\forall x$ 对其进行约束。

替换 (replacement) 是指将自由变元 x 替换成任意一个项 t。无冲突的替换 (zero-contradiction replacement) 则是指在替换时，不能出现 t 内的某变元 y 被全称量词 $\forall y$ 约束，使得其从自由变元变成约束变元的情况。

因此，判断无冲突的替换，重点就是在于全称量词开头的公式。应当判断 x 是否在公式 $\forall y\psi$ 中自由出现。如果是 x 在公式中约束出现，则一定可以无冲突替换。否则，需要判断 y 不是 t 中的自由变元，并且 ψ 内 t 也可以无冲突替换 x。

整体代码实现采用简单 dfs 即可。

D Broken Dream

回忆 Kruskal 构建 MST 的过程，本质上是图中有若干个连通块，连通块内都是一个小的 MST。加入一条边的时候就会连接两个连通块。

一条边能被加入的条件是什么呢？如果有比它还小的边存在，而且和它连接的是相同的两个连通块，则这条边肯定不能被加入。

因此就有了一个状压 dp 的思想。设 $f[i][s]$ 表示目前考虑 s 这个点集，而且只考虑前 i 小、两端都在 s 中的边，这个连通块内连出了 MST 的概率，以及设 $g[i][s]$ 表示这种情况下 MST 的期望。

将边从小到大排序，不妨设第 i 条边连接了 u, v，长度为 w。枚举 s 的子集，要求子集和子集的补集分别包含了 u, v，设子集为 L，子集的补集为 R。设所有比 w 短的边都不存在的概率为 p，那么显然有

$$f_{i,s} = f_{i-1,s} + \sum f_{i-1,L} \times f_{i-1,R} \times p$$

$$g_{i,s} = g_{i-1,s} + p \times \sum_{L} (g_{i-1,L} \times f_{i-1,R} + f_{i-1,L} \times g_{i-1,R} + w \times f_{i-1,L} \times f_{i-1,R})$$

现在问题是是怎么求 p。

设 sum_s 表示 s 由前 $i-1$ 条边构成的导出子图中所有边都不存在的概率，那么 $p = \dfrac{sum_s}{sum_L \times sum_R}$。转移之后将 sum_s 乘上第 i 条边不存在的概率即可。题目保证了 $p_i \neq 1$，所以 p 可以直接除两个分母。

时间复杂度 $\mathcal{O}(m3^n)$。

E List Reshape

先将输入列表中的全部数字提取出来。设输出列表的形状为 $x \times y$，只需按原列表顺序将每 y 个数字装入一个列表输出即可。

F Peek

对每个窗口，考虑通过叉乘求出所有产生遮挡的转折点。从窗口开始逆时针遍历凸包，观察/证明可知这些转折点产生的阴影一定是连续的。因此，可以对每个转折点求出其对应的投影点。注意，在逆时针遍历的过程中，转折点有可能先于投影点被遍历，也可能后于投影点被遍历，因此要处理两种情况。

处理完后，就可以忽略多边形本身，在多边形的边界上求只保存距离信息的阴影区间。对所有投影区间取并，就可以求出该窗口的不可见区间。

对于所有窗口的不可见区间，对它们取交，就可以求出总的不可见区间。答案即为多边形周长减去不可见区间的长度。

G Average

路径长度仅可能属于 $[k, 2*k-1]$。否则，你可以将路径拆为两段均大于 k 的路径，并选其中权值更大的那段。不考虑询问，处理每个节点为根，子树内长度小于 $2k$ 的链的最大、次大和路径，暴力合并更新答案。因为答案路径长度小于 $2k$，每次更新暴力向上更新 $2k$ 个父亲并暴力更新答案即可。时间复杂度 $\mathcal{O}(nk^2 + qk^3)$.

H Coins

答案为 $\sum_{i=1}^{n}\sum_{j=i+1}^{n} a_i a_j$.

考虑设 $T_p = \sum_{i=1}^{n}\sum_{j=i+1}^{n} a_i a_j + p$，可以证明 T_p 为鞅。设 N 为其停时，则 $T_N = N$ 为停止时的轮数。根据鞅的停时定理，则 $E[T_N] = E[N] = E[T_0] = \sum_{i=1}^{n}\sum_{j=i+1}^{n} a_i a_j$，也即期望轮数为 $\sum_{i=1}^{n}\sum_{j=i+1}^{n} a_i a_j$。

接下来只要证明 T_p 为鞅即可。

不妨假设当前时刻为 p，只剩下 n 个人，分别的钱数为 $a_1, a_2, \cdots, a_n$，则

$$
\begin{aligned}
&n(n-1)E[a_i', a_j' \mid a_i, a_j] \\
&= (n-2)(n-3)a_i a_j + (n-2)[(a_i+1)a_j + (a_i-1)a_j + a_i(a_j+1) + a_i(a_j-1)] + \\
&(a_i-1)(a_j+1) + (a_i+1)(a_j-1) \\
&= n(n-1)a_i a_j - 2
\end{aligned}
$$

于是

$$
\begin{aligned}
E[T_{p+1}] &= \sum_{i=1}^{n}\sum_{j=i+1}^{n} E[a_i' a_j'] + p + 1 \\
&= \sum_{i=1}^{n}\sum_{j=i+1}^{n} a_i a_j + p = E[T_p]
\end{aligned}
$$

于是就证明了 T_p 是鞅。

I Sequence

设计个 $2t \cdot n$ 的网格，若数列里有数 $i \cdot n + j$ 则把 (i, j) 染黑，注意到条件二等价于若一个格子染黑了，则其上下距离其奇数都不能染黑。初始 $1, 2tn$ 染黑。注意到若第一行染

了 p 个，则偶数行都最多染 $n-p$ 个，能得到两行最多染 n 个。而只有 $2t$ 行，要染 tn 个，所以 $2i-1,2i$ 两行必须染 n 个。若第一行固定了 p 个黑色，则偶数行的染色最大数量都固定为 $n-p$ 个。又第二行必须染 $n-p$ 个，则第三行染色数量 $\le p$。而第四行染色数量 $\le n-p$，所以必须同时取上界。也即若第一行确定了，则接下来所有行都确定了。

第一行的情况已知是第一个位置为黑，最后一个位置不能为黑。而条件一表明了不能有连续 d 个黑或白，于是题意转化成染长度为 n 的序列，第一个为黑，最后一个为白，不能有连续 d 个黑或白，问方案数。

注意到一定为黑白间隔，且黑白段数相同，设一共有 m 段，则题意转化为 $\sum_{i=1}^{m} a_i+b_i=n$，满足 $a_i,b_i\in[1,d-1]$，$a_1+b_m\le d-1$，问方案数。也即 $\sum_{i=1}^{2m-2} a_i+p=n$，等价于求 $[x^n]F(x)(x\frac{1-x^{d-1}}{1-x})^{2m-2}$，其中 $F(x)=\sum_{i=2}^{d-1}(i-1)x^i$，而需要枚举 m，所以答案为

$$[x^n]F(x)\sum_{m=1}^{\infty}(x\frac{1-x^{d-1}}{1-x})^{2m-2}=[x^n]F(x)\frac{1}{1-(x\frac{1-x^{d-1}}{1-x})^2}$$

分式递推即可，时间复杂度 $O(d\log d\log n)$。

J MST problem

令第 i 种权值的边有 c_i 条。首先考虑二分答案，二分答案后则对于每种权值 i，限制不能取超过 x_i 条边。不妨将问题转化为，限制每种权值的边不能删去超过 $x_i'=c_i-x_i$ 条边，并且要求删去选出来的边之后全图联通。不难发现，存在一棵生成树满足条件 (权值 i 出现次数不超过 x_i)，等价于在新的限制下，能删去的边的总数等于 $\sum x_i'$。另一方面，删除边集小于等于 x_i 和删边后联通均为拟阵，即可使用拟阵交解决问题。复杂度 $\mathcal{O}(n^3\log(W))$

K Cargo

考虑将每次选择的物品种类写成一个序列，并求出他的生成函数。对于特定的物品 i，单次被选中的概率为 $\frac{c_i}{n}$，故选择 j 件的概率为 $(\frac{c_i}{n})^j$，则其 EGF 为 $\sum_{j=0}^{\infty}\frac{(\frac{c_i}{n})^j}{j!}\times x^j=e^{\frac{c_i}{n}x}$。特殊的是，$x^{c_i}$ 这一项因为不能从恰好 c_i 个不同的商店购买，需要扣掉 $\frac{c_i!}{n^{c_i}}$ 的概率。故其 EGF 为 $\prod(e^{\frac{c_i}{n}}-\frac{c_i!}{n^{c_i}}x^{c_i})$。注意到这个式子关于 $e^{\frac{1}{n}}$ 和 x 齐次，故可以直接令 $z=\frac{e^{\frac{1}{n}}}{x}$ 作为占位量进行卷积。卷积完成后根据每一项求出 x^k 的系数即可。时间复杂度 $\mathcal{O}(n\log n)$

L Inference

题目是给出一个特征的关系图，在已知其他特征的取值的情况下，根据大数据对最后一个特征 x_m 进行估计。

朴素做法是通过 n 条数据可以统计出相关特征的某些取值共同出现的概率。采用题面内所给的公式，在图上计算概率的连乘，再比较 x_m 的各种取值哪个概率更大即可。注意，在实际计算中连乘会导致概率过小，因此应当采用取对数的方式将连乘转换为连加。

如果观察到除 x_m ，其他特征的出现次数是确定的，即可发现只需关注与 x_m 相连的特征 $\pi(x_m)$ 。如果 Alice 的 $\pi(x_m)$ 的取值在大数据中出现过，即可将对应 x_m 取值的出现次数加一。最后统计 x_m 哪个取值的出现次数最多即可。