DP 杂题选讲

HenryHuang

关于题目难度

- 应该都不算太难，希望大家补题的时候都不会自闭。
- 当然如果你觉得是部分题是大水题没有写的必要不写也没关系（

「QOJ 5571」Five Letter Warning

- 给定一个字符串 s，求其长度为 5 的回文子序列数量。
- $|s| \le 10^6$,字符集大小 ≤ 100。

「QOJ 5571」Five Letter Warning

- 思路大概有很多，关键是要避免超时和爆空间的问题。

「QOJ 5571」Five Letter Warning

- 考虑回文串的形式 $abcba$，我们考虑从这个形态入手解决问题。
- 首先容易发现 c 是不重要的，于是可以设
  - $dp_1(a)$ 表示当前序列中形如 "a" 的序列个数。
  - $dp_2(a,b)$ 表示当前序列中形如 "ab" 的序列个数。
  - $dp_3(a,b)$ 表示当前序列中形如 "$ab?$" 的序列个数。
  - $dp_4(a,b)$ 表示当前序列中形如 "$ab?b$" 的序列个数。
- 统计答案和转移是容易的。
- 时间复杂度为 $O(n|\Sigma|)$。（我也不记得是不是这个符号了，就当它是字符集大小吧 owo）

「EC Final 2022」Best Carry Player 2

- 给定 x, k，求最小的 y 使得 $x + y$ 进行竖式运算恰好进位 k 次。
- 共 T 组数据，多测。
- $1 \le T \le 10^5, 1 \le x < 10^{18}, 0 \le k \le 18$。

「EC Final 2022」Best Carry Player 2

- 考虑数位 DP。
- 设 $dp_{i,j,0/1}$ 表示当前考虑到第 i 位，进位了 j 次，当前位是否进位。
- 转移是容易的，不作赘述。

「EC Final 2022」Best Carry Player 2

- 注意有一些边界条件需要处理。
- 如最终答案可能超出 10^{18}，如 $x = 10^{17}, k = 18$。
- 注意到末尾的零不会影响进位次数，去掉即可。

「CF1715E」Long Way Home

- 给定一张 n 个点 m 条边的有边权无向图，同时你可以花费 $(u-v)^2$ 的代价从 u 点移动到 v 点进行“瞬移”，但最多只能进行 k 次。
- 求点 1 到其他点的最短路。
- $2 \le n \le 10^5, 1 \le m \le 10^5, 1 \le k \le 20$。

「CF1715E」Long Way Home

- 由于 k 很小，很容易想到分层图最短路的做法。
- 这样的复杂度为 $O\left(k(n^2+m)\log_2(n^2+m)\right)$，瓶颈在于对于特殊的移动方式，我们难以找到有效的建图方式。
- 我们尝试换一种方式进行处理。

「CF1715E」Long Way Home

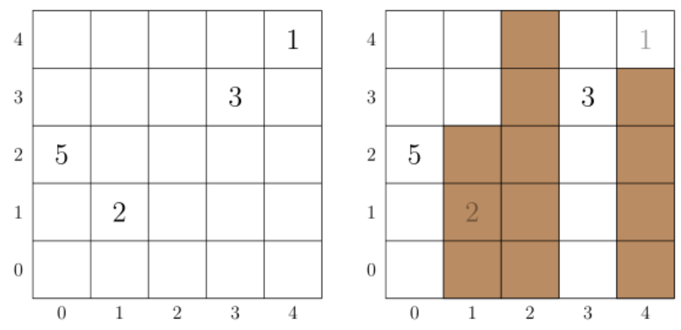
- 在层与层之间考虑 DP。
- 设 $dp_{i,u}$ 表示当前用了 i 次“瞬移”，到第 u 个点的最短路。
- 于是有

$$dp_{i,u} = \min\{dp_{i-1,v} + (u-v)^2\}$$

- 显然斜率优化即可。
- 这里 u, v 并没有大小限制，可以对两种大小关系都做一遍，当然解决方法很多。
- 复杂度为 $O(k(m\log_2 m + n))$。

「IOI2022」鲶鱼塘

- 给定 $n \times n$ 的网格单元，有 m 个单元中有鲶鱼，位置为 (x_i, y_i)，重量为 w_i。
- 对于每一列，你可以从底部建立长堤。对于第 i 条鲶鱼，其被抓住，当且仅当 $(x_i - 1, y_i)$ 或 $(x_i + 1, y_i)$ 被长堤覆盖，且 (x_i, y_i) 未被长堤覆盖。
- 求能抓住鲶鱼的最大重量。
- $2 \le n \le 10^5, 1 \le m \le 3 \times 10^5$。



「IOI2022」鲶鱼塘

- 考虑相邻的三个长堤 a,b,c 的高度:
  - 若 $a > b, c > b$，那么将 b 取为 0 一定不会使答案变劣。
  - 若 $a < b, c < b$，那么将 b 取为 n 一定不会使答案变劣。
- 这告诉我们对于一组连续的长堤，其一定是单峰的，且最高点一定为 n，最低点一定为 0。

「IOI2022」鲶鱼塘

- 设计状态为 $dp_{i,0/1}$ 表示最后一条抓到的鱼的编号为 i，当前长堤是处在上升/下降阶段。
- 容易发现长堤的修建要么为零，要么顶满，要么一定恰好修到某条鱼的下方一格，转移时按列的顺序进行转移即可。
- 时间复杂度为 $O(m\log_2 m + n)$。

「GYM104128E」Color the Tree

- 给一棵 n 个节点的有根树。
- 操作 $i(0 \le i \le n-1)$ 可以以 a_i 的代价，任意指定一个 u，将 u 子树里，所有的节点距离 u 距离恰为 i 的点染成黑色。
- 一开始所有 n 个节点都是白色的，求将所有节点染成黑色的最小代价。
- $2 \le n \le 10^5$。

「GYM104128E」Color the Tree

- 操作与深度相关，容易想到设计这样的状态：
- 设 $dp_{u,i}$ 为将以节点 u 为根的子树内所有与 u 距离为 i 的点染黑的最小代价，转移为

$$dp_{u,i} = \min(a_i, \sum dp_{v,i-1})$$

- 朴素做法的时间复杂度做法为 $O(n^2)$。

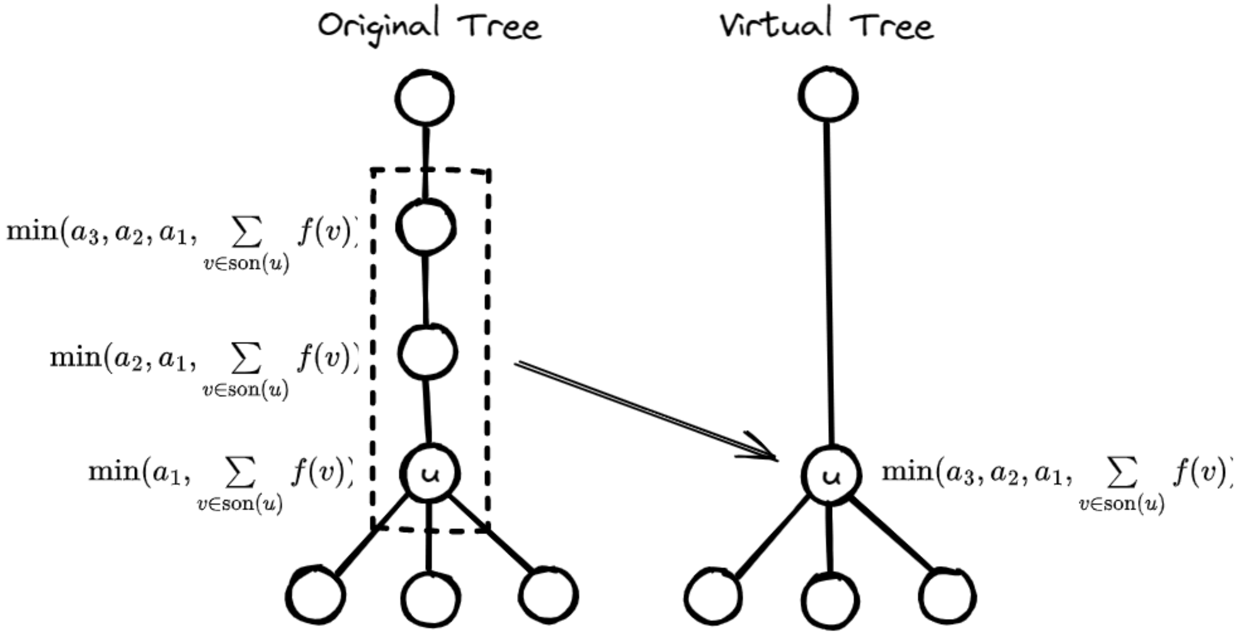
「GYM104128E」Color the Tree

- 考虑一条链的情况。
- 我们发现实际上就是求 $\min_{i=l}^{r} a_i$。

「GYM104128E」Color the Tree

- ~~当然是可以长剖的当然是可以的~~
- 官方题解的思路是对于深度为 i 的节点建立虚树，这样转移方程变为

$$dp_{u,i} = \min(\min\{a_j\}, \sum dp_{v,i-1})$$



「GYM104128E」Color the Tree

- 两种做法的复杂度都是 $O(n\log_2 n)$。

「GYM104160E」Graph Completing

- 给定一张 n 个点 m 条边的简单联通无向图，在保证其仍是简单图的前提下，可以加入任意多条（包括零条）边，使其边双联通。
- 求方案数，答案对 998244353 取模。
- $2 \le n \le 5 \times 10^3, n-1 \le m \le \min(\frac{n(n-1)}{2}, 10^4)$。

「GYM104160E」Graph Completing

- 首先进行边双联通缩点，于是我们得到了一棵树，问题转化为：加若干条边，使得这棵树边双联通。
- 我们假设每添加一条非树边 (u, v) 就会使树上节点 u 到节点 v 的路径进行覆盖，那么当所有树边都被覆盖至少一次后，就得到了一个边双联通图。
- 考虑对所有树边都被覆盖至少一次的方案进行计数。

「GYM104160E」Graph Completing

- 我们考虑容斥。问题转化为钦定一部分边不进行覆盖的方案数。
- 考虑 DP。
- 设 $dp_{u,i}$ 为考虑以 u 为根的子树，包含 u 的联通块大小为 i 的方案数。
- 转移时考虑 (u,v) 这条边：
  - 如果钦定其被覆盖，那么系数为 2^{ab-1}，其中 a,b 为状态中第二维的下标，即两个联通块除了 (u,v) 外任意连边。
  - 如果钦定其不被覆盖，那么系数为 -1，即容斥系数。
- 合理地预处理后时间复杂度为 $O(n^2)$。

「JOI 2023 Final」Cat Exercise

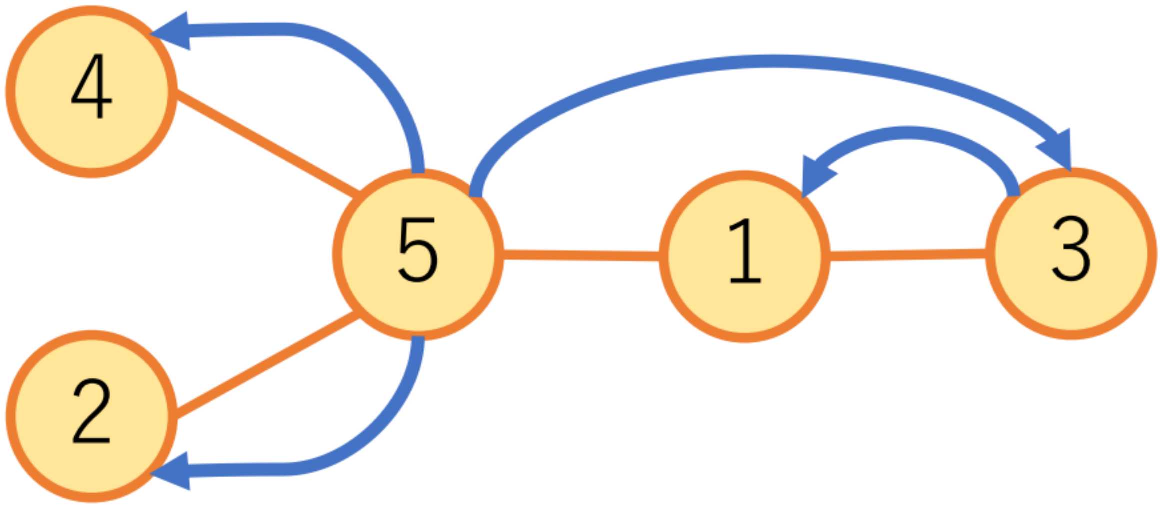
- 给定一棵 n 个节点的树，每个节点有高度 h_i，其构成 $1\sim n$ 的一个排列。起始时有一只猫呆在高度为 n 的节点上。接下来对这只猫进行锻炼，每次选择一个未放置路障的节点并在其上放置路障。
  - 如果猫不待在选定的节点中，则什么也不会发生。
  - 如果猫待在选定的节点中，并且与选定节点相邻的每个节点上都有路障，则锻炼结束。
  - 否则，猫将沿最短路径前往，仅通过移动到相邻的节点且不经过放置过路障的节点，能到达的所有节点中，高度最高的那个。
- 求以最优策略放置路障，猫走过的路径长度的最大值。
- $n \leq 2 \times 10^5$。

「JOI 2023 Final」Cat Exercise

- 注意到事实上我们不关心节点的编号，只关心节点的高度，且高度互不相同，题目已经钦定以高度为 n 的节点为根，故我们在建树时对于 (u, v)，可以直接对 (h_u, h_v) 进行连边。

「JOI 2023 Final」Cat Exercise

- 先考虑 $n \le 5000$ 的情况怎么做。
- 我们可以合理地放置障碍物，让我们能够向着某一棵子树的最高节点行走。
- 事实上我们即希望建出这样一张图：



「JOI 2023 Final」Cat Exercise

- 这样我们可以设计 dp_u 表示当前在 u 点所能走的最长路径和。
- 转移为 $dp_u = \max\{dp_v + \mathrm{dis}(u,v)\}$，其中 v 是在图上 u 的所有后继。
- 暴力建图时间复杂度为 $O(n^2)$。

「JOI 2023 Final」Cat Exercise

- 问题转换为如何更快地维护这一信息。
- 我们考虑使用并查集。
- 进行如下过程：按高度从低到高加入节点，对于节点 u，找与其相连的节点 v。
- 如果 v 所在联通块的最高点仍然比 u 的高度低，那么不存在从这个最高点向 u 转移的可能性，更新连通块状态并进行转移即可。
- 时间复杂度为 $O(n\log_2 n)$。

「GYM104197G」Graph Problem With Small $\boldsymbol{n}$

- 给定一张无向图，对于每个点对 (u, v)， 求出是否存在从 u 开始，到 v 结束的哈密顿路。
- $2 \le n \le 24$。

「GYM104197G」Graph Problem With Small n

- 一个直接的想法是设 $dp_{s,u,v}$ 表示现在存在经过了集合 s 中的点，从 u 出发，到 v 结束的路径。
- 转移是显然的。
- 这样时间复杂度较高，为 $O(2^n n^3)$。

「GYM104197G」Graph Problem With Small n

- 考虑优化。
- 我们可以将最后一维 v 保存在 dp 数组中，即 $dp_{1_{s,u}}$ 的第 v 位为 $1 \Leftrightarrow dp_{s,u,v} = 1$。
- 转移时用 and 转移即可。
- 时间复杂度为 $O(2^n n^2)$，仍然无法通过本题。

「GYM104197G」Graph Problem With Small n

- 考虑继续优化。
- 注意到刚才的状态设计依然计算了许多重复的状态，因为哈密顿路的要求是经过所有的点。
- 如果我们只求出 $dp_{1_{s,x}}$，其中 x 是一个特定点，能不能推出其他点的哈密顿路情况呢？

「GYM104197G」Graph Problem With Small n

- 考虑任意两个点 u, v，那么其有哈密顿路的条件是:
- 可以将所有点划分为两个集合 s_u, s_v，使得:
  - $u \in s_u, v \in s_v$
  - $x \in s_u, x \in s_v$
  - $u \in dp_{s_u,x}, v \in dp_{s_v,x}$
  - $s_u \cup s_v = \{1, \ldots, n\}$
- 问题转化为如何快速判断这些条件。

「GYM104197G」Graph Problem With Small n

- 注意到当 s_u 确定，那么 s_v 也随之确定了。
- 于是枚举 s_u，枚举 s_u 中的节点并利用对应的 $dp_{s_v,x}$ 对答案进行更新。
- 两部分的复杂度都为 $O(2^n n)$。

「CF1428G2」Lucky Numbers

- 对于给定的数 n，你需要将其划分为 k 个数。一个数的权值为各个数位的权值和，具体如图。
- 求将 n 划分后的最大权值和。
- 多测，k 固定，共 q 组数据。
- $1 \le q \le 10^5, 1 \le n < 10^6$。
- $1 \le k < 10^6, 1 \le F_i \le 10^9$。

	Position					
Digit	*1*	*10*	*100*	*1000*	*10000*	*100000*
3	F_0	F_1	F_2	F_3	F_4	F_5
6	$2F_0$	$2F_1$	$2F_2$	$2F_3$	$2F_4$	$2F_5$
9	$3F_0$	$3F_1$	$3F_2$	$3F_3$	$3F_4$	$3F_5$
Other	0	0	0	0	0	0

「CF1428G2」Lucky Numbers

- 我们可以证明，对于每一个数位，最多只有一个数不为 0,3,6,9。
- 反证：设有两个数位不符合条件，设为 a, b。
  - 若 $a+b \leq 9$，即可将这两个数位加起来
  - 若 $a+b>9$，则可凑成 $a+b=(a+b-9)+9$
- 以上两种方式均不会使权值和减少，进而我们证明至少存在对于每一个数位，最多只有一个数不为 0,3,6,9 的一个最优方案。

「CF1428G2」Lucky Numbers

- 每一位上都是 $0,3,6,9$ 的数一定是 3 的倍数。
- 假如给定的数是 3 的倍数，那么很简单，对于第 x 位拆成容量为 3×10^x，价值为 F_x，共有 $3k$ 个，做多重背包即可。

「CF1428G2」Lucky Numbers

- 每一位上都是 $0,3,6,9$ 的数一定是 3 的倍数。
- 假如给定的数是 3 的倍数，那么很简单，对于第 x 位拆成容量为 3×10^x，价值为 F_x，共有 $3k$ 个，做多重背包即可。
- 问题在于如何解决给定数非 3 倍数的情况。
- 我们可以把所有不为 $0,3,6,9$ 的数位交换到同一个数上，这样我们可以对剩余的 $k-1$ 个数做多重背包，然后枚举即可。

「CF1428G2」Lucky Numbers

- 还得解决多测。
- 回想下最后的过程，事实上其等价于对于每一个数位，有十种可选择的值，但对于每个数位只能选择一个值。

「CF1428G2」Lucky Numbers

- 还得解决多测。
- 回想下最后的过程，事实上其等价于对于每一个数位，有十种可选择的值，但对于每个数位只能选择一个值。
- 分组背包即可。
- 需要注意朴素的多重背包会导致超时，需要使用优化。

「CF1290F」Making Shapes

- 给定 n 个向量，满足任意两个向量不共线，现按照如下方式绘制图案：
  - 初始点为 $(0,0)$
  - 设当前点为 (x,y)，选择一个向量 $\vec{a}$，使得当前点变为 $(x+\vec{a}_x, y+\vec{a}_y)$。
  - 重复第二个操作，直到回到初始点。
- 将刚才的所有点顺次连接，可以得到一个多边形。
- 求有多少种画法使得此多边形是凸的，且最终可以被放入 $m \times m$ 且与坐标轴平行的矩形里，答案对 998244353 取模。
- $1 \le n \le 5, 1 \le m \le 10^9, |x_i|, |y_i| \le 4$。

「CF1290F」Making Shapes

- 我们发现，如果已经确定每种向量的数量，那么如果其能够组成一个凸多边形，那么其方案唯一，即将其按极角排序后顺次连接。
- 同时，我们可以形式化地转换题目条件：
  - $\sum \vec{a} = \vec{0}$
  - $\sum_{\vec{a}_{i_x}>0} c_i \vec{a}_{i_x} \le m$
  - $\sum_{\vec{a}_{i_y}>0} c_i \vec{a}_{i_y} \le m$

「CF1290F」Making Shapes

- 一个自然的思路是设状态为选了前 i 个，当前 A,B 的大小，然后在维护正负的差值信息。
- 但这样会不可避免地使状态与 m 相关，不能够解决本题。

「CF1290F」Making Shapes

- 如果每种向量只能选一次，那么由于 n 极小，也就是说如果我们暴力枚举每一种向量是否选择，复杂度是可以承受的。

「CF1290F」Making Shapes

- 将刚才的想法拓展一下。
- 我们将 c_i 进行二进制分解，对于 c_i 的每一位，我们可以暴力枚举选/不选，并对相关信息进行更新。
- 由于 $|x_i|, |y_i| \le 4$，进位带来的空间消耗也是可以承受的。

「CF1290F」Making Shapes

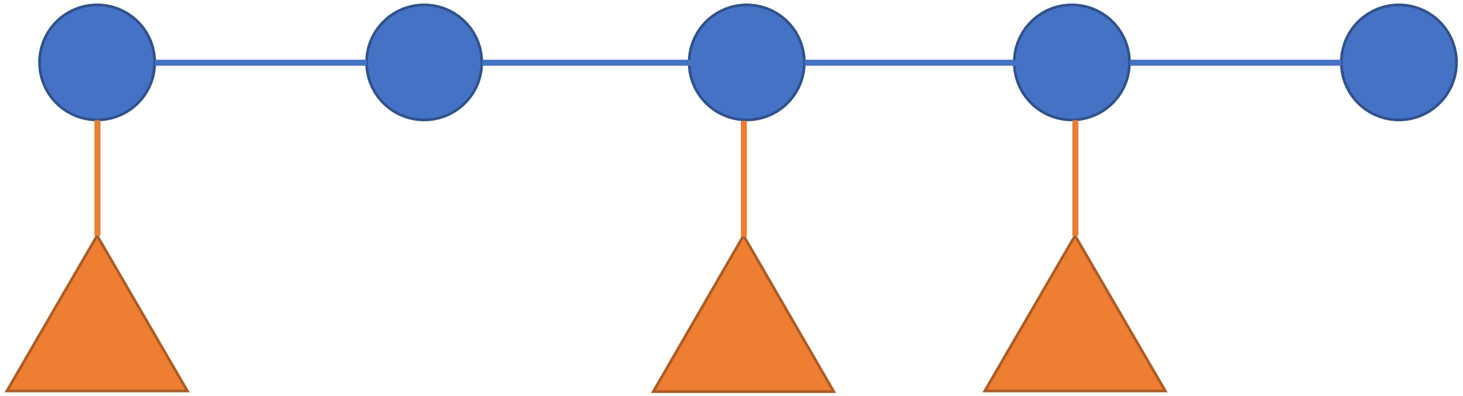
- 所以我们可以设计一个与数位 DP 极为类似的状态：
- 设 $dp_{i,a,b,c,d,0/1,0/1}$ 表示当前已经确定了最低 i 位，a,b,c,d 分别表示四种 “$\sum_{\vec{a}_{i_x}>0} c_i\vec{a}_{i_x}$” 状物的进位情况，以及当前极差的最低 i 位与 m 的最低 i 位的大小关系。
- 转移同样和数位 DP 高度类似。
- 时间复杂度为 $O(2^n(n\max\{|x_i|,|y_i|\})^4\log_2 m)$。

「GYM103470G」Paimon's Tree

- 给定一棵 $n+1$ 个结点的树，以及一个长度为 n 的序列 a。
- 初始任意选择一个结点涂黑，其余结点均为白色。然后进行如下的 n 次操作:
- 第 i 次任意选择一个与黑色结点直接相连的白色结点，将该结点涂黑，并将此边的边权设为 a_i。
- 求使用最优策略后，该树直径的最大值。
- 多测。
- $T \leq 10, n \leq 150$。

「GYM103470G」Paimon's Tree

- 我们首先考虑弱化的问题：
- 如果我们在树上钦定一条路径为我们最终所要选定的路径，如何使这条路径长度最大？
- 我们当然希望路径上的边权最大，所以我们可以把所有不需要的权值放在未被钦定的边上。

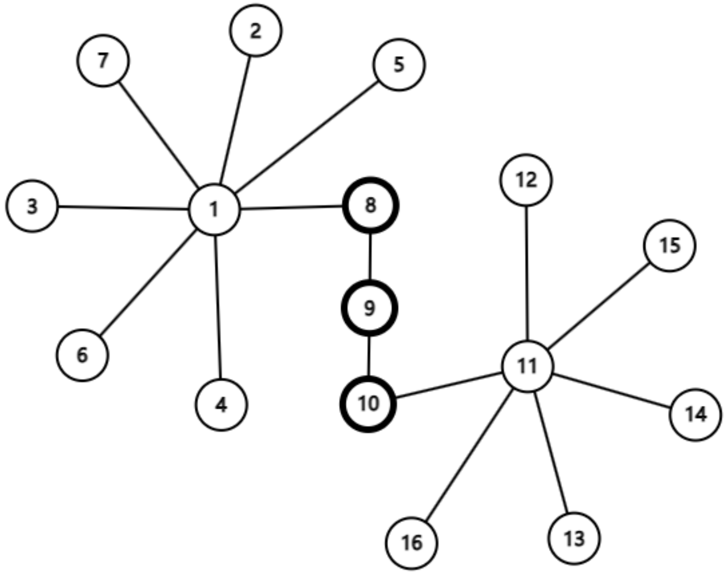


「GYM103470G」Paimon's Tree

- 由此可以设计这样一个 DP：
- 设 $dp_{l,r,t}$ 表示已经染黑了第 l 个结点到第 r 个结点，进行了 t 次操作所得到的第 l 个结点到第 r 个结点的最长路径。
- 转移是比较经典的区间 DP 套路。

「GYM103470G」Paimon's Tree

- 那么回到原问题，我们得到了一个非常朴素的做法：
- 即枚举树上的两个叶子以钦定选择的路径，然后进行我们刚才的过程。
- 最坏时间复杂度为 $O(n^5)$，大概用这样的形状能卡满。



「GYM103470G」Paimon's Tree

- 考虑不枚举，将刚才的做法拓展到树上。
- 我们发现困难之处在于端点处可供我们浪费的结点数目不确定，因为在树上我们可以向任意方向转移。
- 我们考虑把这一信息也记录进状态。

「GYM103470G」Paimon's Tree

- 设 $dp_{u,v,t,0/1,0/1}$ 表示**确定染黑** u 到 v 路径上的结点，且已染黑了除 u/v 外路径上所有的结点，共进行了 t 次操作，后两维度分别表示 u/v 是否被染黑，所得到的最长路径。
- 转移是十分类似的，时间复杂度为 $O(n^3)$。

「ZJOI2019」麻将

- 有这样一副麻将，每张牌的面值是 $1\sim n(n \leq 100)$ 中的一个整数，每种面值的牌有且仅有四张，并标有序号。
- 定义：
  - 对子：两张面值一样的牌。
  - 刻子：三张面值一样的牌。
  - 顺子，三张面值成公差为 1 的等差数列的牌。
- 一副牌能胡当且仅当：有 14 张且由 1 个对子和 4 个面子（刻子或顺子）或 7 个对子。
- 现在你手里有 13 张牌，每次你随机从牌堆里摸一张牌，求你手中的牌能产生一个能胡的子集的期望摸牌次数。

「ZJOI2019」麻将

- 不妨将原问题拆解。
- 首先考虑判定牌构成的集合是否存在胡的子集。
- 首先注意到相同的顺子数量不会超过 2，否则我们可以将其改为三个刻子，而并不影响面子的数量。
- 同时，顺子的计算和牌的大小息息相关，这启发我们按照顺序加牌，并保存当前顺子的组成状态进行 DP。

「ZJOI2019」麻将

- 不妨将原问题拆解。
- 首先考虑判定牌构成的集合是否存在胡的子集。
- 设 $dp_{w,i,j,0/1}$ 表示当前已经处理掉 $\leq w$ 的所有牌，且预留了 i 组 $(w-1,w)$，预留了 j 组 (w)，是否有雀头（对子）所得到的面子数的最大值。
- 转移即枚举和 $(w-2,w-1)$ 匹配的 w 数量以及与 $(w-1)$ 匹配的 w 的数量，是否组成对子，剩余贪心组成刻子即可。
- 额外特判七对子的情形。
- 时间复杂度为 $O(kn)$，k 是一堆常数。

「ZJOI2019」麻将

- 现在加入“加牌”的操作。
- 首先将期望转换为概率之和，即有多大概率在一共摸了 x 张牌后，仍然不存在胡的子集。
- 对于这个概率，我们就将问题转化为求在初始 13 张牌给定的情况下，不存在胡的子集的长度为 x 的麻将牌排列的数量，最后除掉分母即可。
- 排列看起来就不太好求，不如先求集合，最后再乘上阶乘。

「ZJOI2019」麻将

- 问题的关键转化为如何求这样的集合数量。
- 我们不妨在此处同样钦定从小到大加牌，这样可以使我们很方便地判断胡牌情况。
- 设 $g_{w,num,s,t}$ 表示当前已经加入了 $\le w$ 的牌，一共加了 num 张牌，dp 数组为 s，且已凑齐了 t 组对子的方案数。
- 转移是很自然的，需要注意在转移时由于牌的花色不同所需要乘的二项式系数。

「ZJOI2019」麻将

- 事实上，dp 数组存在大量的不合法情况（如已经胡了），也存在大量的状态是事实上等价的。
- 我们不能也没有必要将其全部压进状态之中。
- 可以通过搜索预先处理出所有合法的状态，再在此基础上进行转移。
- 事实上状态并不多，实测可以通过本题。
- 如果对这一步希望有更为形式化的理解，可以参照这篇博客。

撒花！

- 感谢大家的聆听。
- 本人不才，如有疏忽望您海涵。