dp

2019.6.

一些常见的dp模型

树形dp
数位dp
状压dp
dp of dp

LOJ2546

有一颗 n 个节点 $n-1$ 条边的无向树，树上节点用 $1, 2, \ldots, n$ 编号。
初始时每个节点都时白色。如果选中了节点 u，则对于树中的每条边 (u, v)，v 都会被染成黑色。注意 u 自身不会被染黑。
现在总共要选择恰好 k 个点，问有多少种方案使得所有节点被染黑。
数据范围：$1 \leq n \leq 10^5, 1 \leq k \leq \min(n, 100)$。

LOJ2546

令 $dp(u,k,color,choice)$ 表示对于以 u 为根的子树，里面选了 k 个点，u 的颜色为黑/白，u 有没有被选中。u 子树外点的选择情况之能影响到 u 子树中 u 点的颜色，u 子树中只有 u 点的选择情况能影响到 u 子树外的点的颜色。

LOJ2546

令 $dp(u, k, color, choice)$ 表示对于以 u 为根的子树，里面选了 k 个点，u 的颜色为黑/白，u 有没有被选中。u 子树外点的选择情况之能影响到 u 子树中 u 点的颜色，u 子树中只有 u 点的选择情况能影响到 u 子树外的点的颜色。
时间复杂度：$O(nk)$。

CF908G

定义 $S(n)$ 为将 n 在 10 进制下的所有数位从小到大排序后得到的数。例如：$S(1) = 1$，$S(50394) = 3459$，$S(323) = 233$。
给定 X 求 $\sum_{i=1}^{X} S(i)$ 对 $10^9 + 7$ 取模的结果。
数据范围：$1 \leq X \leq 10^{700}$。

CF908G

考虑如何计算答案，可以通过分别计算每一位对总和的贡献来求。定义 $cnt(i, x)$ 为 $S(1), S(2), ..., S(X)$ 中第 i 位为 x 的数量。

CF908G

考虑如何计算答案，可以通过分别计算每一位对总和的贡献来求。定义 $cnt(i, x)$ 为 $S(1), S(2), \dots, S(X)$ 中第 i 位为 x 的数量。
直接求 $cnt(i, x)$ 仍然较为困难，考虑差分。令 $dlt(i, x)$ 为第 i 位上有多少个数 $\geq x$。对于一个 $S(y)$ 中第 i 位 $\geq x$ 的数 y，可以发现其必须满足有至少 i 个数位 $\geq x$，这样就可以进行dp了。

CF908G

考虑如何计算答案，可以通过分别计算每一位对总和的贡献来求。定义 $cnt(i, x)$ 为 $S(1), S(2), \ldots, S(X)$ 中第 i 位为 x 的数量。
直接求 $cnt(i, x)$ 仍然较为困难，考虑差分。令 $dlt(i, x)$ 为第 i 位上有多少个数 $\geq x$。对于一个 $S(y)$ 中第 i 位 $\geq x$ 的数 y，可以发现其必须满足有至少 i 个数位 $\geq x$，这样就可以进行dp了。
另 $dp(i, j, x, cmp)$ 表示填了前 i 位，有至少 j 个数字 $\geq x$，与 N 的大小关系位 cmp。

CF908G

考虑如何计算答案，可以通过分别计算每一位对总和的贡献来求。定义 $cnt(i, x)$ 为 $S(1), S(2), \ldots, S(X)$ 中第 i 位为 x 的数量。

直接求 $cnt(i, x)$ 仍然较为困难，考虑差分。令 $dlt(i, x)$ 为第 i 位上有多少个数 $\geq x$。对于一个 $S(y)$ 中第 i 位 $\geq x$ 的数 y，可以发现其必须满足有至少 i 个数位 $\geq x$，这样就可以进行dp了。

另 $dp(i, j, x, cmp)$ 表示填了前 i 位，有至少 j 个数字 $\geq x$，与 N 的大小关系位 cmp。

转移时直接枚举第 $i+1$ 位数字是多少即可。

CF1012F

你准备去 N 个国家进行旅行，去第 i 个国家的旅行会在第 s_i 天的早上出发，第 $s_i + len_i - 1$ 天的晚上回家。
你有 P 本护照，在每次旅行前必须让其中一本护照办理该国的签证，如果在第 x 天开始对某本护照办理第 i 个国家的签证，那么第 x 天不能在旅行，且第 $x + t_i$ 天中午可完成签证拿回该护照（允许在旅行时拿到）。判断旅行计划能否完成，如果能，给出一种签证方案（时间及哪本护照）。
数据范围：$1 \le N \le 22$，$1 \le P \le 2$。

CF1012F

$P = 1$ 时怎么做？

CF1012F

$P = 1$ 时怎么做？
由于 N 的范围只有 22，不妨考虑状压dp。令 $f(S)$ 表示办完 S 集合的所有签证，拿回护照的时间最早是多少。

CF1012F

$P = 1$ 时怎么做？
由于 N 的范围只有 22，不妨考虑状压dp。令 $f(S)$ 表示办完 S 集合的所有签证，拿回护照的时间最早是多少。
转移时直接枚举下一个办签证的国家即可。

CF1012F

可以发现，如果有 2 个护照，对每个护照办签证的时间基本上是相似的，差别仅仅在于每个护照所签的国家进行旅行时该护照必须在身上。

CF1012F

可以发现，如果有 2 个护照，对每个护照办签证的时间基本上是相似的，差别仅仅在于每个护照所签的国家进行旅行时该护照必须在身上。

不妨修改 $f(S)$ 的定义，表示用一个护照办完 S 集合的所有签证，并且用这个护照旅行 S 的所有国家的拿回护照的最早时间。

CF1012F

可以发现，如果有 2 个护照，对每个护照办签证的时间基本上是相似的，差别仅仅在于每个护照所签的国家进行旅行时该护照必须在身上。
不妨修改 $f(S)$ 的定义，表示用一个护照办完 S 集合的所有签证，并且用这个护照旅行 S 的所有国家的拿回护照的最早时间。

转移时枚举下一个办签证的国家，判断是否可行并更新其它的 f 值。在判断可行时，可以按照下一个办签证国家时间长短的顺序依次枚举。可以证明这样求出的更新其它 f 的值是单调不减的，所以用指针维护即可。时间复杂度 $O(2^N\ N)$。

CF1012F

可以发现，如果有 2 个护照，对每个护照办签证的时间基本上是相似的，差别仅仅在于每个护照所签的国家进行旅行时该护照必须在身上。
不妨修改 $f(S)$ 的定义，表示用一个护照办完 S 集合的所有签证，并且用这个护照旅行 S 的所有国家的拿回护照的最早时间。

转移时枚举下一个办签证的国家，判断是否可行并更新其它的 f 值。在判断可行时，可以按照下一个办签证国家时间长短的顺序依次枚举。可以证明这样求出的更新其它 f 的值是单调不减的，所以用指针维护即可。时间复杂度 $O(2^N\ N)$。
最后求答案是只需要枚举第一个护照办理的签证即可。

一些优化dp转移的方法

状态数优化
转移数优化
斜率优化
分治优化
数据结构维护

状态数优化dp

有时候设计出的状态并不一定会满，真正有用状态可能并不多。
可以类似最简状态自动机一样对一些无用状态进行压缩合并。

状态数优化dp

有时候设计出的状态并不一定会满，真正有用状态可能并不多。
可以类似最简状态自动机一样对一些无用状态进行压缩合并。
在实现的时候往往可以采用 map 等数据结构存储出有用的状态，
这样可以高效的枚举有用的状态，避免时空超出限制。

状态数优化dp

有时候设计出的状态并不一定会满，真正有用状态可能并不多。
可以类似最简状态自动机一样对一些无用状态进行压缩合并。
在实现的时候往往可以采用 map 等数据结构存储出有用的状态，
这样可以高效的枚举有用的状态，避免时空超出限制。
插头dp也可以看成一种状态数优化的dp。

状态数优化dp

有时候设计出的状态并不一定会满，真正有用状态可能并不多。
可以类似最简状态自动机一样对一些无用状态进行压缩合并。

状态数优化dp

有时候设计出的状态并不一定会满，真正有用状态可能并不多。
可以类似最简状态自动机一样对一些无用状态进行压缩合并。
在实现的时候往往可以采用 map 等数据结构存储出有用的状态，
这样可以高效的枚举有用的状态，避免时空超出限制。

网友串

网友喜欢混乱的、毫无规律可言的字符串：网友喜欢看到长度为偶数的且前半段不等于后半段的字符串。举例来说，串 ‘0011‘ 就符合网友的审美，但是 ‘001‘, ‘0000‘ 都不符合网友的审美。

最近，在老板的要求下，网友要组织举办一个小活动，他需要给这个活动写一段标语。根据老板的要求，标语应当由 n 个单词组成，第 i 个单词 s_i 是一个长度为 a_i 的 ‘01‘ 字符串。

因为网友闲着没事，他定义这个标语中第 i 个单词是混沌的当且仅当存在 $1 \leq j < i$ 使得字符串 $s_j s_i$ 是符合网友审美的。例如在标语 ‘01‘, ‘10‘, ‘111‘ 中，只有第二个串是混沌的。

显然可能的标语一共有 $2^{\sum_{i=1}^{n} a_i}$ 种。网友现在想要对每一个 $k \in [0, n]$，计算恰好有 k 个单词是混沌的标语个数。

数据范围：$1 \leq n \leq 50, 1 \leq a_i \leq 7$。

网友串

首先，奇数和偶数可以分开处理，最后再将答案卷积得到最终答案。
预处理每一对网友数是否能够构成混沌串。
我们用三元组 (i,j,s) 来描述一个状态，表示处理了前 i 个数，出现了 j 个混沌串，并且能与集合 s 中的串组成混沌串的字符串出现过至少 1 次。
一个直观的想法是动态规划，记 $dp_{i,j,s}$ 表示到达 (i,j,s) 的方案数，利用预处理的结果，我们可以轻松地完成转移，但可能的 s 似乎很多。但实际上，可能出现的 s 少之又少，因此直接按照此方式动态规划即可。
时间复杂度 $O(N^2 * Cnt * 2^{a_i})$，其中 Cnt 为合法的 s 的个数，考虑 $a_i = 1, 3, 5, 7$ 时， $Cnt = 238$，考虑 $a_i = 2, 4, 6$ 时，Cnt=106。

转移数优化dp

与状态数优化一样，有时候许多转移也是没有用的。

CF674E

要求维护一棵树，支持以下两种操作：
1. 以某个节点为父亲，插入一个节点；
2. 询问对于以某个节点为根的子树，若子树当中每条边有 12 的概率被删除，那么整棵子树最大深度的期望值是多少。
初始时，树中仅有一个节点，共 q 次操作。对于操作2，输出实数，误差在 10^{-6} 以内。
数据范围：$1 \leq q \leq 500000$。

CF674E

令 $dp_{i,h}$ 表示对于以节点 i 为根的子树，操作之后最大深度不超过 h 的概率是多少。
对于每次询问操作，直接求 $h \cdot (dp_{u,h} - dp_{u,h-1})$ 即可。
对于每次加点操作，只会影响 $dp_{u,0}$，$dp_{par_u,1}$，$dp_{par_{par_u},2}$ $\dots$。
时间复杂度 $O(qH)$。

CF674E

令 $dp_{i,h}$ 表示对于以节点 i 为根的子树，操作之后最大深度不超过 h 的概率是多少。
对于每次询问操作，直接求 $h \cdot (dp_{u,h} - dp_{u,h-1})$ 即可。
对于每次加点操作，只会影响 $dp_{u,0}$，$dp_{par_u,1}$，$dp_{par_{par_u},2}$ $\dots$。
时间复杂度 $O(qH)$。
不难发现深度如果很大的话概率会很小，故只用对 $H \le 60$ 的 dp 就行了。

斜率优化

转移方程常常为 $dp(i) = max\{dp(j) + a(j) \cdot b(i)\}$。
将 $(a(j), dp(j))$ 看成平面上的点，则 $dp(i)$ 即为 $(b(i), 1)$ 与这些点的点积，故只用维护这些点的凸包即可。

染色游戏

给一个长度为 n 的序列，选出一个上升子序列使得 $\sum a - \sum \frac{b(b+1)}{2}$ 最大，其中 a 是被选择的所有数，b 是所有空白连续段的长度。

数据范围：$n \leq 10^6$。

染色游戏

令 f_i 表示最后一个选择的数为 i 时的最大价值，容易发现 $f_i = max_{a_j \le a_i, j < i}\ f_j + a_i - \frac{(i-j)(i-j-1)}{2}$。
没有 $a_j \le a_i$ 时就是一个简单的斜率优化问题，于是我们按照权值分治后就能套用斜率优化了。归并排序预处理每一层排序的结果，时间复杂度 $O(nlogn)$。

分治优化dp

转移式常常为 $dp_x(i) = max\{dp_{x-1}(j) + f(j,i)\}$ 如果转移有单调性，可以用分治优化。
实现函数 $work(l_1, r_1, l_2, r_2)$ 表示用 $f_x(l_1 \dots r_1)$ 更新 $f_{x+1}(l_2 \dots r_2)$，每次计算出更新 $f_{x+1}(mid_2)$ 的最优值 $f_x(mid_1)$，并递归
$work(l_1, mid_1, l_2, mid_2 - 1)$，$work(mid_1, r_1, mid_2 + 1, r_2)$。
时间复杂度为 $O(nlogn)$。

CF868 F

给序列 $a[1 \ldots n]$ 和 k，分成 k 个连续子段，使每一段中重复数对个数的和最小。
数据范围：$2 \leq n \leq 10^5$，$2 \leq k \leq \min(n, 20)$。

CF868 F

令 $f_x(i)$ 表示前 x 段覆盖了 $a[1 \dots i]$ 最小代价。转移具有单调性。

CF868 F

令 $f_x(i)$ 表示前 x 段覆盖了 $a[1 \dots i]$ 最小代价。转移具有单调性。
用分治进行优化。
计算 $f_x(i), (l_1 \le i \le r_1)$ 对 $f_{x+1}(mid_2)$ 的贡献可以类似莫队算法，记录指针 s，t 和当前 $a[s \dots t]$ 中每个数出现次数以及重复数对个数 cur，可以发现指针移动次数与递归函数中下标移动是同阶的，故复杂度为 $O(knlogn)$。

数据结构维护dp

适用于二维或者更高维的 dp。
对于每个 i，将 $dp_i(x)$ 看成整体。
有时，从 i 转移到 $i+1$ 的所有 $dp_{i+1}(x)$ 转移可以通过数据结构的方式来优化。

ARC073F

你有两个整数a和b。
现在n个操作，依次执行，每次给你 x_i，你选择两个整数中的一个y变成 x_i，代价为 $|x_i - y|$。
求做完所有操作的最小代价。
数据范围：$n, x_i, a, b \leq 200000$。

ARC073F

设 $f[i]$ 表示做完前 i 个操作，其中一个整数变成 x_i，另一个变成 x_{i-1} 的最小代价。
第一次操作枚举是哪个变成 x_1 做两次dp，以 a 为例，那么 $f[1] = |a - x_1|$，然后 $x_0 = b$，即为初值。
转移是 $f[i] = min(f[j] + sum(j+1, i-1) + |x_i - x_j - 1|)$。
按绝对值讨论拆开，然后维护两颗线段树即可。

一些dp题

CEOI2016 Kangaroo

求有多少个排列 $\{p_1, p_2, \ldots, p_n\}$ 满足 $p_1 = s$，$p_n = t$ 且对于 $1 < i < n$ 满足 $p_{i-1} < p_i$ 且 $p_{i+1} < p_i$ 或 $p_{i-1} > p_i$ 且 $p_{i+1} > p_i$。输出答案对 $10^9 + 7$ 取模。

数据范围：$1 \le s, t \le n \le 2000$，$s \ne t$。

CEOI2016 Kangaroo

令 $f(i, j, x)$ 表示前 i 个点已经填好了，构成了 j 条不包含 s, t 的链，s 和 t 所在的链状态为 x。

CEOI2016 Kangaroo

令 $f(i, j, x)$ 表示前 i 个点已经填好了，构成了 j 条不包含 s，t 的链，s 和 t 所在的链状态为 x。
转移时枚举第 $i+1$ 个点连上哪些链即可。

AGC030F

有一个长度为 $2N$ 的数列，A_1，A_2，...，A_{2N}，$A_i \in \{-1, 1, 2, ..., 2N\}$。非 -1 的数在 A 中只会出现最多一次。
现在要将所有 -1 替换成 1 到 $2N$ 中的一个数，使得 A 为排列。
定义 B_1，B_2，...，B_N，其中 $B_i = min(A_{2i-1}, A_{2i})$。求不同的 B 的个数，对 $10^9 + 7$ 取模。
数据范围：$n \leq 300$。

AGC030F

考虑一对 (A_{2i-1}, A_{2i})，如果它们都有值了就直接扔掉，剩下的只有两种：$(-1,-1)$ 和 $(-1,x)$ 。对于 $(-1,-1)$ ，我们忽略它们的位置关系，最后答案乘上其个数的阶乘。
考虑从大到小填数，记 $f(i,j,k)$ 表示当前考虑到 i ，有 j 个 -1 没有匹配，有 k 个 x 没有匹配，考虑转移：
如果 i 是某个 x ：

1. 不匹配，转移到 $f(i-1,j,k+1)$
2. 匹配一个 -1 ，转移到 $f(i-1,j-1,k)$

如果 i 不是某个 x ：

1. 不匹配，转移到 $f(i-1,j+1,k)$
2. 匹配一个 -1 ，因为是无序的所以直接转移到 $f(i-1,j-1,k)$
3. 匹配一个 x ，因为 x 的位置有关所以有 k 种方案转移到 $f(i-1,j,k-1)$

祝大家NOI取得好成绩!