dp 选讲

朱剑枫

2023.7.28

模拟试题

T1

直接按照题目模拟，设 $f_{i,j,k,0/1}$ 表示从 $(1,1)$ 走到 (i,j)，异或和为 k，且没有扔掉/扔掉一个珍珠的方案数。

模拟试题

T1

直接按照题目模拟，设 $f_{i,j,k,0/1}$ 表示从 $(1,1)$ 走到 (i,j)，异或和为 k，且没有扔掉/扔掉一个珍珠的方案数。
每次直接枚举 (i,j,k) 是向下走还是向右走就行。

模拟试题

T1

直接按照题目模拟，设 $f_{i,j,k,0/1}$ 表示从 $(1,1)$ 走到 (i,j)，异或和为 k，且没有扔掉/扔掉一个珍珠的方案数。
每次直接枚举 (i,j,k) 是向下走还是向右走就行。
时间复杂度 $O(nma)$。

考虑将一条合法的路径在 LCA 处计算进答案。可以设 $f_{i,j}$ 表示 i 子树内一条祖先-儿子链，且祖先 $>$ 儿子，最后一个点是 j 的方案数，$g_{i,j}$ 则是祖先 $<$ 儿子，最后一个点是 j 的方案数。

模拟试题

T2

考虑将一条合法的路径在 LCA 处计算进答案。可以设 $f_{i,j}$ 表示 i 子树内一条祖先-儿子链，且祖先 $>$ 儿子，最后一个点是 j 的方案数，$g_{i,j}$ 则是祖先 $<$ 儿子，最后一个点是 j 的方案数。

当 u 合并一颗子树 v 的时候，答案会增加

$\sum\limits_{a<b} f_{u,a} \times g_{v,b} + f_{u,b} \times g_{v,a}$。

时间复杂度 $O(n^2)$。

区间 dp
朋友

有 n 个房间，m 个人。第 i 个人能进入第 $[l_i, r_i]$ 个房间。若最终一个房间里有 x 个人，则这个房间会有 x 的快乐值。问最大的快乐值是多少。

$n \le 300$

区间 dp
朋友

可以证明，最优的答案存在一个房间，使得这个房间的人最多，且能到这个房间的人全部来了。

区间 dp

朋友

可以证明，最优的答案存在一个房间，使得这个房间的人最多，且能到这个房间的人全部来了。

因为对于任何一个最大值，有个人没来，那么这个人到这里来之后一定会使答案增大。

区间 dp

朋友

可以证明，最优的答案存在一个房间，使得这个房间的人最多，且能到这个房间的人全部来了。
因为对于任何一个最大值，有个人没来，那么这个人到这里来之后一定会使答案增大。
所以设 $f_{i,j}$ 表示只考虑只能到区间 $[i,j]$ 里面的人，最大的快乐值是什么。

区间 dp
朋友

可以证明，最优的答案存在一个房间，使得这个房间的人最多，且能到这个房间的人全部来了。
因为对于任何一个最大值，有个人没来，那么这个人到这里来之后一定会使答案增大。
所以设 $f_{i,j}$ 表示只考虑只能到区间 $[i,j]$ 里面的人，最大的快乐值是什么。
枚举最大值在 k，那么 $f_{i,j} = \max f_{i,k-1} + f_{k+1,j} + v^2$，其中 v 表示这里面能到 k 的人。

区间 dp
朋友

可以证明，最优的答案存在一个房间，使得这个房间的人最多，且能到这个房间的人全部来了。
因为对于任何一个最大值，有个人没来，那么这个人到这里来之后一定会使答案增大。
所以设 $f_{i,j}$ 表示只考虑只能到区间 $[i,j]$ 里面的人，最大的快乐值是什么。
枚举最大值在 k，那么 $f_{i,j} = \max f_{i,k-1} + f_{k+1,j} + v^2$，其中 v 表示这里面能到 k 的人。
那么直接求 v 不太好求，那么考虑不能到 k 的人，他们的区间必须在 $[i,k-1]$ 或 $[k+1,j]$ 里面，减掉就行。

区间 dp
朋友

可以证明，最优的答案存在一个房间，使得这个房间的人最多，且能到这个房间的人全部来了。
因为对于任何一个最大值，有个人没来，那么这个人到这里来之后一定会使答案增大。
所以设 $f_{i,j}$ 表示只考虑只能到区间 $[i,j]$ 里面的人，最大的快乐值是什么。
枚举最大值在 k，那么 $f_{i,j}=\max f_{i,k-1}+f_{k+1,j}+v^2$，其中 v 表示这里面能到 k 的人。
那么直接求 v 不太好求，那么考虑不能到 k 的人，他们的区间必须在 $[i,k-1]$ 或 $[k+1,j]$ 里面，减掉就行。
时间复杂度 $O(n^3)$。

区间 dp
AGC033D

定义一个矩阵的凌乱度是：
如果这个矩阵字符全部相等，那么是 0；
否则将这个矩阵切一刀，得到两个矩阵的凌乱度 (a, b)，对所有 $\max(a, b) + 1$ 求最小值就是这个矩阵的凌乱度。
现在给定一个 $n \times m$ 的矩阵，求它的凌乱度。
$n, m \leq 185$

区间 dp

AGC033D

最暴力的 dp：设 $f_{i,j,k,l}$ 表示左上角 (i,j)，右下角 (k,l) 的矩阵的凌乱度，那么转移直接枚举切了哪一刀就行。
时间复杂度 $O(n^5)$，优化可以做到 $O(n^4)$，很难通过。

区间 dp
AGC033D

最暴力的 dp：设 $f_{i,j,k,l}$ 表示左上角 (i,j)，右下角 (k,l) 的矩阵的凌乱度，那么转移直接枚举切了哪一刀就行。
时间复杂度 $O(n^5)$，优化可以做到 $O(n^4)$，很难通过。
但是可以注意到，$f_{i,j,k,l}$ 会很小：具体地，若区间长或宽不等于 1，则可以将它对半切开，这样的答案即是 $\log n + \log m$。

区间 dp

AGC033D

最暴力的 dp：设 $f_{i,j,k,l}$ 表示左上角 (i,j)，右下角 (k,l) 的矩阵的凌乱度，那么转移直接枚举切了哪一刀就行。
时间复杂度 $O(n^5)$，优化可以做到 $O(n^4)$，很难通过。
但是可以注意到，$f_{i,j,k,l}$ 会很小：具体地，若区间长或宽不等于 1，则可以将它对半切开，这样的答案即是 $\log n + \log m$。
所以上面 $f_{i,j,k,l}$ 的状态有很多其实是浪费（重复）的：因为它们的 dp 值一样。

区间 dp

AGC033D

而且很显然的一件事情是：i, j, k 不变，随着 l 增大，$f_{i,j,k,l}$ 不会变小，所以可以设：$g_{t,i,j,k}$ 表示最大的 l，使得 $f_{i,j,k,l} \leq t$。

区间 dp

AGC033D

而且很显然的一件事情是：i, j, k 不变，随着 l 增大，$f_{i,j,k,l}$ 不会变小，所以可以设：$g_{t,i,j,k}$ 表示最大的 l，使得 $f_{i,j,k,l} \le t$。
这样状态数就是 $O(n^3 \log n)$ 了。

区间 dp

AGC033D

而且很显然的一件事情是：i,j,k 不变，随着 l 增大，$f_{i,j,k,l}$ 不会变小，所以可以设：$g_{t,i,j,k}$ 表示最大的 l，使得 $f_{i,j,k,l} \le t$。
这样状态数就是 $O(n^3 \log n)$ 了。
考虑转移：如果是横着切的，那么肯定贪心地让上面下面都尽可能多，那么上面肯定切到 $p = g_{t-1,i,j,k}$，下面就会切到 $g_{t-1,i,p+1,k}$，那么横着切的最小凌乱度就是这个。

区间 dp
AGC033D

而且很显然的一件事情是：i, j, k 不变，随着 l 增大，$f_{i,j,k,l}$ 不会变小，所以可以设：$g_{t,i,j,k}$ 表示最大的 l，使得 $f_{i,j,k,l} \le t$。
这样状态数就是 $O(n^3 \log n)$ 了。
考虑转移：如果是横着切的，那么肯定贪心地让上面下面都尽可能多，那么上面肯定切到 $p = g_{t-1,i,j,k}$，下面就会切到 $g_{t-1,i,p+1,k}$，那么横着切的最小凌乱度就是这个。
那么如果是竖着切，则枚举切在了 x 右边，则 $g_{t,i,j,k} = \max\min(g_{t-1,i,j,x}, g_{t-1,x+1,j,k})$，这样仍然无法通过。

区间 dp

AGC033D

可以发现，随着 x 增大，$g_{t-1,i,j,x}$ 增大，$g_{t-1,x+1,j,k}$ 减小，所以我们只需要找到最大的 x，使得 $g_{t-1,i,j,x} < g_{t-1,x+1,j,k}$，然后算出切在 x 右边和 $x+1$ 右边哪个优就行。

区间 dp

AGC033D

可以发现，随着 x 增大，$g_{t-1,i,j,x}$ 增大，$g_{t-1,x+1,j,k}$ 减小，所以我们只需要找到最大的 x，使得 $g_{t-1,i,j,x} < g_{t-1,x+1,j,k}$，然后算出切在 x 右边和 $x+1$ 右边哪个优就行。
那如何找到 x 呢？仍然是利用单调性：当 i,j 不变时，若 k 变成 $k+1$，则刚刚找到的 x，$g_{t-1,i,j,x}$ 不变，$g_{t-1,x+1,j,k}$ 可能变大。所以当 k 变大时，x 也会跟着变大。所以双指针，枚举 k 的同时维护 x 即可。
时间复杂度 $O(n^3 \log n)$。

数位 dp

luoguP9387

为了省去一些不必要的知识，直接给出简化题意：
给定 n, m，设 $x = 1 \oplus 2 \oplus \cdots \oplus n-1 \oplus n \oplus m$，求有多少组 (a, b, c)，满足 $a+b+c \leq n$ 或 $a+b+c = m$，且 $(a+b+c) \oplus a \oplus c = x, b > 0$，对 10^9+7 取模。(如果 $a+b+c$ 既比 n 小也等于 m，那么需要算两次)
$n, m \leq 10^{18}$，多组数据，每组数据要求复杂度 $O(\log n)$。

数位 dp
luoguP9387

先不管 $a+b+c=m$，因为会了 $a+b+c \leq n$ 自然就能得到 $a+b+c=m$ 的方案数。

数位 dp

luoguP9387

先不管 $a+b+c=m$，因为会了 $a+b+c\le n$ 自然就能得到 $a+b+c=m$ 的方案数。
可以从低到高进行 dp，设 $f_{i,j,0/1,0/1}$ 表示从低到高 dp 完前 i 位，向第 $i+1$ 位进位的个数使 j，b 是否大于 0，$a+b+c$ 的低 i 位是否大于 n。转移暴力枚举 a,b,c 的低 i 位是什么就行，最后答案是 $f_{60,0,1,0}$。

数位 dp

luoguP9387

先不管 $a+b+c=m$，因为会了 $a+b+c\leq n$ 自然就能得到 $a+b+c=m$ 的方案数。
可以从低到高进行 dp，设 $f_{i,j,0/1,0/1}$ 表示从低到高 dp 完前 i 位，向第 $i+1$ 位进位的个数使 j，b 是否大于 0，$a+b+c$ 的低 i 位是否大于 n。转移暴力枚举 a,b,c 的低 i 位是什么就行，最后答案是 $f_{60,0,1,0}$。
单组时间复杂度 $O(\log n)$。

可以类似上面的题，但由于要算和，所以分别要记 $f_{i,j}$ 表示所有数确定完低 i 位，且低 i 位的和等于 m 的低 i 位的方案数，$g_{i,j}$ 表示所有合法方案中的异或和。

数位 dp
模拟 T3

可以类似上面的题，但由于要算和，所以分别要记 $f_{i,j}$ 表示所有数确定完低 i 位，且低 i 位的和等于 m 的低 i 位的方案数，$g_{i,j}$ 表示所有合法方案中的异或和。

转移的话，$f_{i,j}$ 直接枚举第 $i+1$ 位有几个 1，设为 x，然后只要 $x+j$ 的奇偶性和 m 第 $i+1$ 位一样就行。

数位 dp

模拟 T3

可以类似上面的题，但由于要算和，所以分别要记 $f_{i,j}$ 表示所有数确定完低 i 位，且低 i 位的和等于 m 的低 i 位的方案数，$g_{i,j}$ 表示所有合法方案中的异或和。

转移的话，$f_{i,j}$ 直接枚举第 $i+1$ 位有几个 1，设为 x，然后只要 $x+j$ 的奇偶性和 m 第 $i+1$ 位一样就行。

$g_{i,j}$ 有两个部分的贡献：一个是低于 $i+1$ 位的贡献，一个是第 $i+1$ 位的贡献。分别算即可。即 $g_{i,j} \times \binom{n}{x} + 2^{i+1} \times f_{i,j} \times \binom{n}{x}$。

数位 dp
模拟 T3

可以类似上面的题，但由于要算和，所以分别要记 $f_{i,j}$ 表示所有数确定完低 i 位，且低 i 位的和等于 m 的低 i 位的方案数，$g_{i,j}$ 表示所有合法方案中的异或和。

转移的话，$f_{i,j}$ 直接枚举第 $i+1$ 位有几个 1，设为 x，然后只要 $x+j$ 的奇偶性和 m 第 $i+1$ 位一样就行。

$g_{i,j}$ 有两个部分的贡献：一个是低于 $i+1$ 位的贡献，一个是第 $i+1$ 位的贡献。分别算即可。即 $g_{i,j} \times \binom{n}{x} + 2^{i+1} \times f_{i,j} \times \binom{n}{x}$。

时间复杂度 $O(n^2 \log m)$。

决策单调性

模拟 T4

首先，假设当我们已经选好了每个点的标准，那么每个人就一定会贪心地去他所能到达的店中标准最高的。

首先，假设当我们已经选好了每个点的标准，那么每个人就一定会贪心地去他所能到达的店中标准最高的。

那么，跟刚刚的题一样，仍然可以枚举这个区间的最大值是什么，即设 $f_{i,j}$ 表示只考虑区间 $[i,j]$ 里的人的最大收益是什么，然后枚举标准最高的点 k，设能到达 k 的人有 t 个，那么就是 $f_{i,k-1}+f_{j,k+1}+g_{k,t}$，其中 $g_{k,t}$ 表示有 t 个人到第 k 家店，合理选择标准能达到的最高收益是什么。

首先，假设当我们已经选好了每个点的标准，那么每个人就一定会贪心地去他所能到达的店中标准最高的。
那么，跟刚刚的题一样，仍然可以枚举这个区间的最大值是什么，即设 $f_{i,j}$ 表示只考虑区间 $[i,j]$ 里的人的最大收益是什么，然后枚举标准最高的点 k，设能到达 k 的人有 t 个，那么就是 $f_{i,k-1}+f_{j,k+1}+g_{k,t}$，其中 $g_{k,t}$ 表示有 t 个人到第 k 家店，合理选择标准能达到的最高收益是什么。
那么接下来就是求出所有 $g_{k,t}$。由于每个人都求这个（过程一样），于是下面用 c_i 代表标准为 i 的花费，f_j 表示有 j 个人来，那么最高收益是什么。

决策单调性

模拟 T4

显然 $f_j = \max i \times j - c_i$。这个是一个标准的斜率优化式子，建出凸包来即可。

显然 $f_j = \max i \times j - c_i$。这个是一个标准的斜率优化式子，建出凸包来即可。
但是斜率优化是有限制的，比如这些都是能放在平面直角坐标系上面的直线（也就是说贡献都可以写成 $kx + b$）的形式。

决策单调性

模拟 T4

显然 $f_j = \max i \times j - c_i$。这个是一个标准的斜率优化式子，建出凸包来即可。
但是斜率优化是有限制的，比如这些都是能放在平面直角坐标系上面的直线（也就是说贡献都可以写成 $kx + b$）的形式。
下面给一种个人感觉通用性更高，且理解起来并不比斜率优化复杂的决策单调性优化方法（虽然写起来其实本质相同）。

决策单调性

模拟 T4

对于一个 j，假设 $i_1 < i_2$，且这个时候选 i_2 比 i_1 更加优秀，那么对于所有 $k > j$，i_2 都会比 i_1 优秀，那么 i_1 在这之后就没有用了。

决策单调性

模拟 T4

对于一个 j，假设 $i_1 < i_2$，且这个时候选 i_2 比 i_1 更加优秀，那么对于所有 $k > j$，i_2 都会比 i_1 优秀，那么 i_1 在这之后就没有用了。

那么我们按 i 从小到大的顺序加入 $i \times j - c_i$ 这条直线。假设已经处理完了之前的所有直线，那么对于所有可能成为最优选择的 $i_1 < i_2 < \cdots i_k$，一定有 i_x 是 i_{x-1} 后面一些点的最优解。

决策单调性

模拟 T4

对于一个 j，假设 $i_1 < i_2$，且这个时候选 i_2 比 i_1 更加优秀，那么对于所有 $k > j$，i_2 都会比 i_1 优秀，那么 i_1 在这之后就没有用了。
那么我们按 i 从小到大的顺序加入 $i \times j - c_i$ 这条直线。假设已经处理完了之前的所有直线，那么对于所有可能成为最优选择的 $i_1 < i_2 < \cdots i_k$，一定有 i_x 是 i_{x-1} 后面一些点的最优解。
我们对每个 i_x 记录 p_x 表示对于 $j < p_x$，i_x 比 i_{x+1} 优，之后则是 i_{x+1} 优，那么显然 $p_1 < p_2 < \cdots p_{k-1}$。

决策单调性

模拟 T4

对于一个 j，假设 $i_1 < i_2$，且这个时候选 i_2 比 i_1 更加优秀，那么对于所有 $k > j$，i_2 都会比 i_1 优秀，那么 i_1 在这之后就没有用了。
那么我们按 i 从小到大的顺序加入 $i \times j - c_i$ 这条直线。假设已经处理完了之前的所有直线，那么对于所有可能成为最优选择的 $i_1 < i_2 < \cdots i_k$，一定有 i_x 是 i_{x-1} 后面一些点的最优解。
我们对每个 i_x 记录 p_x 表示对于 $j < p_x$，i_x 比 i_{x+1} 优，之后则是 i_{x+1} 优，那么显然 $p_1 < p_2 < \cdots p_{k-1}$。
否则若 $p_x > p_{x+1}$，那么在 i_{x+1} 比 i_x 优之前，i_{x+2} 就已经比 i_{x+1} 优了，所以 i_{x+1} 就不可能成为最优选择了。

那么加入 i 的时候，先算出来 i 比 i_k 什么时候优（因为 $i > i_k$，所以一定有个时间更优），设为 y，那么如果 $x_{k-1} > y$，那么根据上面 p_x 单调的解释，就需要删除 i_k，那么这样做就能维护出对于所有 j，什么 i 最优。

那么加入 i 的时候，先算出来 i 比 i_k 什么时候优（因为 $i > i_k$，所以一定有个时间更优），设为 y，那么如果 $x_{k-1} > y$，那么根据上面 p_x 单调的解释，就需要删除 i_k，那么这样做就能维护出对于所有 j，什么 i 最优。
做完这个之后直接枚举 j，然后双指针不断从前往后扫到最后一个 $p_x < j$ 的 x，i_{x+1} 就是最优的答案。

那么加入 i 的时候，先算出来 i 比 i_k 什么时候优（因为 $i > i_k$，所以一定有个时间更优），设为 y，那么如果 $x_{k-1} > y$，那么根据上面 p_x 单调的解释，就需要删除 i_k，那么这样做就能维护出对于所有 j，什么 i 最优。
做完这个之后直接枚举 j，然后双指针不断从前往后扫到最后一个 $p_x < j$ 的 x，i_{x+1} 就是最优的答案。
这样的时间复杂度是线性。于是，整个题时间复杂度 $O(n^3 + nk)$。

决策单调性

luoguP3515

给定长为 n 的序列 a，对每个 i，找到最小的非负整数 p_i，满足对于 $1 \leq j \leq n$，有 $a_j \leq a_i + p_i - \sqrt{|i-j|}$。
$n \leq 5 \times 10^5$

决策单调性

luoguP3515

忽略非负整数的限制，显然 $p_i = \max_{1 \le j \le n} a_j + \sqrt{|i-j|} - a_i$。

决策单调性

luoguP3515

忽略非负整数的限制，显然 $p_i = \max\limits_{1 \le j \le n} a_j + \sqrt{|i-j|} - a_i$。

先假设 $j < i$，之后再倒着做一遍即可。

决策单调性

luoguP3515

忽略非负整数的限制，显然 $p_i = \max_{1 \leq j \leq n} a_j + \sqrt{|i-j|} - a_i$。
先假设 $j < i$，之后再倒着做一遍即可。
这个也是有决策单调性的，即若 $j_1 < j_2$，j_2 在 i 处比 i 更大，那么 j_1 就没有用了（证明即考虑 $i - j_2 < i - j_1$，那么 $\sqrt{i - j_2}$ 增速快于 $\sqrt{i - j_1}$）。

决策单调性

luoguP3515

忽略非负整数的限制，显然 $p_i = \max\limits_{1\le j\le n} a_j + \sqrt{|i-j|} - a_i$。
先假设 $j < i$，之后再倒着做一遍即可。
这个也是有决策单调性的，即若 $j_1 < j_2$，j_2 在 i 处比 i 更大，那么 j_1 就没有用了（证明即考虑 $i - j_2 < i - j_1$，那么 $\sqrt{i-j_2}$ 增速快于 $\sqrt{i-j_1}$）。
所以也可以像刚刚那样做，算分界点的话手解一下方程即可。时间复杂度 $O(n)$。

决策单调性

luoguP3515

那么这种方法，如果 x 没有一个确定的表达式，而是若干题目所给的变量来决定的呢?

决策单调性

luoguP3515

那么这种方法，如果 x 没有一个确定的表达式，而是若干题目所给的变量来决定的呢?
比如 $f_i = \max g_j + w(j, i)$，这个 $w(j, i)$ 是要靠读入的变量来算的，那么这个时候可以通过二分找到 x，时间复杂度 $O(n \log n)$。

决策单调性

luoguP3515

那么这种方法，如果 x 没有一个确定的表达式，而是若干题目所给的变量来决定的呢？
比如 $f_i = \max g_j + w(j,i)$，这个 $w(j,i)$ 是要靠读入的变量来算的，那么这个时候可以通过二分找到 x，时间复杂度 $O(n\log n)$。
但还有另外一种简单、好写的 $O(n\log n)$ 的方法：分治。设 `solve(l, r, L, R)` 表示现在要找 f_l 到 f_r 的值，且它们的最优决策点在 L 到 R 之间。

那么这种方法，如果 x 没有一个确定的表达式，而是若干题目所给的变量来决定的呢？

比如 $f_i = \max g_j + w(j, i)$，这个 $w(j, i)$ 是要靠读入的变量来算的，那么这个时候可以通过二分找到 x，时间复杂度 $O(n \log n)$。

但还有另外一种简单、好写的 $O(n \log n)$ 的方法：分治。设 `solve(l, r, L, R)` 表示现在要找 f_l 到 f_r 的值，且它们的最优决策点在 L 到 R 之间。

每次可以找到 $mid = \frac{l+r}{2}$，然后遍历 L 到 R 并找到最优决策点 y，然后递归到 `solve(l, mid - 1, L, y)` 和 `solve(mid + 1, r, y, R)`。

决策单调性

luoguP3515

那么这种方法，如果 x 没有一个确定的表达式，而是若干题目所给的变量来决定的呢？

比如 $f_i = \max g_j + w(j, i)$，这个 $w(j, i)$ 是要靠读入的变量来算的，那么这个时候可以通过二分找到 x，时间复杂度 $O(n \log n)$。

但还有另外一种简单、好写的 $O(n \log n)$ 的方法：分治。设 `solve(l, r, L, R)` 表示现在要找 f_l 到 f_r 的值，且它们的最优决策点在 L 到 R 之间。

每次可以找到 $mid = \frac{l+r}{2}$，然后遍历 L 到 R 并找到最优决策点 y，然后递归到 `solve(l, mid - 1, L, y)` 和 `solve(mid + 1, r, y, R)`。

这样分治树总共 $\log n$ 层，且每层 L, R 的范围都是后面一个接着前面一个的，所以复杂度 $O(n \log n)$。

设 $w(l,r)$ 为 (l,r) 划分在一起的代价，那么如果对于 $a \le b \le c \le d$，有 $w(a,c)+w(b,d) \le w(b,c)+w(a,d)$，且有 $f_{i,j}=\min\{f_{i,k}+f_{k+1,j}\}+w(i,j)$，设 $g_{i,j}$ 为使 $f_{i,j}$ 最小的 k，那么有 $g_{i,j-1} \le g_{i,j} \le g_{i+1,j}$。

设 $w(l,r)$ 为 (l,r) 划分在一起的代价，那么如果对于 $a \le b \le c \le d$，有 $w(a,c)+w(b,d) \le w(b,c)+w(a,d)$，且有 $f_{i,j} = \min\{f_{i,k}+f_{k+1,j}\}+w(i,j)$，设 $g_{i,j}$ 为使 $f_{i,j}$ 最小的 k，那么有 $g_{i,j-1} \le g_{i,j} \le g_{i+1,j}$。
那么 $f_{i,j}$ 中，只需要枚举在 $g_{i,j-1}$ 到 $g_{i+1,j}$ 中的 k 就行。时间复杂度 $O(n^2)$。复杂度证明即考虑所有长为 l 的枚举决策其实遍历了长为 $l-1$ 的相邻两个最优点的区间，总长度不超过 $2n$。

设 $w(l,r)$ 为 (l,r) 划分在一起的代价，那么如果对于 $a \le b \le c \le d$，有 $w(a,c)+w(b,d) \le w(b,c)+w(a,d)$，且有 $f_{i,j}=\min\{f_{i,k}+f_{k+1,j}\}+w(i,j)$，设 $g_{i,j}$ 为使 $f_{i,j}$ 最小的 k，那么有 $g_{i,j-1} \le g_{i,j} \le g_{i+1,j}$。
那么 $f_{i,j}$ 中，只需要枚举在 $g_{i,j-1}$ 到 $g_{i+1,j}$ 中的 k 就行。时间复杂度 $O(n^2)$。复杂度证明即考虑所有长为 l 的枚举决策其实遍历了长为 $l-1$ 的相邻两个最优点的区间，总长度不超过 $2n$。
对于 $f_i=\min\{h_j+w(j,i)\}$，则 $i-1$ 最优的 j 小于等于 i 最优的 j。

设 $w(l,r)$ 为 (l,r) 划分在一起的代价，那么如果对于 $a \le b \le c \le d$，有 $w(a,c)+w(b,d) \le w(b,c)+w(a,d)$，且有 $f_{i,j}=\min\{f_{i,k}+f_{k+1,j}\}+w(i,j)$，设 $g_{i,j}$ 为使 $f_{i,j}$ 最小的 k，那么有 $g_{i,j-1} \le g_{i,j} \le g_{i+1,j}$。
那么 $f_{i,j}$ 中，只需要枚举在 $g_{i,j-1}$ 到 $g_{i+1,j}$ 中的 k 就行。时间复杂度 $O(n^2)$。复杂度证明即考虑所有长为 l 的枚举决策其实遍历了长为 $l-1$ 的相邻两个最优点的区间，总长度不超过 $2n$。
对于 $f_i=\min\{h_j+w(j,i)\}$，则 $i-1$ 最优的 j 小于等于 i 最优的 j。
但事实上，很多人会直接通过打表来直接“猜”出决策单调性。

状压 dp

luoguP1896

$n \times n$ 的棋盘上面要放置 k 个国王（国王攻击范围是周围 3×3 的格子），使得它们互相不能攻击到，问方案数。
$n \leq 9, k \leq n^2$

设 $f_{i,j,S}$ 表示 dp 完前 i 行，国王个数是 j，第 i 行填的国王方案是 S 的方案数。

状压 dp

luoguP1896

设 $f_{i,j,S}$ 表示 dp 完前 i 行，国王个数是 j，第 i 行填的国王方案是 S 的方案数。
S 实际上是取它的二进制表示，即 $0 \le S < 2^n$。比如 $n = 8$ 时，$S = (01001001)_2$ 表示第 $2, 5, 8$ 位填了国王，剩下的没有填。

状压 dp

luoguP1896

设 $f_{i,j,S}$ 表示 dp 完前 i 行，国王个数是 j，第 i 行填的国王方案是 S 的方案数。
S 实际上是取它的二进制表示，即 $0 \le S < 2^n$。比如 $n = 8$ 时，$S = (01001001)_2$ 表示第 $2, 5, 8$ 位填了国王，剩下的没有填。
转移可以枚举第 $i + 1$ 位的国王情况 T，那么 $S\&T, S\&(T << 1), S\&(T >> 1), T\&(T >> 1)$ 都需要是 0（因为国王不能相互攻击，所以相邻差距不能 ≤ 1）。

状压 dp

luoguP1896

设 $f_{i,j,S}$ 表示 dp 完前 i 行，国王个数是 j，第 i 行填的国王方案是 S 的方案数。

S 实际上是取它的二进制表示，即 $0 \le S < 2^n$。比如 $n = 8$ 时，$S = (01001001)_2$ 表示第 $2, 5, 8$ 位填了国王，剩下的没有填。

转移可以枚举第 $i + 1$ 位的国王情况 T，那么 $S\&T, S\&(T << 1), S\&(T >> 1), T\&(T >> 1)$ 都需要是 0（因为国王不能相互攻击，所以相邻差距不能 ≤ 1）。

时间复杂度 $O(n^3 2^{2n})$，但是显然是完全跑不满的。

状压 dp

luoguP6622

有 m 个信号站，最左边有一个信号塔。
有一个长为 n 的传递序列 $a_1, a_2, \cdots, a_n$，表示要按顺序进行 $n-1$ 次信号传递，第 i 次传递时，需要将信号从信号站 a_i 传送到 a_{i+1}。
当信号从 x 传递到 y 时，若 $x \le y$，则用普通传递，走一单位路程消耗一单位时间；
若 $x > y$，则用特殊传递，先把信号从 x 传递到信号塔，再从信号塔传递到 y，走一单位路程消耗 k 单位之间。
现在可以任意排列信号站，问最小传递事件。
$m \le 23, n \le 10^5$。

状压 dp

luoguP6622

题目简化一下，给定 $n-1$ 条消息传递关系 $x \to y$，要求合理安排消息站的顺序，使得时间花费最小。

状压 dp
luoguP6622

题目简化一下，给定 $n-1$ 条消息传递关系 $x \to y$，要求合理安排消息站的顺序，使得时间花费最小。
对于一条普通传递 $x \to y, x \le y$，它消耗 x 与 y 之间的距离的单位时间，这个可以通过把时间分散到 x 与 y 之间的每个空隙做到;

状压 dp
luoguP6622

题目简化一下，给定 $n-1$ 条消息传递关系 $x \to y$，要求合理安排消息站的顺序，使得时间花费最小。
对于一条普通传递 $x \to y, x \le y$，它消耗 x 与 y 之间的距离的单位时间，这个可以通过把时间分散到 x 与 y 之间的每个空隙做到;
对于一条特殊传递 $x \to y, x > y$，它消耗 $k\times$ （x 到信号塔距离加上 y 到信号塔距离，于是只要知道每个信号站有多少次特殊传递和它有关）。

状压 dp

luoguP6622

所以设 f_S 表示从左到右 dp，已经在左边的信号站的集合是 S 的最小时间花费。记 T 表示右边点的集合，即 $S + T = 2^m - 1$。

状压 dp

luoguP6622

所以设 f_S 表示从左到右 dp，已经在左边的信号站的集合是 S 的最小时间花费。记 T 表示右边点的集合，即 $S+T=2^m-1$。那么首先，f_S 要先加上有多少对信息传递 (x,y)，满足 $x \in S, y \in T$，即 S 和 T 之间的这个空隙被多少普通传递经过。

状压 dp
luoguP6622

所以设 f_S 表示从左到右 dp，已经在左边的信号站的集合是 S 的最小时间花费。记 T 表示右边点的集合，即 $S+T=2^m-1$。那么首先，f_S 要先加上有多少对信息传递 (x,y)，满足 $x\in S, y\in T$，即 S 和 T 之间的这个空隙被多少普通传递经过。这个可以设 $g_{i,S}$ 表示有多少信号传递对 $i\to j$，满足 $j\in S$，可以时空 $O(m2^m)$ 做到（即对于一条 $(x\to y)$，给 $g_{x,2^y}$ 加一，然后对于 S 不是二的次幂，直接用去掉 S 最低位的 1 加上 S 最低位的 1 就行)。

状压 dp

luoguP6622

所以设 f_S 表示从左到右 dp，已经在左边的信号站的集合是 S 的最小时间花费。记 T 表示右边点的集合，即 $S+T=2^m-1$。
那么首先，f_S 要先加上有多少对信息传递 (x,y)，满足 $x\in S, y\in T$，即 S 和 T 之间的这个空隙被多少普通传递经过。
这个可以设 $g_{i,S}$ 表示有多少信号传递对 $i\to j$，满足 $j\in S$，可以时空 $O(m2^m)$ 做到（即对于一条 $(x\to y)$，给 $g_{x,2^y}$ 加一，然后对于 S 不是二的次幂，直接用去掉 S 最低位的 1 加上 S 最低位的 1 就行)。
然后上面的就是 $\sum\limits_{x\in S} g_{x,T}$。

状压 dp

luoguP6622

然后是特殊传递，枚举下一个信号站是 w，那么跟 w 有关的特殊传递是多少对 $x \to w$，满足 $x \in T$ 和多少对 $w \to y$，满足 $y \in S$。这个和上面的 $g_{i,S}$ 是类似的。之后转移到 f_{S+w^2}。

状压 dp

luoguP6622

然后是特殊传递，枚举下一个信号站是 w，那么跟 w 有关的特殊传递是多少对 $x \to w$，满足 $x \in T$ 和多少对 $w \to y$，满足 $y \in S$。这个和上面的 $g_{i,S}$ 是类似的。之后转移到 f_{S+w^2}。
这样的时空复杂度都是 $O(m2^m)$。时间复杂度没问题，但空间复杂度爆炸了。

状压 dp

luoguP6622

然后是特殊传递，枚举下一个信号站是 w，那么跟 w 有关的特殊传递是多少对 $x \to w$，满足 $x \in T$ 和多少对 $w \to y$，满足 $y \in S$。这个和上面的 $g_{i,S}$ 是类似的。之后转移到 f_{S+w^2}。
这样的时空复杂度都是 $O(m2^m)$。时间复杂度没问题，但空间复杂度爆炸了。
空间复杂度的 $O(m2^m)$ 是在于 $g_{i,S}$。

然后是特殊传递，枚举下一个信号站是 w，那么跟 w 有关的特殊传递是多少对 $x \to w$，满足 $x \in T$ 和多少对 $w \to y$，满足 $y \in S$。这个和上面的 $g_{i,S}$ 是类似的。之后转移到 f_{S+w^2}。
这样的时空复杂度都是 $O(m2^m)$。时间复杂度没问题，但空间复杂度爆炸了。
空间复杂度的 $O(m2^m)$ 是在于 $g_{i,S}$。
但是，注意到 $g_{i,S}$ 只是对于 S 里所有的 1 求和，那么为什么没有必要把所有的 S 记录下来。也就是说，我们可以把 S 拆成两半，分为编号大于等于 $\frac{m}{2}$ 的和小于 $\frac{m}{2}$ 的。

状压 dp

luoguP6622

然后是特殊传递，枚举下一个信号站是 w，那么跟 w 有关的特殊传递是多少对 $x \to w$，满足 $x \in T$ 和多少对 $w \to y$，满足 $y \in S$。这个和上面的 $g_{i,S}$ 是类似的。之后转移到 f_{S+w^2}。
这样的时空复杂度都是 $O(m2^m)$。时间复杂度没问题，但空间复杂度爆炸了。
空间复杂度的 $O(m2^m)$ 是在于 $g_{i,S}$。
但是，注意到 $g_{i,S}$ 只是对于 S 里所有的 1 求和，那么为什么没有必要把所有的 S 记录下来。也就是说，我们可以把 S 拆成两半，分为编号大于等于 $\frac{m}{2}$ 的和小于 $\frac{m}{2}$ 的。
也就是说，设 g_{0,i,S_0} 和 g_{1,i,S_1} 分别表示编号小于 $t = \lfloor \frac{m}{2} \rfloor$ 的 2^t 种和和大于等于 t 的 2^{m-t} 种和，那么 $g_{i,S}$ 就可以快速表示为 $g_{0,i,S\&(2^t-1)} + g_{1,i,S>>t}$。$g$ 的时空复杂度为 $m2^{\frac{m}{2}}$。

状压 dp
luoguP6622

然后是特殊传递，枚举下一个信号站是 w，那么跟 w 有关的特殊传递是多少对 $x \to w$，满足 $x \in T$ 和多少对 $w \to y$，满足 $y \in S$。这个和上面的 $g_{i,S}$ 是类似的。之后转移到 f_{S+w^2}。
这样的时空复杂度都是 $O(m2^m)$。时间复杂度没问题，但空间复杂度爆炸了。
空间复杂度的 $O(m2^m)$ 是在于 $g_{i,S}$。
但是，注意到 $g_{i,S}$ 只是对于 S 里所有的 1 求和，那么为什么没有必要把所有的 S 记录下来。也就是说，我们可以把 S 拆成两半，分为编号大于等于 $\frac{m}{2}$ 的和小于 $\frac{m}{2}$ 的。
也就是说，设 g_{0,i,S_0} 和 g_{1,i,S_1} 分别表示编号小于 $t = \lfloor \frac{m}{2} \rfloor$ 的 2^t 种和和大于等于 t 的 2^{m-t} 种和，那么 $g_{i,S}$ 就可以快速表示为 $g_{0,i,S\&(2^t-1)} + g_{1,i,S>>t}$。$g$ 的时空复杂度为 $m2^{\frac{m}{2}}$。
总时间复杂度 $O(m2^m)$，空间复杂度 2^m。

状压 dp
gym104197G

给一个 n 个点的有向图，对于所有 (i, j)，判断有没有一条 i 到 j 的哈密顿路径。哈密顿路是每个点恰好只经过一次的路径。
$n \leq 24$

状压 dp

gym104197G

先假设只要对 0 作为起点，其它的点作为终点进行判断。

状压 dp

gym104197G

先假设只要对 0 作为起点，其它的点作为终点进行判断。
设 $f_{S,i}$ 表示是否能从 1 走完 S 里的点，且最后一个走到的点是 i，然后再枚举 $i \to j$ 转移到 $f_{S+2^j,j}(j \notin S)$?

状压 dp

gym104197G

先假设只要对 0 作为起点，其它的点作为终点进行判断。
设 $f_{S,i}$ 表示是否能从 1 走完 S 里的点，且最后一个走到的点是 i，然后再枚举 $i \to j$ 转移到 $f_{S+2^j,j}(j \notin S)$?
时间复杂度 $O(n^2 2^n)$，必然过不了。

状压 dp

gym104197G

先假设只要对 0 作为起点，其它的点作为终点进行判断。
设 $f_{S,i}$ 表示是否能从 1 走完 S 里的点，且最后一个走到的点是 i，然后再枚举 $i \to j$ 转移到 $f_{S+2^j,j}(j \notin S)$?
时间复杂度 $O(n^2 2^n)$，必然过不了。
注意到 $f_{S,i}$ 是一个 bool 数组，也就是说这只是一个可行性判定，那么我们也可以把答案状压，即设 g_S 表示从 1 走完 S 里的点，最后一个可以走到的点的集合是什么。那么这样状态就变成 2^n 了，但如果暴力转移，给 g_{S+2^j} 按位或上 2^j 还是不行。

状压 dp

gym104197G

注意到对于一个 S，从 i_1 走到 j 和 i_2 走到 j 是一样的，所以我们只需要知道从 g_S 里的点开始，下一步能走到哪些点。

状压 dp

gym104197G

注意到对于一个 S，从 i_1 走到 j 和 i_2 走到 j 是一样的，所以我们只需要知道从 g_S 里的点开始，下一步能走到哪些点。
设 i 能走到的点状压时候是 e_i，即 $e_i = \sum\limits_{i \to j} 2^j$，那么 g_S 能走的点就是 $\mathrm{OR}_{i \in g_s} e_i$，然后直接遍历这个结果的所有二进制位 j，给 g_{S+2^j} 按位或上 2^j 就行。

状压 dp

gym104197G

注意到对于一个 S，从 i_1 走到 j 和 i_2 走到 j 是一样的，所以我们只需要知道从 g_S 里的点开始，下一步能走到哪些点。
设 i 能走到的点状压时候是 e_i，即 $e_i = \sum\limits_{i \to j} 2^j$，那么 g_S 能走的点就是 $\mathrm{OR}_{i \in g_s} e_i$，然后直接遍历这个结果的所有二进制位 j，给 g_{S+2^j} 按位或上 2^j 就行。
时间复杂度 $O(n2^n)$，在时限内可以通过。

状压 dp

gym104197G

但是现在要对每个 i 作为起点判断，做 n 遍刚刚的会超时。

状压 dp

gym104197G

但是现在要对每个 i 作为起点判断，做 n 遍刚刚的会超时。
但是刚刚的东西又应该是必须要做的，那有什么办法尽可能利用刚刚求出来的东西呢？它不只答案 g_{2^n-1}，而是有所有 g_S。

状压 dp
gym104197G

但是现在要对每个 i 作为起点判断，做 n 遍刚刚的会超时。
但是刚刚的东西又应该是必须要做的，那有什么办法尽可能利用刚刚求出来的东西呢？它不只答案 g_{2^n-1}，而是有所有 g_S。
刚刚做的是已经把所有从 0 开始走出一个集合，这个集合所有可能的终止点求了出来。那么如果将一条哈密顿路径 $i \to j$ 拆成 $i \to 0 \to j$，且已经确定了 $0 \to j$ 中有哪些点，那么就能快速判断 $0 \to j$ 合不合法。

状压 dp

gym104197G

但是现在要对每个 i 作为起点判断，做 n 遍刚刚的会超时。
但是刚刚的东西又应该是必须要做的，那有什么办法尽可能利用刚刚求出来的东西呢？它不只答案 g_{2^n-1}，而是有所有 g_S。
刚刚做的是已经把所有从 0 开始走出一个集合，这个集合所有可能的终止点求了出来。那么如果将一条哈密顿路径 $i \to j$ 拆成 $i \to 0 \to j$，且已经确定了 $0 \to j$ 中有哪些点，那么就能快速判断 $0 \to j$ 合不合法。
那么对于 $i \to 0$，这件事只是把所有的边的方向取反，得到的 h_S，表示在原图上走过 S 里面的点，终点是 0，可能的起点集合。

状压 dp

gym104197G

那么就可以枚举 $i \to 0$ 里面的点 S，由于是哈密顿路径，$0 \to j$ 里面的点集合就是 $T = 2^n - 1 - S + 1$（加一是因为 0 号点也在里面）。

状压 dp

gym104197G

那么就可以枚举 $i \to 0$ 里面的点 S，由于是哈密顿路径，$0 \to j$ 里面的点集合就是 $T = 2^n - 1 - S + 1$（加一是因为 0 号点也在里面）。
那么对于所有 $i \in h_S, j \in g_T$，就存在一条 $i \to j$ 的哈密顿路径。

状压 dp

gym104197G

那么就可以枚举 $i \to 0$ 里面的点 S，由于是哈密顿路径，$0 \to j$ 里面的点集合就是 $T = 2^n - 1 - S + 1$（加一是因为 0 号点也在里面）。
那么对于所有 $i \in h_S, j \in g_T$，就存在一条 $i \to j$ 的哈密顿路径。
暴力枚举 i, j 仍然是 $O(n^2 2^n)$ 的。但是显然也可以将最终答案状压。

状压 dp

gym104197G

那么就可以枚举 $i \to 0$ 里面的点 S，由于是哈密顿路径，$0 \to j$ 里面的点集合就是 $T = 2^n - 1 - S + 1$（加一是因为 0 号点也在里面）。
那么对于所有 $i \in h_S, j \in g_T$，就存在一条 $i \to j$ 的哈密顿路径。
暴力枚举 i, j 仍然是 $O(n^2 2^n)$ 的。但是显然也可以将最终答案状压。
即设 ans_i 表示将所有存在 $i \to j$ 的哈密顿路径的 j 状压的结果，那么对于 $i \in S$，给 ans_i 按位或上 g_T 即可。

状压 dp

gym104197G

那么就可以枚举 $i \to 0$ 里面的点 S，由于是哈密顿路径，$0 \to j$ 里面的点集合就是 $T = 2^n - 1 - S + 1$（加一是因为 0 号点也在里面）。
那么对于所有 $i \in h_S, j \in g_T$，就存在一条 $i \to j$ 的哈密顿路径。
暴力枚举 i, j 仍然是 $O(n^2 2^n)$ 的。但是显然也可以将最终答案状压。
即设 ans_i 表示将所有存在 $i \to j$ 的哈密顿路径的 j 状压的结果，那么对于 $i \in S$，给 ans_i 按位或上 g_T 即可。
时间复杂度 $O(n2^n)$。

树上背包

luoguP4516

给定一棵 n 个点的树，求有多少种标记 k 个点的方案，使得每个点要么被标记，要么它周围的某一个点被标记。对 10^9+7 取模。
$n \le 10^5, k \le \min(n, 100)$

树上背包

luoguP4516

状态是很好设计的，即 $f_{i,j,0/1/2}$ 表示只考虑第 i 个点的子树内，标记了 $j(j \leq k)$ 个点，且 i 还不合法/周围有个点被标记但本身没被标记/本身被标记。

树上背包

luoguP4516

状态是很好设计的，即 $f_{i,j,0/1/2}$ 表示只考虑第 i 个点的子树内，标记了 $j(j \le k)$ 个点，且 i 还不合法/周围有个点被标记但本身没被标记/本身被标记。
答案就是 $f_{1,k,1} + f_{1,k,2}$。

树上背包

luoguP4516

状态是很好设计的，即 $f_{i,j,0/1/2}$ 表示只考虑第 i 个点的子树内，标记了 $j(j \le k)$ 个点，且 i 还不合法/周围有个点被标记但本身没被标记/本身被标记。

答案就是 $f_{1,k,1} + f_{1,k,2}$。

初始时有 $f_{i,0,0} = f_{i,1,2} = 1$。合并一颗子树时直接暴力枚举所有状态即可。

树上背包

luoguP4516

重点在于时间复杂度：看起来是 $O(nk^2)$ 的，但实际上是 $O(nk)$ 的。

树上背包

luoguP4516

重点在于时间复杂度：看起来是 $O(nk^2)$ 的，但实际上是 $O(nk)$ 的。
证明就是考虑合并的时候枚举 j 那一维，如果有一方 $size < k$，那么就可以看成枚举那一方子树里的所有点，然后给他们加上另一方 $size$ 的贡献，这样每个点最终贡献 $\leq k$；

树上背包

luoguP4516

重点在于时间复杂度：看起来是 $O(nk^2)$ 的，但实际上是 $O(nk)$ 的。
证明就是考虑合并的时候枚举 j 那一维，如果有一方 $size < k$，那么就可以看成枚举那一方子树里的所有点，然后给他们加上另一方 $size$ 的贡献，这样每个点最终贡献 $\le k$；
当双方 $size = k$ 时，一次的复杂度是 $O(k^2)$ 的，但这时 $size$ 加起来仍然会丢失 k，那么虽多丢失 $O(\frac{n}{k})$ 次，这部分复杂度也是 $O(nk)$。

换根 dp

Subtree

给定一棵 n 个点的树，要将一些点染成黑色，且这些黑色的点必须形成一个连通块。
对于每个 i，求出 i 被染成黑色时，有多少种合法染色方案。
$n \leq 10^5$

换根 dp

Subtree

即对每个点求有多少包含他的连通块。

换根 dp

Subtree

即对每个点求有多少包含他的连通块。
对于 i，设 f_i 表示 i 子树内有多少包含 i 的连通块，那么 $f_i = \prod_{i \to j} (1 + f_j)$。

换根 dp

Subtree

即对每个点求有多少包含他的连通块。
对于 i，设 f_i 表示 i 子树内有多少包含 i 的连通块，那么 $f_i = \prod_{i \to j} (1 + f_j)$。
那么 $ans_1 = f_1$，然后对于每个点 i，再设 g_i 表示以 i 为根时，i 以 1 为根的父亲子树内有多少包含他的连通块。

换根 dp

Subtree

即对每个点求有多少包含他的连通块。
对于 i，设 f_i 表示 i 子树内有多少包含 i 的连通块，那么 $f_i = \prod_{i \to j} (1 + f_j)$。
那么 $ans_1 = f_1$，然后对于每个点 i，再设 g_i 表示以 i 为根时，i 以 1 为根的父亲子树内有多少包含他的连通块。
那么就有 $ans_i = f_i \times g_i$。

换根 dp

Subtree

$$g_i = 1 + g_{fa_i} \times \prod_{fa_i \to j, j \neq i} (1 + f_j)。$$

换根 dp

Subtree

$g_i = 1 + g_{fa_i} \times \prod_{fa_i \to j, j \neq i} (1 + f_j)$。

这样的话第一遍 dfs 求出所有 f_i，第二遍 dfs 的时候每个点再根据上式求一下 g_i 就行。

$g_i = 1 + g_{fa_i} \times \prod_{fa_i \to j, j \neq i} (1 + f_j)$。

这样的话第一遍 dfs 求出所有 f_i，第二遍 dfs 的时候每个点再根据上式求一下 g_i 就行。

注意不能暴力求，会超时，所以先算出 $g_{fa_i} \times \prod_{fa_i \to j} (1 + f_j)$ 再除掉 $1 + f_i$ 就行。但仍然需要注意一下 $1 + f_i$ 是 0 的情况，这样别的 j 的那一部分就是 0，i 这里就可以暴力算了。

换根 dp

Subtree

$g_i = 1 + g_{fa_i} \times \prod_{fa_i \to j, j \neq i} (1 + f_j)$。
这样的话第一遍 dfs 求出所有 f_i，第二遍 dfs 的时候每个点再根据上式求一下 g_i 就行。
注意不能暴力求，会超时，所以先算出 $g_{fa_i} \times \prod_{fa_i \to j} (1 + f_j)$ 再除掉 $1 + f_i$ 就行。但仍然需要注意一下 $1 + f_i$ 是 0 的情况，这样别的 j 的那一部分就是 0，i 这里就可以暴力算了。
时间复杂度 $O(n)$。

其它

luoguP8321

有两个长为 n 的数组 a,b，现在将 a 和 b 中的元素两两随机匹配，共 $n!$ 种可能情况。
假设 a_i 匹配了 b_{p_i}，那么这种匹配的价值就是 $\prod \min(a_i, b_{p_i})$。
求随机匹配的期望价值。
$n \leq 5000$

其它

luoguP8321

不管期望价值，直接求出所有匹配的价值之和，除掉 $n!$ 就是答案。

其它

luoguP8321

不管期望价值，直接求出所有匹配的价值之和，除掉 $n!$ 就是答案。
将 a, b 放到一起，并按照从大到小的顺序排序。注意到是 $\min$ 乘起来，所以我们可以只注意每组匹配的 $\min$ 是什么。

其它

luoguP8321

不管期望价值，直接求出所有匹配的价值之和，除掉 $n!$ 就是答案。
将 a, b 放到一起，并按照从大到小的顺序排序。注意到是 $\min$ 乘起来，所以我们可以只注意每组匹配的 $\min$ 是什么。
于是设 $f_{i,j}$ 表示前 i 个数，总共匹配了 j 对的价值之和。那么转移可以是 $i+1$ 作为一组匹配的较大值，那么 j 不变。

其它

luoguP8321

不管期望价值，直接求出所有匹配的价值之和，除掉 $n!$ 就是答案。
将 a, b 放到一起，并按照从大到小的顺序排序。注意到是 $\min$ 乘起来，所以我们可以只注意每组匹配的 $\min$ 是什么。
于是设 $f_{i,j}$ 表示前 i 个数，总共匹配了 j 对的价值之和。那么转移可以是 $i+1$ 作为一组匹配的较大值，那么 j 不变。
否则 $i+1$ 就是较小值，那么就要乘上 $i+1$ 代表的数。那么还需要在前面找到跟它不在一个数组里面且没有匹配的元素，可以任意匹配，只要乘上（不在同个数组个数 $-j$）就行。j 加一。

其它

luoguP8321

不管期望价值，直接求出所有匹配的价值之和，除掉 $n!$ 就是答案。
将 a, b 放到一起，并按照从大到小的顺序排序。注意到是 $\min$ 乘起来，所以我们可以只注意每组匹配的 $\min$ 是什么。
于是设 $f_{i,j}$ 表示前 i 个数，总共匹配了 j 对的价值之和。那么转移可以是 $i+1$ 作为一组匹配的较大值，那么 j 不变。
否则 $i+1$ 就是较小值，那么就要乘上 $i+1$ 代表的数。那么还需要在前面找到跟它不在一个数组里面且没有匹配的元素，可以任意匹配，只要乘上（不在同个数组个数 $-j$）就行。j 加一。
时间复杂度 $O(n^2)$。

作业

- luoguP5851, luoguP3195, luoguP3959, luoguP2986
- CF1830D, CF1842E

以后可以尝试:

- AGC056B
- luoguP9379, luoguP3343, luoguP6803, luoguP8341