动态规划选讲

——一些动态规划的常见套路

Dew

武汉大学 ACM 集训队

ABC213,G

题意：给你一个 $n(\leq 17)$ 点 m 条边的无重边无自环的无向图，求对于每个 $k=2,\ldots,n$ ，有多少个子图保证了 1 到 k 联通，这里的子图定义为原图中删去若干条边的图。

ABC213,G

这里主要还是体现的是一类计数方法，在很多图计数中都有应用。

ABC213,G

这里主要还是体现的是一类计数方法，在很多图计数中都有应用。

首先，一个自然的想法是用 dp_s 表示点集 s 联通的方案数，这样我们枚举每个点就可以算出最终答案，但是直接转移有困难，于是我们考虑先算出点集 s 构成的图的个数，然后把所有不连通的图减去，这里令 cnt_s 表示图的个数。

ABC213,G

考虑转移，因为一个非联通图是由若干个联通图组合而成的，所以我们可以枚举第一个联通图，然后剩下的构成任意图。这里为了避免联通图在任意图中重复出现，我们有一种套路方法，即钦定某个点必定属于联通图，这样的话任意图中不可能出现我们枚举过的联通图。

ABC213,G

考虑转移，因为一个非联通图是由若干个联通图组合而成的，所以我们可以枚举第一个联通图，然后剩下的构成任意图。这里为了避免联通图在任意图中重复出现，我们有一种套路方法，即钦定某个点必定属于联通图，这样的话任意图中不可能出现我们枚举过的联通图。

转移可以写成 $dp_s = cnt_s - \sum_{p \in T \subset S} dp_T \times cnt_{S \setminus T}$

EDU115,F

题意：给 $n(\leq 20)$ 个括号序列，总长不大于 4e5，要求把这 n 个括号序列重排满足合法括号前缀个数最多，输出最多前缀个数

EDU115,F

可以考虑对某个括号序列来说，如果确定了它的前缀’(’ 减去’)’ 的数量，那么它产生的合法括号前缀个数是确定的，可以通过一些单调性的性质求出。预处理每个集合就能得到’(’ 减去’)’ 的数量，然后做一个状压就可以了，但是要注意如果某个时刻’)’ 更多，它会在这里断掉。

2020ICPC 银川,B

题意：给你一个长为 $n(n \leq 10000)$ 的整数序列，对于每个 $k = 1, 2, \ldots, n$，将序列分为 k 个连续段，使所有连续段的最大值减最小值之和最大。

2020ICPC 银川,B

这题苟就苟在当时的标算是 n^2 的，虽然后来 2021 沈阳 L 出了个数据加强版需要 polylog 但是太难了（现场无人过）这里就不说了。

2020ICPC 银川,B

这题苟就苟在当时的标算是 n^2 的，虽然后来 2021 沈阳 L 出了个数据加强版需要 polylog 但是太难了（现场无人过）这里就不说了。

这里的关键性质是对于每个连续段来说，任选两个值的差的绝对值一定比最大值减最小值更小，而我们这里需要求的是和的最大值，因此哪怕我们选出的不是最大值和最小值，答案也不会出错，只是没有用而已。

2020ICPC 银川,B

这个思想就是放宽不影响结果的条件而方便我们去 dp，Noip 2017 的宝藏也用了这个思想。

2020ICPC 银川,B

这个思想就是放宽不影响结果的条件而方便我们去 dp，Noip 2017 的宝藏也用了这个思想。

这个时候我们 dp 的时候只需要保证正贡献值和负贡献值两个两个出现就可以了，比如用 $dp_{i,j,0/1/2}$ 表示前 i 个选出 j 段还没选“最大值”和“最小值”/只选了“最大值”/只选了“最小值”的最大和，显然转移是 $O(1)$ 的

CF282,1C

题意：给一个长 $n(\leq 10^5)$ 的正整数序列 a_i，现在有 $q(\leq 5\times 10^3)$ 个要么相互覆盖要么完全不交的区间 $[l_i, r_i]$，每个区间独立有 q_i 的概率令区间加一，求最终序列最大值的期望

CF282,1C

观察到要么相互覆盖要么完全不交的区间实际上是树形结构，补充一个 $[1,n]$ 的超级根后就是一颗树，所以这个题实际上是在做树形 dp。

CF282,1C

观察到要么相互覆盖要么完全不交的区间实际上是树形结构，补充一个 $[1,n]$ 的超级根后就是一颗树，所以这个题实际上是在做树形 dp。

求期望的一个自然的想法就是求出每个最大值的概率，我们令 $dp_{i,j}$ 表示 i 号区间最大值为 $j+\max_{l_i\le p\le r_i} a_p$ 的概率，后面状态这样设计是保证了这一维的大小为 $O(q)$ 级别。

CF282,1C

观察到要么相互覆盖要么完全不交的区间实际上是树形结构，补充一个 $[1,n]$ 的超级根后就是一颗树，所以这个题实际上是在做树形 dp。

求期望的一个自然的想法就是求出每个最大值的概率，我们令 $dp_{i,j}$ 表示 i 号区间最大值为 $j+\max_{l_i\le p\le r_i} a_p$ 的概率，后面状态这样设计是保证了这一维的大小为 $O(q)$ 级别。

但是这样设计有一点不好在于我们需要钦定这个最大值到底在哪里取到的，这样会比较麻烦，因此这里采用一个套路，令 $dp_{i,j}$ 表示最大值不大于 $j+\ldots$ 的概率，就好转移很多。

[SDOI2010] 地精部落

题意：求长度 $n(\leq 5000)$ 的波动排列的数量，这里的波动定义为对所有 $1 < i < n$ 有 $a_{i-1} < a_i, a_i > a_{i+1}$ 或 $a_{i-1} > a_i, a_i < a_{i+1}$ 成立

[SDOI2010] 地精部落

关于满足特定性质的排列计数问题，我们一般关注的是相对大小。

[SDOI2010] 地精部落

关于满足特定性质的排列计数问题，我们一般关注的是相对大小。

对于这个题，我们一个自然的设计是把目前大小以及末尾是波峰还是波谷压进状态，然后考虑转移，我们需要知道把排列扩张 1 位后这一位的方案数，于是也很自然的将当前末尾的 rank 压入状态，发现就可以转移了。

[CSP-S 2021] 括号序列

题意：定义只含'(', ')' 与'*' 的串为超级括号序列当且仅当其由如下规则生成。1. "()" , "(S)" 均合法，S 表示不超过 k 个'*' 组成的非空字符串（下同）；2. 若 A, B 合法，那么 AB, ASB 均合法；3. 若 A 合法，那么 (A), (SA), (AS) 均合法。

[CSP-S 2021] 括号序列

题意：定义只含'(', ')' 与'*' 的串为超级括号序列当且仅当其由如下规则生成。1. "()" , "(S)" 均合法，S 表示不超过 k 个'*' 组成的非空字符串（下同）；2. 若 A, B 合法，那么 AB, ASB 均合法；3. 若 A 合法，那么 (A), (SA), (AS) 均合法。

现给定正整数 $n(\leq 500), k$ 以及长 n 字符集为 `{(,),?,*}` 的字符串 S，问将'?' 替换成其余三种字符后，有多少种合法的超级括号序列。

[CSP-S 2021] 括号序列

通过观察我们可以发现这题在各方面都很区间 dp，于是令 $dp_{i,j}$ 表示区间 $[i,j]$ 的超级序列方案数。规则 1 与规则 3 的添加较为类似，可以通过两端括号与极长 * 来转移（可以发现 * 不会出现在合法序列的两端）。

[CSP-S 2021] 括号序列

通过观察我们可以发现这题在各方面都很区间 dp，于是令 $dp_{i,j}$ 表示区间 $[i,j]$ 的超级序列方案数。规则 1 与规则 3 的添加较为类似，可以通过两端括号与极长 * 来转移（可以发现 * 不会出现在合法序列的两端）。

但是规则 2 则会遇到一点麻烦，因为其两端点的左右括号不一定是匹配括号，如果我们也像 1,3 那样简单的枚举中间点然后拼接，就会重复计数，例如”()()()”。

[CSP-S 2021] 括号序列

这里去掉重复有很多种方法，都需要添加额外的状态与信息，这里介绍一种。考虑到第二中情况拼接后有很多种可行中间点，我们不妨每次只枚举第一个中间点来进行拼接，于是考虑额外开一个 f 之类的表示无拼接（无中间点）情况的方案数，转移的时候就 $\sum_i f_{l,i} \sum_j dp_{r-i-j,r}[a_{i+1} \sim a_{r-i-j-1} = S]$ ，可以拿前缀和之类的优化一下得到 $O(n^3)$

CF752,1C

题意：对于某个正整数序列 B，每次可以选择某个位置 b_i 分裂成两个值 x, y 满足 $x + y = b_i$，x, y 取代原来 b_i 的位置，其余顺序不变，令 $f(B)$ 表示令序列 B 变得单调不降的最少操作次数。现在给你长 1e5 值域 1e5 的序列 A，求 $\sum_{1 \le l \le r \le n} f(\{a_l, a_{l+1}, \ldots, a_r\})$

CF752,1C

首先考虑一个序列的次数，发现可以很平凡的从右到左考虑，如果后 i 为考虑完毕且目前开头为 x，那么 a_{i-1} 一定会分成 $\lceil \frac{a_{i-1}}{x} \rceil$ 个，且若干个值的极差不超过 1 。可以发现，无论 x 是什么，分裂后的极差都不超过 1，由整除分块可以得到，新开头的值的个数是根号级别的。

CF752,1C

首先考虑一个序列的次数，发现可以很平凡的从右到左考虑，如果后 i 为考虑完毕且目前开头为 x，那么 a_{i-1} 一定会分成 $\lceil \frac{a_{i-1}}{x} \rceil$ 个，且若干个值的极差不超过 1 。可以发现，无论 x 是什么，分裂后的极差都不超过 1，由整除分块可以得到，新开头的值的个数是根号级别的。

于是令 $dp_{i,j}$ 表示以 i 为连续子序列的末尾且分裂后的值为 j（这个时候正倒无所谓）的操作次数，由上面内容可知，第二维 j 只有根号级别，转移一下即可，注意系数。

[NOIP2021] 方差

题意：给你一个长度为 $n(\leq 10^4)$ 的正整数非降序列 $A(a_i \leq 600, a_i \times n \leq 5 \times 10^5)$，现在有一种操作：选择某 $a_i(1 < i < n)$，然后令 $a_i = a_{i+1} + a_{i-1} - a_i$，问操作若干次后方差最小值的 n^2 倍是多少。

[NOIP2021] 方差

首先观察一下这个操作有什么性质，通过一定的手玩我们可以意识到，这个操作等价于交换两个相邻的差分数组，于是我们考虑差分数组。

[NOIP2021] 方差

首先观察一下这个操作有什么性质，通过一定的手玩我们可以意识到，这个操作等价于交换两个相邻的差分数组，于是我们考虑差分数组。

然后通过对方差的理解，我们可以很感性的发现这个差分数组是单谷形状时，有最小方差（因为这样更加集中）。

[NOIP2021] 方差

首先观察一下这个操作有什么性质，通过一定的手玩我们可以意识到，这个操作等价于交换两个相邻的差分数组，于是我们考虑差分数组。

然后通过对方差的理解，我们可以很感性的发现这个差分数组是单谷形状时，有最小方差（因为这样更加集中）。

通过高中数学我们知道把方差拆开后有最终答案

$ans = n^2\sigma^2 = n\sum_{i=1}^{n} a_i^2 - (\sum_{i=1}^{n} a_i)^2$，考虑到整数和不太大，我们可以压到 dp 的某一维中这样我们最小化前面那部分即可。

[NOIP2021] 方差

令 $dp_{i,j}$ 表示从小到大放置差分数组放到第 i 个后，$\sum a_i = j$ 时的最小值，转移的时候我们只需要考虑当前这个差分数组 x 放到左端还是右端即可，经过简单数学处理即可发现和与平方和的变化情况。

[NOIP2021] 方差

令 $dp_{i,j}$ 表示从小到大放置差分数组放到第 i 个后，$\sum a_i = j$ 时的最小值，转移的时候我们只需要考虑当前这个差分数组 x 放到左端还是右端即可，经过简单数学处理即可发现和与平方和的变化情况。

注意到值域很小，差分数组很多地方是 0，可以将复杂度优化到 $O(nA\min(n, A))$。

[NOIP2021] 方差

令 $dp_{i,j}$ 表示从小到大放置差分数组放到第 i 个后，$\sum a_i = j$ 时的最小值，转移的时候我们只需要考虑当前这个差分数组 x 放到左端还是右端即可，经过简单数学处理即可发现和与平方和的变化情况。

注意到值域很小，差分数组很多地方是 0，可以将复杂度优化到 $O(nA\min(n, A))$。

实际上有更优秀的做法，但是较为复杂，这里按下不表。

CF767,1D

题意：A 与 B 玩游戏，有 $n(\leq 10^6)$ 轮，A 每轮可以从 $0 \sim k$ 中选择一个实数，然后 B 选择令最终结果加上这个实数或者减去这个实数（B 知道 A 选择了什么）。最终结果初始为 0，A 的目标是最大化最终结果，B 的目标是最小化。现在 B 被要求至少选择 $m(\leq n)$ 加法（A 知道 m 的值），问最终结果对 1e9+7 取模的结果。

CF767,1D

一般来说，两个绝顶聪明的人做决策的依据是后续所有的可能状态，因此我们肯定要倒着考虑。然后我们可以发现，实际上影响决策的状态只有 B 在剩下的轮中被强制加了几次，而每轮的结果之间互相并没有影响，于是令 $dp_{i,j}$ 表示剩下 i 轮时 B 还要 j 次强制加法的结果。

CF767,1D

一般来说，两个绝顶聪明的人做决策的依据是后续所有的可能状态，因此我们肯定要倒着考虑。然后我们可以发现，实际上影响决策的状态只有 B 在剩下的轮中被强制加了几次，而每轮的结果之间互相并没有影响，于是令 $dp_{i,j}$ 表示剩下 i 轮时 B 还要 j 次强制加法的结果。

假设目前做状态 (i,j) 现在我们发现 A 决策选的数，假设为 x，B 决策要不要，即选择 $dp_{i-1,j}-x$ 或 $dp_{i-1,j-1}+x$，现在 B 会选大的，那么 A 知道这件事，这两个式子的变化又是线性的，于是 A 肯定尽可能的令这两个式子相等，会令 $x=\frac{dp_{i-1,j}-dp_{i-1,j-1}}{2}$，如果这个值不在 $[0,k]$ 范围内，肯定会取端点。

CF767,1D

不过注意到这个题是取模的，所以比较大小不太容易，我们需要再研究一下这个式子，先带入 x 取等时的情况，可以发现有 $dp_{i,j}=\frac{dp_{i-1,j}+dp_{i-1,j-1}}{2}$，这个式子很好看，合理猜测刚好 x 能够全部取等。

CF767,1D

不过注意到这个题是取模的，所以比较大小不太容易，我们需要再研究一下这个式子，先带入 x 取等时的情况，可以发现有 $dp_{i,j}=\frac{dp_{i-1,j}+dp_{i-1,j-1}}{2}$，这个式子很好看，合理猜测刚好 x 能够全部取等。考虑验证，先写出初始化，我们有全加 $dp_{i,i}=i\times k$，与全减 $dp_{i,0}=0$，然后把三角形的转移矩阵一画，发现每一步的转移中的差值的确是小于 $2k$ 的，可以随便证明一下。

CF767,1D

这个时候直接暴力求就可以过 D1 了，现在考虑 D2。

CF767,1D

这个时候直接暴力求就可以过 D1 了，现在考虑 D2。

这个转移式子长得和组合数很像，只是底下多了个 $\frac{1}{2}$ 的系数，直接分离好像比较困难，于是我们考虑独立算出每个初始位置上的贡献，也就是对角上 $dp_{i,i}$ 的贡献。我们发现，对于 $dp_{i,j}$ ，如果 $i>j$，它要么转移给 $dp_{i+1,j}$，要么转移给 $dp_{i+1,j+1}$ ，如果 $i=j$，它只会转移给 $dp_{i+1,j}$，则 i 每次都有变化，于是我们可以拿 i 来表示 $\frac{1}{2}$ 系数发生的次数，那么 $dp_{i,i}$ 到 $dp_{n,m}$ 实际上走了 $n-i$ 步，走出了 $1/2^{n-i}$ 的系数。

CF767,1D

现在考虑 j，可以发现 $n-i$ 步中需要选择 $m-i$ 步走 $j+1$，且第一步不能走 $j+1$，实际上就是一个路径方案数，为 $\binom{n-i-1}{m-i}$。

CF767,1D

现在考虑 j，可以发现 $n-i$ 步中需要选择 $m-i$ 步走 $j+1$，且第一步不能走 $j+1$，实际上就是一个路径方案数，为 $\binom{n-i-1}{m-i}$。

于是我们有最终结果 $dp_{n,m}=\sum_i ik\binom{n-i-1}{m-i}2^{i-n}$

CF750,F

题意：给一个长 $n = 10^6$ 的序列 A，非负整数值域 $V = 5000$，问 A 递增子序列的异或和的所有结果并输出

CF750,F

一般而言，我们会想到将当前末尾与异或和压入状态，但是可以发现这样转移是 $O(V)$ 的，好像不太好优化。

CF750,F

一般而言，我们会想到将当前末尾与异或和压入状态，但是可以发现这样转移是 $O(V)$ 的，好像不太好优化。

这里就有这样一个套路，令 $dp_{i,j}$ 表示构成末尾值为 j 且异或和为 i 的在原序列的最早位置，这样我们从左到右遍历到 x 时，会对 $dp_{k,x}$ 从 $\min_{j<x} dp_{k\oplus x,j}$ 更新，显然可以 $O(\log n)$ 转移，但是 $O(1)$ 好像还是有点难度。

CF750,F

但是我们发现，直接取 min 好像还是体现不了最早位置的性质，因此考虑换一种枚举顺序进行 dp，按 a_i 的大小顺序，这样的话第二维 j 就没必要枚举，我们枚举到 $x = a_i$ 时，直接考虑是否有 $dp_k < i$，如果有就看看能不能修改 $dp_{k \oplus x}$ 更小。

CF750,F

但是我们发现，直接取 min 好像还是体现不了最早位置的性质，因此考虑换一种枚举顺序进行 dp，按 a_i 的大小顺序，这样的话第二维 j 就没必要枚举，我们枚举到 $x = a_i$ 时，直接考虑是否有 $dp_k < i$，如果有就看看能不能修改 $dp_{k \oplus x}$ 更小。

虽然这样转移是 $O(1)$ 了，但是序列长度过大，考虑到值域够小，我们可以一次把所有相同的值一起转移了。对于某值 $x = a_{i_p}, i_1 < i_2 < \dots$，我们会考虑找到每个 dp_k 的第一个小于的 i_p，这里直接二分查是值域平方 *log 的，常数比较小可以通过。

CF750,F

再分析，如果我们从图论的角度去思考，先在想象中把 dp 的 DAG 建出来，以 (k, x) 为点建图，k 为异或和，x 为子序列末尾值（也就是我们按值枚举的 a_i），这个点的点权（或者 dp 值）就是最早位置，点个数为 $O(V^2)$，朴素的边个数为 $O(nV)$，很显然这里我们是按 x 的一个分层图，边为 $(k, ?) \to (k \oplus a_{i_p}, a_{i_p}), ? < a_{i_p}$ 。现在按照前面所说的内容，我们需要 $O(1)$ 的找到正确更新每个点的转移边。那么如果我们以最早位置作为 topo 序进行 dp，那么 $(k, ?)$ 通过边 a_i 可以将 $(k \oplus a_i, a_i + 1), (k \oplus a_i, a_i + 2), \ldots, (k \oplus a_i, V)$ 这个后缀更新为 i，可以发现，这个边就是我们需要的边。又因为更新时所有小于 a_i 的 ? 对后缀的更新完全等价，因此边 a_i 仅从 $(k, a_i - 1)$ 转移即可，可以发现这个转移仅第一次是有用的。

CF750,F

那么实现的时候，维护每个末尾 v 还未更新过别人的异或和以及每个异或和的当前末尾值，每次新到一个 a_i 时，去找末尾 a_i-1 中还未更新过别人的异或和尝试去更新，如果激活了新点塞到相应末尾即可，因为 V^2 的点每个点最多只会尝试更新别人一次，因此总复杂度 $O(n+V^2)$

CF750,F

那么实现的时候，维护每个末尾 v 还未更新过别人的异或和以及每个异或和的当前末尾值，每次新到一个 a_i 时，去找末尾 a_i-1 中还未更新过别人的异或和尝试去更新，如果激活了新点塞到相应末尾即可，因为 V^2 的点每个点最多只会尝试更新别人一次，因此总复杂度 $O(n+V^2)$

其实如果有类似的分析的经验，直接到最后一步也是可以的。