FOI2022算法夏令营基础班day8——动态规划

清华大学 陆宏

cf1256e Yet Another Division Into Teams

- 有n个正整数，第i个为a_i。
- 将这n个数分成若干组，每组至少有三个数。
- 最小化所有组的极差之和，其中每组的极差为该组最大的数减最小的数。
- $n \leq 2 \times 10^5, a_i \leq 10^9$

cf1256e Yet Another Division Into Teams

- 注意到每组的数字应尽可能接近，将所有数字从小到大排序，容易发现，每组选的数字一定都是一段连续的区间。
- 记$f(i)$表示只考虑前i个数，得到的极差之和的最小值。
- $f(i) = \min\limits_{j \le i-3} (f(j) + a_i - a_{j+1})$，时间复杂度$O(n^2)$。
- 注意到a_i是定值，于是$f(i) = \min\limits_{j \le i-3} (f(j) - a_{j+1}) + a_i$，记录$f(j) - a_{j+1}$的前缀最小值即可将转移优化至$O(n)$。
- 总时间复杂度为$O(n \log n)$

cf1256e Yet Another Division Into Teams

- 另一个做法。
- 注意到每组的大小一定不超过5，否则可以将其分为至少两组以获得更小的极差之和。
- $f(i) = \min_{i-5 \le j \le i-3} (f(j) + a_i - a_{j+1})$
- 转移部分时间复杂度$O(n)$。
- 总时间复杂度为$O(n\log n)$。

cf1114d Flood Fill

- n个方块排成一排，第i个颜色为c_i。
- 每次操作可以一个区间$[l,r]$，满足这个区间内所有方块颜色相同，并且颜色与$l-1$和$r+1$的方块均不同，然后将$[l,r]$中所有方块的颜色换成另一种。
- 最小化将$[1,n]$全部变成同一种颜色的操作次数。
- $1 \le n, c_i \le 5000$。

cf1114d Flood Fill

- 先把相同颜色的方块缩成一个方块。
- 将一段区间$[l,r]$染成同一种颜色，在最优方案下，最后染成的颜色要么与区间左端点相同，要么与区间右端点相同。
- 令$f(l,r)$表示将区间$[l,r]$染成同一种颜色所需要的最小操作次数。考虑最后一次是如何操作的以得到转移。
- 若l,r颜色相同，则$f(l,r)=f(l+1,r-1)+1$；否则$f(l,r)=\min\left(f(l,r-1),f(l+1,r)\right)+1$。时间复杂度$O(n^2)$。

cf1067a Array Without Local Maximums

- 有一个长度为n的数组a。数字大小均介于1到200之间。满足对于每个位置i，$a_i \leq \max(a_{i-1}, a_{i+1})$。
- 但是数组中有一些位置的数不见了，你希望还原这个数组（在这些位置上填入1~200之间的数字），使其满足上述条件。问方案数，对998244353取模。
- $n \leq 10^5$

cf1067a Array Without Local Maximums

- $f(i,j,0/1)$表示考虑了前i个数，第i个位置填的是j（0/1分别表示 $a_{i-1} < a_i$与$a_{i-1} \geq a_i$）的方案数。
- $f(i,j,0) = \sum_{k=1}^{j-1} (f(i-1,k,1) + f(i-1,k,0))$
- $f(i,j,1) = f(i-1,j,0) + \sum_{k=a}^{200} f(i-1,k,1)$
- 维护$f(\cdot\ ,k,\cdot)$的前缀和转移。
- 时间复杂度$O(200n)$。

cf1572a Book

- 有一本有n个章节的书。每个章节都需要在阅读所有该章节要求的前置章节后，才能读懂。
- 一开始你所有章节都不理解。每一次读书都会按从$1 \sim n$的顺序阅读全部章节，其中遇到无法理解的章节则直接跳过。
- 问需要读多少遍这本书才能理解每一章节。若无论读多少遍书都无法理解所有章节，则输出-1。
- $n \leq 2 \times 10^5$，每个章节要求的前置章节数量之和$\leq 2 \times 10^5$。

cf1572a Book

- 将每个章节的所有前置章节向这个章节连一条边，若前置章节编号比该章节小，则边权为0，否则边权为1。
- 那么图中存在环则无解。否则答案为原图的最长路。
- 记$f(i)$表示理解第i个章节需要读多少遍书（即终点为点i的最长路长度）。
- $f(i)=\max\limits_{j}(f(j)+[j>i])$，其中$j$是$i$的前置章节。
- 按拓扑序转移（或使用记忆化搜索），时间复杂度$O(n)$。

cf1427c The Hard Work of Paparazzi

- 有一个$r \times r$的网格，网格上的每个点用二元组(x, y)表示，从(x, y)移动到(x', y')需要$|x - x'| + |y - y'|$分钟。
- 有n个人，第i个人会在第t_i分钟在(x_i, y_i)出现（并在第$t_i + 1$分钟前离开），只有当你在第t_i分钟时正好在(x_i, y_i)处才能见到他。
- 你初始位于$(1,1)$，希望见到尽可能多的人。
- 问最多能见到多少个人。
- $r \le 500, n \le 10^5, t_i \le 10^6, t_i < t_{i+1}$

cf1427c The Hard Work of Paparazzi

- 令$f(i)$表示考虑了前i个人，你目前正好见到了第i个人（即目前是第t_i分钟且位于(x_i, y_i)），最多能见到多少人。
- $f(i) = \max\limits_{1 \le j < i, |x_i - x_j| + |y_i - y_j| \le t_i - t_j} f(j) + 1$，直接转移是$O(n^2)$的。
- 但是注意到r很小，且网格图上任意两点间移动时间均不超过$2r$，也就是说，对于$1 \le j < i - 2r$，i一定能从j转移过来。
- 于是维护前缀最大值，仅需枚举$i - 2r \le j < i$的j并进行判断即可完成转移。时间复杂度$O(nr)$。

cf1433f Zero Remainder Sum

- 给定$n \times m$的矩阵A和整数p。
- 要求每行选取不超过$\lfloor m/2 \rfloor$个数。并且所有选取的数之和是p的倍数。
- 最大化选取的数之和。
- $n, m, p, A_{i,j} \le 70$

cf1433f Zero Remainder Sum

- $f(i,j)$表示考虑了前i行，选取的数之和模p余j时的最大值是多少。
- $f(i,j) = f(i-1,(j-l) \bmod p) + s_{i,l}$，其中$s_{i,l}$为第$i$行选取不超过$\lfloor m/2 \rfloor$个数，选取的数之和模$p$余$l$时的最大值。
- 那么$s_{i,k}$同样可以由DP求出。$g(j,k,l)$表示考虑了第i行的前j个数，选了k个，模p余l时的最大值是多少。
- $g(j,k,l) = \max\left(g(j-1,k,l), g\left(j-1,k-1,\left(l-A_{i,j}\right) \bmod p\right) + A_{i,j}\right)$
- 那么$s_{i,l} = \max\limits_{k \le \lfloor m/2 \rfloor} g(m,k,l)$。时间复杂度$O(nm^2p)$。

cf1509c The Sports Festival

- 给定长度为n的序列s，你可以任意改变s的顺序。
- 最小化$\sum_{i=1}^{n}\left(\max_{1\le j\le i} s_j - \min_{1\le j\le i} s_j\right)$。
- $n \le 2\times 10^3, s_i \le 10^9$

cf1509c The Sports Festival

- 当$i = n$时，无论怎么排序列s，$\max\limits_{1\le j\le i} s_j - \min\limits_{1\le j\le i} s_j$都是定值。也就是相当于最小化$\sum_{i=1}^{n-1}(\max\limits_{1\le j\le i} s_j - \min\limits_{1\le j\le i} s_j)$。
- 那么重排后的序列的最后一个位置一定填s中的最小值或最大值。
- 证明：若重排后的序列的最小值在最大值前面，那么将最后一个位置上的数与最大值交换一定不会更劣，否则将最后一个位置上的数与最小值交换一定不会更劣。

cf1509c The Sports Festival

- 同理，倒数第二位一定填剩下的数中的最小值或最大值。
- 那么若将s从小到大排序，重排后的序列任意前缀一定都是排好序的s的一段区间。
- 令$f(l,r)$表示只考虑区间$[l,r]$，答案最小是多少。
- $f(l,r)=\max(f(l+1,r),f(l,r-1))+a_r-a_l$
- 时间复杂度$O(n^2)$。

cf1012c Hills

- 有n座山，第i座高度为a_i，若$a_i > \max(a_{i-1}, a_{i+1})$，则你可以在$i$处建房子。
- 你有一台挖掘机，可以花费1小时使任意一座山的高度减1。
- 问对于所有$1 \le k \le \lceil n/2 \rceil$，建$k$栋房子最少需要多少时间。询问之间是独立的。
- $n \le 5000$

cf1012c Hills

- 如果要在i处盖房子，那么$i-1$与$i+1$处一定不能盖房子，于是第i座山一定不会被挖。
- 令$f(i,j,0/1)$表示考虑了前i座山，盖了j栋房子，第i座山有没有盖房子$(0/1)$的最少需要的时间。
- 考虑如何转移。
- 若第i座山不盖房子。若第$i-1$座山盖房子，那么第i座山的高度需挖至$a_{i-1}-1$；否则保持不变。

cf1012c Hills

- 若第i座山要盖房子，那么第$i-1$座山一定不能盖房子。若第$i-2$座山也盖了房子，那么第$i-1$座山的高度需要挖至$\min(a_i, a_{i-2})-1$；否则仅需挖至a_i-1。
- $f(i,j,0)=\min(f(i-1,j,0), f(i-1,j,1)+\max(0, a_i-a_{i-1}+1))$
- $f(i,j,1)=\min(f(i-2,j-1,0)+\max(0, a_{i-1}-a_i+1), f(i-2,j-1,1)+\max(0, a_{i-1}-\min(a_{i-2}, a_i)+1)$
- 时间复杂度$O(n^2)$。

cf1381b Unmerge

- 给定一个长度为$2n$的排列，问是否存在两个长度为n的队列，使得每次取出两个队列的较小队首后，取出的数能排成原排列。（类似归并）
- 例如，3,2,6,1,5,7,8,4能够由3,2,8,4与6,1,5,7归并得到。
- $n \leq 2000$

cf1381b Unmerge

- 注意到，对于那两个队列而言，如果一个数被取出，那么紧随其后的比这个数小的数会立刻被取出。
- 这就意味着对于原排列中的每个前缀最大值，它及它之后的比它小的数需要在同一个队列中。
- 那么原排列就被分成了若干块，每块被取出的时机仅与这块的第一个数有关，且这些块的第一个数递增。也就是说把这些块按顺序放到两个队列中后，它们就可以归并出原排列。

cf1381b Unmerge

- 由于题目要求每个队列长度均为n，于是题目就转化成，给定若干个大小不一的块，问是否存在一些块使得它们一共包含正好n个数字。由于块大小总和为$2n$，剩下的块一定正好包含另外n个数字，将它们分别划分到两个队列中即可。
- 使用01背包解决，时间复杂度$O(n^2)$。

cf1409f Subsequences of Length Two

- 给定两个字符串s, t（仅包含小写字母），每次操作可以选择s中的一个字符，将其替换为任意一个小写字母。
- 最多进行k次操作，最大化t在s中（作为子序列）的出现次数。
- $k \leq |s| \leq 200, |t| = 2$

cf1409f Subsequences of Length Two

- 当t的两个字符相同时，最优解一定是将s中的不等于t_0的字符尽可能改为t_0。以下考虑$t_0 \neq t_1$的情况。
- $f(i, j, k)$表示考虑了s的前i位，做了j次操作，t_0出现了k次，t在s的前i位中出现次数最大值为多少。
- 分三种情况：不改变第$i+1$个字符，将第$i+1$个字符改为t_0，将第$i+1$个字符改为t_1。

cf1409f Subsequences of Length Two

- 为方便起见，记$e_0 = [s_{i+1} = t_0], e_1 = [s_{i+1} = t_1]$。三种情况转移如下：
- $f(i+1, j, k+e_0) = \max(f(i+1, j, k+e_0), f(i, j, k) + (e_1 ? k : 0))$
- $f(i+1, j+1, k+1) = \max(f(i+1, j+1, k+1), f(i, j, k))$
- $f(i+1, j+1, k) = \max(f(i+1, j+1, k), f(i, j, k) + k)$
- 时间复杂度$O(n^3)$。

cf1110d Jongmah

- 给定n个大小不超过m的数字，你可以选择三个连续或三个相同的数字，将它们组成一个三元组。
- 问最多能组成多少个三元组。
- $n, m \leq 10^6$

cf1110d Jongmah

- 记cnt_i为值为i的数字个数。
- 注意到，对于三个连续的数字$[x, x+1, x+2]$，若其出现≥ 3次，则可以将三个$[x, x+1, x+2]$替换为$[x, x, x], [x+1, x+1, x+1], [x+2, x+2, x+2]$。
- 故我们认为，$\forall x, [x, x+1, x+2]$最多出现两次。

cf1110d Jongmah

- 令$f(i,j,k)$表示考虑了所有权值$\le i$的数字，并且三元组$[i-1,i,i+1]$出现了j次，三元组$[i,i+1,i+2]$出现了k次。
- 枚举三元组$[i+1,i+2,i+3]$的出现次数l得到转移。
- $f(i+1,k,l)=\max\left(f(i+1,k,l),f(i,j,k)+l+\lfloor(cnt_{i+1}-j-k-l)/3\rfloor\right)$
- 时间复杂度$O(n)$。

cf1551f Equidistant Vertices

- 给定一棵n个节点的树，求选出k个点，它们两两距离相同的方案数。
- 对10^9+7取模。
- $k \le n \le 100$

cf1551f Equidistant Vertices

- 当$k > 2$时，一定存在一个“中心点”，使得这k个点到它的距离都相同。
- 考虑枚举中心点，以它为根。那么为了满足上述条件，需要选出的k个点两两之间的最近公共祖先均为根。也就是说，要在每个子树中选出一个点，并且这些点的深度相同。

cf1551f Equidistant Vertices

- 那么枚举这k个点的深度d，记cnt_i表示第i棵子树的深度为d的节点个数。
- $f(i,j)$表示考虑了根的前i棵子树，有j个子树取了点的方案数。
- $f(i,j) = f(i-1,j) + f(i-1,j-1) \times cnt_i$
- 分析一下时间复杂度，对于中心点x，其DP的时间复杂度（包括枚举深度在内）为$O(nks_x)$，其中s_x为第x个点的子树个数。
- 那么总复杂度为$O\left(nk\sum_x s_x\right) = O(n^2k)$。

cf1579g Minimal Coverage

- 给定n条线段，第i条长度为l_i。你需要把它们按顺序放在一条无限长的数轴上。
- 放置需满足当前线段的起点是前一个线段的终点，第一个线段的起点为0。可以选择向左覆盖或是向右覆盖。
- 最小化所有线段的覆盖总长度。
- $n \le 10^4, l_i \le 10^3$

cf1579g Minimal Coverage

- 答案一定不超过$2l_{max}$。
- 证明：若当前线段终点坐标< 0，则下一条线段向右覆盖；否则向左覆盖。
- 令$f(i,j,k)$表示已经考虑了前i个线段，第i条线段终点到前i条线段覆盖的区间左端点的距离为j，到前i条线段覆盖的区间右端点为k是否合法。
- 枚举下一条线段向左覆盖还是向右覆盖转移，时间复杂度$O(nl_{max}^2)$。

cf1579g Minimal Coverage

- 考虑优化状态表示，令$f(i,j)$表示已经考虑了前i个线段，第i条线段终点到前i条线段覆盖的区间左端点的距离为j，到前i条线段覆盖的区间右端点距离最小是多少。
- 同样枚举下一条线段向左覆盖还是向右覆盖转移。
- $f(i+1,\max(0,l-a_{i+1}))=\min(f(i+1,\max(0,l-a_{i+1})),f(i,l)+a_{i+1})$
- $f(i+1,l+a_{i+1})=\min(f(i+1,l+a_{i+1}),\max(0,f(i,l)-a_{i+1}))$
- 时间复杂度$O(nl_{max})$。

cf1348e Phoenix and Berries

- 有n棵树，第i棵树上有a_i个红果子和b_i个蓝果子。
- 有无限个容量为k的篮子，每个篮子里装的果子必须是同一种颜色或来自同一棵树。
- 问最多能装满几个篮子。
- $n, k \le 500$，a_i, $b_i \le 10^9$

cf1348e Phoenix and Berries

- 记s_a为总红果子数，s_b为总蓝果子数，则答案的上界为$\left\lfloor\frac{s_a+s_b}{k}\right\rfloor$（不考虑装果子的限制），下界为$\left\lfloor\frac{s_a}{k}\right\rfloor+\left\lfloor\frac{s_b}{k}\right\rfloor$（将同样颜色的果子装进同一个篮子）。
- 注意到上下界最多差1，也就是说，若将相同颜色的果子装入相同篮子后，剩下的红蓝果子数之和$\geq k$，则上界比下界多1，否则上下界相等。
- 上下界相等的情况可以直接输出答案。

cf1348e Phoenix and Berries

- 当上界比下界多1时，如何判断是否能取到上界呢?
- 注意到刚刚只是把相同颜色的果子放在同一个篮子，并没有考虑同一棵树的果子可以放同一篮子的情况。
- 于是可以将取果子的过程分开考虑，先取同一棵树的果子，再将剩下的果子按颜色分到篮子中，如果最后剩的果子数之和$< k$，则可以取到答案的上界。
- 最后剩的果子数之和$< k$等价于取完来自同一棵树的果子之后，剩余两种颜色的果子数量模k之和$< k$。

cf1348e Phoenix and Berries

- 据此，令$f(i,j)$表示考虑了前i棵树，能否只通过取同一棵树的果子做到取出的红果子数量模k为j。（那么蓝果子数量为$k-j$）
- 那么第i棵树能取的红果子数量为$[k-b_i,a_i]$，容易得到转移：
- $f(i,j)=\max_{j-a_i\le l\le j-(k-b_i)} f(i-1,\ l)$
- 注意取模的问题，实际上转移的范围可能是一段前缀加上一段后缀。这里的max实际上是或运算。
- 直接做的复杂度为$O(nk^2)$。

cf1348e Phoenix and Berries

- 由于转移的范围是一个或两个区间，可以使用前缀和优化。
- 时间复杂度可以优化至$O(nk)$。
- 最后，通过$f(n,\cdot)$可以求出剩余的两种果子模k后的数量，由此便可判断答案。

- 谢谢大家