FOI2022算法夏令营提高班
day4——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。
- $n \le 2 \times 10^5, a_i \le 10^9$

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。
- $f(i)=\min_{i-5\le j\le i-3}\left(f(j)+a_i-a_{j+1}\right)$
- 转移部分时间复杂度$O(n)$。
- 总时间复杂度为$O(n\log n)$。

arc108e Random IS

- 给定一个长度为n的排列a，定义一个排列是合法的当且仅当标记的数满足单调递增。
- 每次等概率选择一个未标记过的且标记后序列合法的数标记，无法标记时结束，求期望的标记个数。
- 对10^9+7取模。
- $n \le 2000$。

arc108e Random IS

- 区间DP。为了方便处理，令$a_0 = 0, a_{n+1} = n + 1$。
- $f(i,j)$表示只考虑区间$[i,j]$，且i,j均已被标记，期望还会标记多少个数。
- 枚举下一个标记的位置k。需要满足$a_i \le a_k \le a_j$。
- $f(i,j) = \frac{1}{cnt}\sum_k \left(f(i,k-1) + f(k+1,j)\right) + 1$。其中$cnt$为合法的$k$的个数。

arc108e Random IS

- $f(i,j)=\frac{1}{cnt}\sum_k \big(f(i,k-1)+f(k+1,j)\big)+1$，$a_i \le a_k \le a_j$

- 注意到$f(i,k-1)$与$f(k+1,j)$无关，即$f(i,j)=\frac{1}{cnt}\sum_k f(i,k-1)+\frac{1}{cnt}\sum_k f(k+1,j)+1$，可以使用线段树分开维护两部分的贡献。

- 时间复杂度为$O(n^2\log n)$。

agc030d Inversion Sum

- 给定一个长度为n的整数序列a。依次进行q轮操作。第i轮操作给定两个整数整数x_i和y_i。每一轮操作可以选择以下两种操作之一进行：
  - 交换a_{x_i}和a_{y_i}
  - 什么也不做
- 因此，存在2^q种不同的方法来执行完所有的操作。每一种方法结束后都可以得到一个最终序列。 $n,q \le 3000$
- 询问所有2^q个最终序列的逆序对数之和是多少。对10^9+7取模。

agc030d Inversion Sum

- 直接统计答案并不容易，不妨换一种方式统计。
- 对于下标$i, j(i < j)$，统计在多少种情况下$a_i > a_j$，即产生逆序对。
- $f(t, i, j)$表示进行了t轮操作，$a_i > a_j$的方案数。
- $f(t, x_t, y_t) = f(t, y_t, x_t) = f(t-1, x_t, y_t) + f(t-1, y_t, x_t)$
- $f(t, x_t, j) = f(t, y_t, j) = f(t-1, x_t, j) + f(t-1, y_t, j)$
- $f(t,\ i, x_t) = f(t, i, y_t) = f(t-1, i, x_t) + f(t-1, i, y_t)$
- 其余DP值乘2。直接转移的时间复杂度为$O(qn^2)$。

agc030d Inversion Sum

- 注意到，若i, j均不等于x_t, y_t，那么$f(t, i, j) = 2 \times f(t-1,\ i,\ j)$，也就是说，如果将乘2放在最后进行（即最后答案乘$2^q$），一次操作只会改变$O(n)$个DP值。
- 于是可以优化掉第一维。
- 时间复杂度为$O(n^2 + qn)$。

cf708e Student's Camp

- 有一个$(n+2) \times m$的长方形，除了第一行和最后一行，其他每一行每一天最左边和最右边未被摧毁的格子都有概率p被摧毁，求k天之后最上面一行与最下面一行四联通的概率。
- 答案对10^9+7取模。
- $n, m \le 1500,\ k \le 10^5$。

cf708e Student's Camp

- 令p_i表示k天恰好摧毁i个格子的概率。$p_i = C_k^i p^i (1-p)^{k-i}$
- $f(i,l,r)$表示前i行，第i行未被摧毁的格子为$[l,r]$且前i行连通的概率。
- $f(i,l,r) = p_{l-1} \times p_{m-r} \times \sum_{[l,r]\cap[l',r']\neq\emptyset} f(i-1,l',r')$
- 直接转移复杂度为$O(n^5)$，考虑用总方案减去不合法的方案。
- $f(i,l,r) = p_{l-1} \times p_{m-r} \times \left(\sum_{l'\le r'} f(i-1,l',r') - \sum_{l'\le r'<l} f(i-1,l',r') - \sum_{r<l'\le r'} f(i-1,l',r')\right)$
- 使用前缀和即可优化至$O(n^3)$

cf708e Student's Camp

- $f(i,l,r) = p_{l-1} \times p_{m-r} \times \left(\sum_{l' \le r'} f(i-1,l',r') - \sum_{l' \le r' < l} f(i-1,l',r') - \sum_{r < l' \le r'} f(i-1,l',r')\right)$
- 由于状态已经达到了$O(n^3)$，故首先需要简化状态，注意到答案为$\sum_{l \le r} f(n,l,r)$，记$g(i,l) = h(i,r) = \sum_{l \le r} f(i,l,r)$。
- 那么$g(i,l) = \sum_{l \le r} p_{l-1} \times p_{m-r} \times \left(\sum_{j=1}^{m} g(i-1,j) - \sum_{j=1}^{l-1} h(i-1,j) - \sum_{j=r+1}^{m} g(i-1,j)\right)$，$h$的转移类似。

cf559e Gerald and Path

- 数轴上有n盏灯，第i盏灯的位置为a_i，照明长度为l_i，每盏灯可以选择往左照或者往右照，如果往左照，则照亮区间$[a_i - l_i, a_i]$，否则照亮区间$[a_i, a_i + l_i]$。
- 最大化被照亮的区间总长。
- $\cancel{n \le 100}$ $n \le 5000$

cf559e Gerald and Path

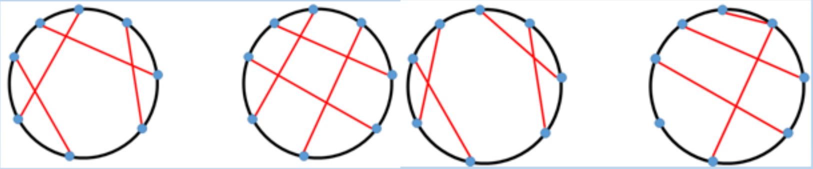
- 先把所有灯按位置排序。有用的点只有$3n$个，于是离散化。
- $f(i,j)$表示前i盏灯，覆盖到的右端点为j，所覆盖的区间总长最大是多少。
- 枚举下一盏往左照的灯在哪，中间的灯往右照。注意，我们认为这盏往左照的灯若照射区间的左端点$< j$，则$< j$的部分不会照亮此前未被照亮的区间。（否则第i盏灯往右照不会更劣）
- 时间复杂度$O(n^3)$。

cf559e Gerald and Path

- 考虑优化掉第一维，使得一个右端点j只对应一个灯i。
- 上面提到中间的灯会往右照，如果照出去的部分仍然覆盖了新的灯，那么这些灯一定也往右照，直到扩展不出新的灯为止。
- 这样i就是j之后的第1盏灯。
- 时间复杂度$O(n^2)$。

agc039e Pairing Points

- 一个圆上排列着$2n$个点，保证任意三对点之间的连线不交于同一点。
- 将这$2n$个点划分成n个点对，并给出01矩阵A，若$A_{i,j}=A_{j,i}=0$，则不能选择i,j构成一个点对。每个点对之间连线，问连线形成一棵树的方案数。
- 右图中只有左一是合法的方案。



- $1\le n\le 20$。

agc039e Pairing Points

- 考虑怎样的连线才能形成一棵树。
- 假设点1与点i相连。找到最小的j满足j与另外一个点k的连线与线段$(1,i)$有交。那么存在$j \le x < i < y \le k$的x, y满足区间$[2,x],[x+1,y-1],[y,2n-1]$中的点只在每个区间内部连线，且每个区间内部连的线段加上每个区间连出去的一条边形成一棵树。
- 那么自然可以得到一个dp。$f(l,r,i)$表示区间$[l,r]$内部除点i外两两配对，点i与外部有连线，形成一棵树的方案数。
- $f(l,r,i) = \sum_{l \le j \le x < i < y \le k \le r, A_{j,k}=1} f(l,x,j) \times f(x+1,y-1,i) \times f(y,r,k)$

agc039e Pairing Points

- $f(l,r,i) = \sum_{l\le j\le x<i<y\le k\le r, A_{j,k}=1} f(l,x,j) \times f(x+1,y-1,i) \times f(y,r,k)$
- 直接做时间复杂度为$O(n^7)$。
- 记$g(l,r,i)$表示区间$[l,r]$外一点i与区间内某个点有连线，区间$[l,r]$内部其它点两两配对，形成一棵树的方案数。
- $g(l,r,i) = \sum_{A_{j,i}=1} f(l,r,j)$
- $f(l,r,i) = \sum_{l\le x<i<y\le r} (f(x+1,y-1,i) \times \left(\sum_{l\le j\le x} f(l,x,j) \times g(y,r,j)\right))$
- 由于计算$\sum_{l\le j\le x} f(l,x,j) \times g(y,r,j)$不需要用到$i$，且仅考虑

agc040e Prefix Suffix Addition

- 给定一个长为n的初始全为0的序列x。
- 有以下两种操作：

1. 选定整数k和不降，非负，长度为k的序列c，将x_i加上c_i。
2. 选定整数k和不升，非负，长度为k的序列c，将x_{n-k+i}加上c_i。

- 最小化操作次数使得操作结束时x为一个给定序列。
- $n \le 2 \times 10^5$。

agc040e Prefix Suffix Addition

- 注意到，在操作的序列c前加入若干个0，便可以做到选择任意一个区间，将其加上一个非负且单调不降/不升的序列c。
- 记b_i, c_i分别为操作1/2在第i个位置加上的权值。显然$x_i = b_i + c_i$。
- 假设知道了b_i, c_i。则答案为将b_i贪心划分为尽可能少的单调不降的序列，c_i贪心划分为尽可能少的单调不升的序列的序列个数之和。为了方便起见，将序列x前后各加上一个0，即$x_0 = x_{n+1} = 0$。
- 则答案为$\sum_{i=0}^{n}([b_i > b_{i+1}] + [c_i < c_{i+1}])$。

agc040e Prefix Suffix Addition

- $f(i,j)$为考虑了前i个数，第i个数的$b_i=j$的$\sum_{k=0}^{i-1}([b_k>b_{k+1}]+[c_k<c_{k+1}])$最小是多少。
- 关于$f(i,j)$，有两个性质：
  - 当i固定时，$f(i,j)$单调不升。
  - $f(i,0)\le f(i,a_i)+2$。
- 随着j的增大，$[b_{i-1}>b_i],[c_{i-1}<c_i]$更不可能成立。即满足性质1。
- $[b_{i-1}>b_i]+[c_{i-1}<c_i]\le 2$，即$f(i,0)$最多比$f(i,a_i)$大2，满足性质2。

agc040e Prefix Suffix Addition

- 这样一来，对于每个i，$f(i,j)$最多只有三种取值且三种取值连续，对于同一取值的DP值，只需考虑b_i最小的点如何进行转移，由此确定$f(i+1,j)$的三种取值的分界点。
- 时间复杂度$O(n)$。

- 谢谢大家