Longest Increasing Subsequence

题目大意

给出一个带空缺的序列(空缺的位置用-1标出)，

用给出的m个数填补空缺（每个数只能使用一次），使得最长（严格）上升子序列最大，输出填补后的序列。

数据范围

序列的长度$n(1 \le n \le 100000)$

空缺的位置数k（$0 \le k \le 1000)$

用来填补空缺的数的个数$m(k \le m \le 10^5)$

序列中的数和用来填补空缺的数都大于0且不超过10^9

解法

因为填同一个数的两个位置不会在同一个上升子序列，所以不会使答案变大，所以可以忽略”每个数只能使用一次“这个条件。

这样就可以用普通的求最长上升的子序列的方法求出$len_i, from_i$，

len_i为在1到i中包含i的最长上升子序列的长度（i不是空缺的位置），

$from_i$为1到i中包含i的最长上升子序列的$len_i - 1$的位置是哪里（i不是空缺的位置）。

我的方法是设f_i表示长度为i的上升子序列的最后一项的最小值（没有长度为i最长上升子序列则值为inf），$last_i$则是f_i的在序列中的位置。

从前往后枚举，如果不是空缺的位置就在f上二分找到最大的f_j小于这一位的值，那么$len_i = j + 1, from_i = last_j$，然后更新对应的$f$与$last$。

如果是空缺位置就枚举从大到小枚举填补空缺的数，用一个指针来代替二分来找到最大的f_j小于枚举的数，然后更新对应的f与$last$。

枚举的过程要记录下当前最长上升子序列的最后一个数的位置。

算出最长上升子序列的长度后考虑，如何还原这个序列。

考虑我们还原到了最长上升子序列的第i个数，在原序列在第pos位，

如果这个位置不是空缺，就可以用$from_{pos}$知道第$i - 1$个数在哪个位置。

如果这个位置是空缺，在不是空缺的位置找到$len_j = i - 1$且$a_j < a_{pos}$且$j < pos$（这个可以用$vector$存下对于每个$len = k$的位置，在$vector$上二分找到pos前的一个位置，判断这个位置的a是否满足条件），

如果存在那么第$i-1$个数就是这个位置j

如果不存在那么第$i-1$个数就是上一个空缺，且这个空缺上的数就是填补的数中比pos为上填的数要小的最大的 那个。

时间复杂度 $O(n\log n+mk)$

空间复杂度 $O(n+m)$

Black Widow

题目大意

给出一个形如这样的式子

$$(v_{1,1}\vee\ldots\vee v_{1,k_1})\oplus(v_{2,1}\vee\ldots\vee v_{2,k_2})\oplus\ldots\oplus(v_{n,1}\vee\ldots\vee v_{n,k_n})=1$$

和m个布尔变量x_i

其中每个$k\le 2$，每个$v_{i,j}$都是一个是$x_{a_{i,j}}$或者$\neg x_{a_{i,j}}$

求解的数量（模$1e9+7$）

数据范围

$1\le n,m\le 100000$

$k_i\le 2$

解法

如果将一个将每一个包含两个变量子句（每一个括号）中的看成连一条边，一个变量看成自环。

那么整张图就是一堆环和一堆链。

对于链就先处理某一个端点（自环），然后设$f[0/1][0/1]$表示上一个变量取$0/1$，异或和为$0/1$的数量，

在链上转移就枚举选0还是1来转移，最后处理另外一个端点的自环。

对于环就先枚举环上某一个点的变量的取值，然后破环为链，按照和链一样dp，最后在处理在这个点与最后一个的子句即可。

注意：可能会有自环会有两个。

时间复杂度$O(n+m)$

空间复杂度$O(n+m)$

Colorful Sequences

题目大意

给出整数N,K，和长度为M整数序列A

所有数都在$[1,K]$的整数序列被称为$colorful$，当且仅当存在长度为K的连续子序列，其中1到K的数都出现一次。

对于每个长度为N $colorful$整数序列，统计这个序列的连续子序列是A的数量，求这些数量的和模10^9+7

数据范围

$1 \le M \le N \le 25000$

$1 \le A_i \le K \le 400$

解法

容易知道如果没有“是$colorful$长度为N的整数序列”而是“长度为N的正整数序列”，所以答案就是$K^{N-M}(N-M+1)$。

答案就是那么就是要减去不是$colorful$序列的数量。

首先考虑一个简单的问题：
　　计算不$colorful$的序列个数。

设$dp[i][j]$表示不$colorful$的序列长度为i，序列最后j个数互不相同且最后$j+1$个数不满足的方案数。

$dp[i][j]$到$dp[i+1][j']$的转移系数如下：

- 当$j+1<j'$，0。
- 当$j+1=j'$，$K-j$。
- 当$j+1>j'$，1。

直接转移是$O(NK^2)$，而加上前缀和优化则是$O(NK)$。

考虑如何计算不是$colorful$序列中A的数量。

case 1. A是colorful

显然为0。

case 2. A中的数互不相同，A不是colorful

如果计算出长度为N的所有$colorful$序列中是的长度为M的连续子序列里的数互不相同的数量。

那么答案就是这个数除$\frac{K!}{(K-M)!}$。

考虑计算出不$colorful$序列中长度为M连续子序列里的数互不相同的数量。

这个可以使用在一开始提到的dp，再多加上一个$sum[i][j]$表示所有$dp[i][j]$中的的序列的长度为M连续子序列里的数互不相同的数量，转移时一起转移，每次如果$j>=M$，那么$sum[i][j]$就加上$dp[i][j]$。

case 3.A中的数有相同的

求出A中最大的前缀，最大的后缀其中的每一个数都互不相同，其长度分别记为pre，suf。

设$f[i]$表示长度为$i+pre$且开头是pre个互不相同的数的不$colorful$序列数，$g[i]$同理。

计算f和g使用一开始提到的dp，然后初值设一下，即可。

那么枚举A开头所在的位置i，计算出$f[i-1]*g[N-M-(i-1)]$就是这个A在开头i不$colorful$序列数。