

我觉得很牛逼 题意:一个序列p，所有的逆序对间直接存在一条边，现在将这些点染色，相邻的点颜色不能相同，问最少用多少种颜色，并输出每个点的颜色

解析：不难想到的是，最少的颜色种类为最长单调递减序列的长度，现在考虑每个点染什么颜色的问题，假设第i个点染得颜色为x,那么i节点之前至少存在一个已i节点结尾的最长单调递减序列，长度为x，因为单调递减序列中的点两两相连，所以需要的颜色为序列的长度。我们通过上述解释可以得到一种涂色方式：某节点的颜色为以该节点结尾的最长单调递减序列的长度。

然后我觉得求最长下降子序列，反过来然后就最长上升子序列就可以了

#include<bits/stdc++.h>

using namespace std;

const int N=1000010;

int a[N],c[N];

int f[N];

int main()

{

int t;

scanf("%d",&t);

while(t--)

{ int m=0;

int n;

scanf("%d",&n);

for(int i=1;i<=n;i++) scanf("%d",&a[i]);

reverse(a+1,a+1+n);

for(int i=1;i<=n;i++)

{

int k=lower\_bound(f+1,f+m+1,a[i])-f;

c[i]=k; f[k]=a[i];

m=max(m,k);

}

reverse(c+1,c+n+1);

printf("%d\n",m);

for(int i=1;i<=n;i++)

printf("%d ",c[i]);

printf("\n");

}

return 0;

}