P239

#include<cstdio>

#include<cstring>

#include<iostream>

#include<sstream>

#include<algorithm>

#include<set>

using namespace std;

int a[1000010];

int main()

{

// freopen("input.txt","r",stdin);

int T,n;

scanf("%d",&T);

while(T--)

{

set<int>s;

scanf("%d",&n);

for(int i=0;i<n;i++)

scanf("%d",&a[i]);

int L=0,R=1,ans=0;

s.insert(a[0]);

while(R<n)

{

while(R<n && s.count(a[R])==0)

{

s.insert(a[R]);

R++;

}

ans=max(ans,R-L);

s.erase(a[L]);

L++;

}

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

}

return 0;

}

另一种方法：用map求出last[i]，即下标i的“上一个相同元素的下标”

#include<iostream>

#include<stdio.h>

#include<map>

const int maxn=1000010;

int a[maxn],last[maxn];

using namespace std;

int main()

{

int T,n;

scanf("%d",&T);

while(T--)

{

map<int,int>m;

for(int i=0;i<maxn;i++)last[i]=0;

scanf("%d",&n);

for(int i=0;i<n;i++)

{

scanf("%d",&a[i]);

if(m.count(a[i])==0) //map中是位置和编号对应

last[i]=-1;

else

last[i]=m[a[i]];//此时a[i]还没有加入m中,所以m[a[i]]中存的是上一个值为a[i]的元素的下标

m[a[i]]=i;

}

int L=0,R=0,ans=0;

while(R<n)

{

while(R<n && last[R]<L)

R++;

ans=max(ans,R-L);

L++;

}

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

}

return 0;

}