线段树 区间最大值和最小值的差 开两棵树

Update操作不需要，因为没有更新

#include <cstdio>

#include<iostream>

#include<vector>

#include<algorithm>

#include<string>

#include<cstring>

#include<set>

#include<map>

using namespace std;

#define ll long long

#define INF 1000000007

int a1[50010\*4],a2[50010\*4];

inline void build(int root,int l,int r)

{

if(l==r)

{

scanf("%d",&a1[root]);

a2[root]=a1[root];

return ;

}

int mid=(l+r)>>1;

build(root<<1,l,mid);

build(root<<1|1,mid+1,r);

a1[root]=max(a1[root<<1],a1[root<<1|1]);

a2[root]=min(a2[root<<1],a2[root<<1|1]);

//a存的是总的区间和

}

inline int getmax(int x,int y,int root,int l,int r)

{

if(x<=l && r<=y)

return a1[root];

int mid=(l+r)>>1;

int ans=0;

if(x<=mid)//中间的mid点还是在左区间上

ans=max(ans,getmax(x,y,root<<1,l,mid));

if(y>mid)

ans=max(ans,getmax(x,y,root<<1|1,mid+1,r));

return ans;

}

inline int getmin(int x,int y,int root,int l,int r)

{

if(x<=l && r<=y)

return a2[root];

int mid=(l+r)>>1;

int ans=INF;

if(x<=mid)//中间的mid点还是在左区间上

ans=min(ans,getmin(x,y,root<<1,l,mid));

if(y>mid)

ans=min(ans,getmin(x,y,root<<1|1,mid+1,r));

return ans;

}

int main()

{

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

int n,q;

while(~scanf("%d%d",&n,&q))

{

build(1,1,n);

int x,y;

while(q--)

{

scanf("%d%d",&x,&y);

printf("%d\n",getmax(x,y,1,1,n)-getmin(x,y,1,1,n));

}

}

return 0;

}