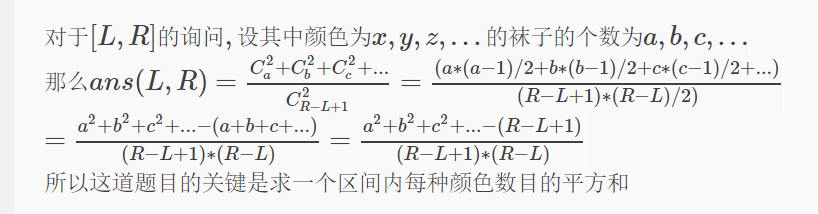
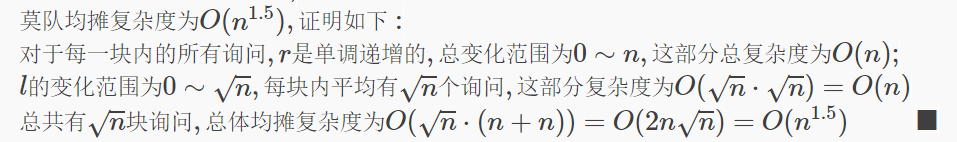
分析：



思考：

莫队的第一题，很好的理解莫队区间之间的转移。若知道[l,r]的答案，可以在O(1)的时间内得到[l,r-1],[l,r+1],[l-1,r],[l+1,r]。



Code:

#include<iostream>

#include<cstdio>

#include<cstring>

#include<string>

#include<algorithm>

#include<cmath>

#include<queue>

#include<map>

#include<vector>

#include<set>

#include<bitset>

#include<stack>

#include<cctype>

#define sc(x) scanf("%d",&x);

#define PR(x) cout<<#x<<": "<<x<<endl;

using namespace std;

typedef long long ll;

typedef pair<int,int> p;

const double eps = 1e-4;

const int mod = 1e9+7;

const ll INF = 0x3f3f3f3f3f3f3f3f;

const int inf = 0x3f3f3f3f;

const int maxn = 5e4+7;

const int maxm = 1e6+7;

const double pi = acos(-1.0);

int n,m,unit;

struct Query{

int l,r,id;

}node[maxn];

ll gcd(ll a,ll b){

if(b==0)

return a;

return gcd(b,a%b);

}

bool cmp(Query a,Query b){

if(a.l/unit!=b.l/unit)

return a.l/unit<b.l/unit;

else

return a.r<b.r;

}

struct Node{

ll a,b;

void reduce(){

ll d=gcd(a,b);

a/=d,b/=d;

}

}ans[maxn];

int a[maxn];

int num[maxn];

void solve(){

ll temp=0;

memset(num,0,sizeof num);

int l=1,r=0;

for(int i=0; i<m; i++){

while(r<node[i].r){

r++;

temp-=(ll)num[a[r]]\*num[a[r]];

num[a[r]]++;

temp+=(ll)num[a[r]]\*num[a[r]];

}

while(r>node[i].r){

temp-=(ll)num[a[r]]\*num[a[r]];

num[a[r]]--;

temp+=(ll)num[a[r]]\*num[a[r]];

r--;

}

while(l<node[i].l){

temp-=(ll)num[a[l]]\*num[a[l]];

num[a[l]]--;

temp+=(ll)num[a[l]]\*num[a[l]];

l++;

}

while(l>node[i].l){

l--;

temp-=(ll)num[a[l]]\*num[a[l]];

num[a[l]]++;

temp+=(ll)num[a[l]]\*num[a[l]];

}

ans[node[i].id].a=temp-(r-l+1);

ans[node[i].id].b=(ll)(r-l+1)\*(r-l);

ans[node[i].id].reduce();

}

}

int main(){

while(scanf("%d%d",&n,&m)!=EOF){

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

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

for(int i=0; i<m; i++){

node[i].id=i;

scanf("%d%d",&node[i].l,&node[i].r);

}

unit=(int)sqrt(n);

sort(node,node+m,cmp);

solve();

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

printf("%lld/%lld\n",ans[i].a,ans[i].b);

}

return 0;

}