思考：

有一个比较重要的常熟优化：O(1)转移的时候要写成一个式子，写两个会T。

#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 = 2e6+7;

const int maxm = 2e6+7;

const double pi = acos(-1.0);

struct FastIO {

static const int S = 1310720;

int wpos; char wbuf[S];

FastIO() : wpos(0) {}

inline int xchar() {

static char buf[S];

static int len = 0, pos = 0;

if (pos == len)

pos = 0, len = fread(buf, 1, S, stdin);

if (pos == len) return -1;

return buf[pos ++];

}

inline int xuint() {

int c = xchar(), x = 0;

while (c <= 32) c = xchar();

for (;'0' <= c && c <= '9'; c = xchar()) x = x \* 10 + c - '0';

return x;

}

inline int xint() {

int s = 1, c = xchar(), x = 0;

while (c <= 32) c = xchar();

if (c == '-') s = -1, c = xchar();

for (; '0' <= c && c <= '9'; c = xchar()) x = x \* 10 + c - '0';

return x \* s;

}

inline void xstring(char \*s) {

int c = xchar();

while (c <= 32) c = xchar();

for(; c > 32; c = xchar()) \*s++ = c;

\*s = 0;

}

inline void wchar(int x) {

if (wpos == S) fwrite(wbuf, 1, S, stdout), wpos = 0;

wbuf[wpos ++] = x;

}

inline void wint(ll x) {

if (x < 0) wchar('-'), x = -x;

char s[24];

int n = 0;

while (x || !n) s[n ++] = '0' + x % 10, x /= 10;

while (n--) wchar(s[n]);

}

inline void wstring(const char \*s) {

while (\*s) wchar(\*s++);

}

~FastIO() {

if (wpos) fwrite(wbuf, 1, wpos, stdout), wpos = 0;

}

} io;

int n,m,unit;

struct Query{

int l,r,id;

}node[maxn];

bool cmp(Query a,Query b){

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

return a.r<b.r;

else

return a.l<b.l;

}

ll ans[maxn];

ll a[maxn];

ll num[maxn];

void solve(){

ll sum=0;

memset(num,0,sizeof num);

int l=1,r=0;

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

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

r++;

sum+=(ll)(num[a[r]]\*2+1)\*a[r];

num[a[r]]++;

}

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

num[a[r]]--;

sum-=(ll)(num[a[r]]\*2+1)\*a[r];

r--;

}

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

num[a[l]]--;

sum-=(ll)(num[a[l]]\*2+1)\*a[l];

l++;

}

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

l--;

sum+=(ll)(num[a[l]]\*2+1)\*a[l];

num[a[l]]++;

}

// cout<<l<<" "<<r<<" "<<sum<<endl;

ans[node[i].id]=sum;

}

}

int main(){

scanf("%d%d",&n,&m);

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

a[i]=io.xint();

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

}

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

node[i].id=i;

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

node[i].l=io.xint();

node[i].r=io.xint();

}

unit=(int)sqrt(n\*1.0)+10;

sort(node,node+m,cmp);

solve();

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

printf("%I64d\n",ans[i]);

return 0;

}