

 #include <bits/stdc++.h>

using namespace std;

const int N = 1e5+20 , SIZE = 1020;

int n,m,size,T,raw[N],val[N],t,cnt[N],cnt\_[N];

int belo[N],L[SIZE],R[SIZE];

long long ans[N],Max,a[N];

struct query{int l,r,id;}q[N];

inline void input(void)

{

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

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

scanf("%lld",&a[i]) , raw[++t] = a[i];

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

scanf("%d%d",&q[i].l,&q[i].r) , q[i].id = i;

}

inline void discrete(void)

{

sort( raw+1 , raw+t+1 );

t = unique( raw+1 , raw+t+1 ) - (raw+1);

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

val[i] = lower\_bound( raw+1 , raw+t+1 , a[i] ) - raw;

}

inline void setblocks(void)

{

size = sqrt(n) , T = n/size;

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

{

if ( i \* size > n ) break;

L[i] = (i-1) \* size + 1;

R[i] = i \* size;

}

if ( R[T] < n ) T++ , L[T] = R[T-1] + 1 , R[T] = n;

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

for (int j=L[i];j<=R[i];j++)

belo[j] = i;

}

inline bool compare(query p1,query p2)

{

if ( belo[p1.l] ^ belo[p2.l] )

return belo[p1.l] < belo[p2.l];

else return p1.r < p2.r;

}

// 加点

inline void insert(int p,long long &Maxval)

{

cnt[val[p]]++;

Maxval = max( Maxval , 1LL \* cnt[val[p]] \* a[p] );

}

// 撤销

inline void resume(int p)

{

cnt[val[p]]--;

}

inline void CaptainMo(void)

{

sort( q+1 , q+m+1 , compare );

int l = 1 , r = 0 , lastblock = 0;

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

{

// 处理同一块中的询问

if ( belo[q[i].l] == belo[q[i].r] )

{

for (int j=q[i].l;j<=q[i].r;j++) cnt\_[val[j]]++;

long long temp = 0;

for (int j=q[i].l;j<=q[i].r;j++)

temp = max( temp , 1LL \* cnt\_[val[j]] \* a[j] );

for (int j=q[i].l;j<=q[i].r;j++) cnt\_[val[j]]--;

ans[ q[i].id ] = temp;

continue;

}

// 如果移动到了一个新的块，就先把左右端点初始化

if ( lastblock ^ belo[q[i].l] )

{

while ( r > R[belo[q[i].l]] ) resume(r--);

while ( l < R[belo[q[i].l]]+1 ) resume(l++);

Max = 0 , lastblock = belo[q[i].l];

}

// 单调地移动右端点

while ( r < q[i].r ) insert(++r,Max);

// 移动左端点回答询问

long long temp = Max; int l\_ = l;

while ( l\_ > q[i].l ) insert(--l\_,temp);

// 回滚

while ( l\_ < l ) resume(l\_++);

ans[ q[i].id ] = temp;

}

}

int main(void)

{

input();

discrete();

setblocks();

CaptainMo();

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

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

return 0;

}