莫队

```cpp
#include<bits/stdc++.h>
#define LL __int128
#define endl '\n'
#define int long long
using namespace std;
typedef long long ll;
typedef unsigned long long ull;
ll gcd(ll x,ll y){return y?gcd(y,x%y):x;}
ll lcm(ll x,ll y){return x/gcd(x,y)*y;}
ll qpow(ll a,ll b,ll p){a%=p; ll ret=1;for(;b;b>>=1,a=a*a%p) if(b&1) ret=ret*a%p; return ret;}
ll qpow(ll a,ll b){ll ret=1; for(;b;b>>=1,a*=a) if(b&1) ret*=a; return ret;}
ll getInv(ll x,ll p){return qpow(x,p-2,p);}
struct T {
    int l, r, id;
};
const int N = 5 + 1e6;
bool np[N];
int tot, pri[N];
vector<int> a[N];
int ans;
int cnt[N], rk[N];
void add(int x) {
    for (auto &v: a[x]) {
        cnt[rk[v]]--;   //维护区间众数，cnt[f(x)]
        rk[v]++;
        cnt[rk[v]]++;
        ans = max(ans, rk[v]);
    }
}
void del(int x) {
    for (auto &v: a[x]) {
        cnt[rk[v]]--;
        if (ans == rk[v] && !cnt[rk[v]]) ans--;
        rk[v]--;
        cnt[rk[v]]++;
    }
}
T q[N];
int res[N];
signed main(){
    ios::sync_with_stdio(false),cin.tie(0),cout.tie(0);
    np[1] = 1;
    for (int i = 2; i < N; i++) {
        if (!np[i]) { pri[++tot] = i; }
        for (int j = 1; j <= tot && i * pri[j] < N; j++) {
            np[i * pri[j]] = 1;
            if (i % pri[j] == 0) { break; }
        }
    }
    int _; cin >> _;
    while (_--) {
        int n, m, block; cin >> n >> m; block = sqrt(n); //*    //分块大小
        for (int i = 1; i <= n; i++) {
            int x; cin >> x;
            a[i].clear();
            for (int j = 1; j <= tot && pri[j] * pri[j] <= x; j++) {
                if (x % pri[j] == 0) {
                    a[i].push_back(pri[j]);
                    while (x % pri[j] == 0) {
                        x /= pri[j];
                    }
                }
            }
            if (x > 1) { a[i].push_back(x); }
        }
        for (int i = 0; i < m; i++) {
            cin >> q[i].l >> q[i].r; q[i].id = i;
        }
        sort(q, q + m, [&](T x, T y) {  //*
            if (x.l / block != y.l / block) { return x.l / block < y.l / block; }
            else if (x.l / block & 1) { return x.r < y.r; } //奇偶优化
            else return x.r > y.r;

        });
        ans = 0;
        int l = 1, r = 0;   //*
        for (int i = 0; i < m; i++) {
            int L = q[i].l, R = q[i].r;
            while (l > L) add(--l); //*
            while (r < R) add(++r); //*
            while (l < L) del(l++); //*
            while (r > R) del(r--); //*
            res[q[i].id] = ans;
        }
        while (l <= r) { del(l++); }
        for (int i = 0; i < m; i++) {
            cout << res[i] << endl;
        }
    }
}
```