分析：首先，将牛按minSPF升序，防晒霜按SPF\_i升序，把当前的防晒霜抹给minSPF小于等于SPF\_i的牛中maxSPF最小的。因为maxSPF越大（假设minSPF相同），可选择的防晒霜越多。至于怎么去maxSPF最小，可以用一个优先队列来维护最小值。

#include <iostream>

#include <algorithm>

#include<vector>

#include<queue>

#include<cstring>

#include<cstdio>

#define ll long long

using namespace std;

struct node1

{

int minn,maxn;

};

node1 cow[3000];

struct node2

{

int spf,cover;

};

node2 bottle[3000];

bool cmp1(node1 x,node1 y)

{

return x.minn<y.minn;

}

bool cmp2(node2 x,node2 y)

{

return x.spf<y.spf;

}

int main()

{

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

ios::sync\_with\_stdio(false);

cin.tie(0);

int c,l;

cin>>c>>l;

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

cin>>cow[i].minn>>cow[i].maxn;

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

cin>>bottle[i].spf>>bottle[i].cover;

sort(cow,cow+c,cmp1);

sort(bottle,bottle+l,cmp2);

int cur(0),ans(0); //cur现在正等待涂防晒霜的奶牛的下标

priority\_queue<int,vector<int>,greater<int> > q;

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

{

while(cow[cur].minn<=bottle[i].spf && cur<c)

{

q.push(cow[cur].maxn);

cur++;

}

while(bottle[i].cover>0 &&!q.empty())

{

int maxspf=q.top();q.pop();

if(maxspf>=bottle[i].spf)

{

bottle[i].cover--;

ans++;

}

}

}

cout<<ans<<endl;

return 0;

}