思路：设长<=宽，将各物体按长从小到大排序。从后向前枚举长，同时将宽加入并排序。则对于位置 j 的宽，有 num-j 物体宽大于它（由于长从小到大枚举，保证新加入物体的长不大于当前物体的长），则体积 a[i].r\*b[j]\*(num-j) 每次取最大即可。

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

struct node

{

int lon,shor;

}a[4010];

bool cmp(node a,node b)

{

if(a.lon==b.lon)return a.shor<b.shor; //先按长从小到大，再按宽从小到大

return a.lon<b.lon;

}

int main()

{

int n,x,y;

cin>>n;

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

{

cin>>x>>y;

a[i].lon=max(x,y);

a[i].shor=min(x,y);

}

sort(a,a+n,cmp);

vector<int> v;

int l;

ll ans(0);

int ansx,ansy;

for(int i=n-1;i>=0;i--)

{

v.push\_back(a[i].shor);

sort(v.begin(),v.end());

l=v.size();

for(int j=0;j<l;j++)//枚举集合里的每个宽度

{

ll tmp=(ll)a[i].lon\*v[j]\*(l-j);

//对于位置j的宽,有l-j的物体宽大于它

//(l-j)就是满足条件的个数

if(tmp>ans)

{

ans=tmp;

ansx=a[i].lon;ansy=v[j];

}

}

}

cout<<ans<<endl;

cout<<ansx<<' '<<ansy<<endl;

return 0;

}