题意

给你n个银行，每次银行的位置x[i],金钱w[i]， 现在让你选择两个不同的银行 使其距离大于等于D 且金钱和最大

输出选择的那两个银行，序号

首先，看到这个题，那肯定是先给银行按坐标排个序

找到比当前i坐标大b的银行的坐标j，那么j-n的银行坐标就都比

当前i坐标的银行大b了，所以从他们中找到一个最大值

这样的话，是n^2

而数据的1e5

题解：

我是先预处理出 i ~n 中 金钱最多的，序号是那个

再遍历一次， 二分当前x[i]+ d的 序号就好了， 即 与其距离满足大于等于d且 金钱最多的是哪一个银行 ，更新答案

Pos[i]存的是从i到n中金钱最多的银行的下标

W[i]存的是i到n中金钱最多的银行的钱数

#include <iostream>

#include <algorithm>

#include<cstring>

#include<cstdio>

#include<cmath>

#define maxn 1e6+10

typedef long long ll;

using namespace std;

int w[1000010];

int pos[1000010];

int v[1000010];

int x[1000010];

int n,d;

void init()

{

w[n+1]=0;

pos[n+1]=-1;

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

{

if(v[i]>w[i+1])

{

w[i]=v[i];

pos[i]=i;

}

else

{

w[i]=w[i+1];

pos[i]=pos[i+1];

}

}

}

int main()

{

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

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

cin>>n>>d;

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

cin>>x[i]>>v[i];

init();

int ans(0),tmp,p;

int pos1=-1,pos2=-1;

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

{

tmp=x[i]+d;

p=lower\_bound(x+1,x+1+n,tmp)-x;

if(p==n+1)break;

if(v[i]+w[pos[p]]>ans)

{

ans=v[i]+w[pos[p]];

pos1=i;pos2=pos[p];

}

}

cout<<pos1<<' '<<pos2<<"\n";

return 0;

}