There are many rook on a chessboard, a rook can attack the row and column it belongs, including its own place.   
  
There are also many queries, each query gives a rectangle on the chess board, and asks whether every grid in the rectangle will be attacked by any rook?

Input

The first line of the input is a integer *T*

, meaning that there are *T* test cases.   
  
Every test cases begin with four integers *n*,*m*,*K*,*Q*.   
*K* is the number of Rook, *Q* is the number of queries.   
  
Then *K* lines follow, each contain two integers *x*,*y* describing the coordinate of Rook.   
  
Then *Q* lines follow, each contain four integers *x*1,*y*1,*x*2,*y*2 describing the left-down and right-up coordinates of query.   
  
1≤*n*,*m*,*K*,*Q*≤100,000.   
  
1≤*x*≤*n*,1≤*y*≤*m*.   
  
1≤*x*1≤*x*2≤*n*,1≤*y*1≤*y*2≤*m*

.

Output

For every query output "Yes" or "No" as mentioned above.

Sample Input

2

2 2 1 2

1 1

1 1 1 2

2 1 2 2

2 2 2 1

1 1

1 2

2 1 2 2

Sample Output

Yes

No

Yes

Hint

Huge input, scanf recommended.

前缀和的应用

题意:在一个棋盘上有一些"车"，他能够攻击到与它同一行或者同一列的棋盘上的所有的格子，现在给出K个棋子的坐标，然后有Q组询问，每一次询问(x1,y1,x2,y2)这个方格内的所有棋子是否能够全部被攻击到。

题解:维护前缀和,统计 (x1-x2) 这一段区间里面的被攻击到的行的数量，统计(y1-y2)这一段区间里面的被攻击到的列的数量,如果sum(x1~x2) == x2-x1+1 ，那么这段区间全部能够被攻击到,列也就不用考虑了，对列的考虑亦如此。

可以发现如果一个矩阵被全部攻击到，

很显然要么是因为它的每一行都有车，

或者每一列都有车。

所以只需要记录一下哪些行和哪些列有车，

对于每个询问只需要做一个前缀和就可以知道答案了。

#include <iostream>

#include<stdio.h>

using namespace std;

int T,n,m,k,q,X,Y,x1,x2,y1,y2;

int x[100010],y[100010];

int main()

{

scanf("%d",&T);

while(T--)

{

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

for(int i=1;i<=n;i++)x[i]=0;

for(int i=1;i<=m;i++)y[i]=0;

while(k--)

{

scanf("%d%d",&X,&Y);

x[X]=1;

y[Y]=1;

}

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

x[i]+=x[i-1];//x[i]存的是1 to x的前缀和

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

y[i]+=y[i-1];

while(q--)

{

scanf("%d%d%d%d",&x1,&y1,&x2,&y2);

if((x[x2]-x[x1-1]==x2-(x1-1)) || (y[y2]-y[y1-1]==y2-(y1-1)))

puts("Yes");

else

puts("No");

}

}

return 0;

}