

最后求得是答案的逆元，很显然，离散化处理后，可以暴力判断出来抹黑哪些方块可以抹黑最后的大方块，那么就可以用状压倒着推出来f[0]的答案，这种题目尤其要注意的一点是，我们真正需要的其实是整个面涂黑，而数据给出的是点，所以我们需要在x2，y2处-1来表示那个面而不是表示点

#include<bits/stdc++.h>

#define ll long long

#define inf 1e18

#define mn 100005

#define mod 998244353

using namespace std;

ll ans,n,m,w,h,b[mn],c[25][25],tot,bj[1050],f[1050];

struct ww{ll x,y,x2,y2;}a[mn];

ll cal(ll x,ll t)

{ll ret=1;

for(;t;t/=2,x=x\*x%mod)

if(t%2==1)ret=ret\*x%mod;

return ret;

}

int main()

{

ll t,x,y,z,i,j,k,l;

char ch;

cin>>t;

while(t--)

{

memset(f,0,sizeof(f));

memset(bj,0,sizeof(bj));

scanf("%lld",&n);

scanf("%lld%lld",&w,&h);

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

scanf("%lld%lld%lld%lld",&a[i].x,&a[i].y,&a[i].x2,&a[i].y2);

n++;

a[n].x=0;a[n].y=0;a[n].x2=w;a[n].y2=h;

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

{

b[(i-1)\*2+1]=a[i].x;

b[i\*2]=a[i].x2;

}

sort(b+1,b+2\*n+1);

tot=unique(b+1,b+2\*n+1)-b-1;

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

{

a[i].x=lower\_bound(b+1,b+tot+1,a[i].x)-b;

a[i].x2=lower\_bound(b+1,b+tot+1,a[i].x2)-b;

}

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

{

b[(i-1)\*2+1]=a[i].y;

b[i\*2]=a[i].y2;

}

sort(b+1,b+2\*n+1);

tot=unique(b+1,b+2\*n+1)-b-1;

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

{

a[i].y=lower\_bound(b+1,b+tot+1,a[i].y)-b;

a[i].y2=lower\_bound(b+1,b+tot+1,a[i].y2)-b;

}

w=a[n].x2;h=a[n].y2;

n--;

ll bjj=0;

for(i=0;i<(1<<n);i++)

{

for(j=1;j<=22;j++)

for(k=1;k<=22;k++)

c[j][k]=0;

for(j=1;j<=n;j++)

if(i&(1<<(j-1)))

{

for(k=a[j].x;k<=a[j].x2-1;k++)//注意需要标记的是面而不是点

for(l=a[j].y;l<=a[j].y2-1;l++)

c[k][l]=1;

}

for(k=1;k<=w-1;k++)

{

for(l=1;l<=h-1;l++)

if(c[k][l]==0)break;

if(l!=h)break;

}

if(k==w){bj[i]=1;bjj=1;}

}

if(bjj==0){printf("-1\n");continue;}

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

if(bj[i]==0)

{

k=0;

for(j=1;j<=n;j++)

if(i&(1<<(j-1)))k++;

ll dd=cal(n-k,mod-2);

for(j=1;j<=n;j++)

{

if(i&(1<<(j-1)))continue;

f[i]=(f[i]+dd\*f[i|(1<<(j-1))]%mod)%mod;

}

f[i]=(f[i]+dd\*n%mod)%mod;

// cout<<i<<" "<<f[i]<<endl;

}

printf("%lld\n",f[0]);

}

return 0;

}