

2020.05.18-2020.05.24 周报

团队训练

2020.05.20 [2016 Multi-University Training Contest 3 prob\[\]6:6:11 rnk\[\]44/?](#)

[20200520比赛记录](#)

_wzx27

[\\$P\acute{o}lya定理\\$](#)

Infinity37

专题

[一些敲简单的博弈论](#)

题目

[B-permutation_bo](#)

[J-rower_bo](#)

[K-teacher_bo](#)

Zars19

题目

[D-Gambler_bo](#)

POJ1054 The Troublesome Frog 暴力

青蛙从田野外跳到田野外的另一端，步长均等，路径是直线，每跳一步都会留下痕迹。给出\$N\$个痕迹的坐标，问痕迹最多的一条青蛙路径有几个痕迹 $3 \leq N \leq 5000, 1 \leq R, C \leq 5000$

题解：不知道为什么乱入DP专题，其实是暴力+剪枝优化。先把点按照横坐标排序，这样二重循环只一个方向即可。然后剪枝，先减掉逆方向跳一次跳不出田野的情况，然后减掉沿着个方向一直到跳出去都无法更新当前答案的情况。然后验证。



```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<cmath>
#include<algorithm>
#define LL long long
using namespace std;
const int N=5005;
int read()
{
    int x=0,f=1;char c=getchar();
    while(c<'0' || c>'9'){if(c=='-')f=-1;c=getchar();}
    while(c>='0'&&c<='9'){x=x*10+c-'0';c=getchar();}
    return x*f;
}
int r,c,n,frog[N][N],vis[N][N],res=2;
struct Node{int x,y;}dot[N];
bool cmp(Node o1,Node o2){return (o1.x==o2.x)?(o1.y<o2.y):(o1.x<o2.x);}
bool in(int x,int y){return x>=1&&x<=r&&y>=1&&y<=c;}
int main()
{
    r=read(),c=read(),n=read();
    for(int i=1;i<=n;i++)
    {
        dot[i].x=read(),dot[i].y=read();
        frog[dot[i].x][dot[i].y]=1;
    }
    sort(dot+1,dot+1+n,cmp);
    for(int i=1;i<=n;i++)
    {
        for(int j=i+1;j<=n;j++)
        {
            if(vis[dot[j].x][dot[j].y]==i)continue;
            vis[dot[j].x][dot[j].y]=i;
            int dx=dot[j].x-dot[i].x,dy=dot[j].y-dot[i].y;
            int tx=dot[i].x-dx,ty=dot[i].y-dy;
            if(in(tx,ty))continue;
            if(!in(dot[i].x+(res-1)*dx,dot[i].y+(res-1)*dy))continue;
            tx=dot[j].x,ty=dot[j].y;
            int num=1;
            while(in(tx,ty))
            {
                num++;
                if(!frog[tx][ty]){num=0;break;}
                tx+=dx,ty+=dy;
            }
            res=max(res,num);
        }
    }
}
```

```

}
printf("%d\n", res<3?0:res);
return 0;
}

```

POJ 1925 Spiderman DP

n 幢楼，蜘蛛侠在第一幢要荡到最后一幢的横坐标那里，每幢楼可以抽象成 $(x_i, 0)$ 到 (x_i, y_i) 的线段，摆荡时绳子不能长于挂绳建筑的高度。问最少摆荡次数。

题解：坐标范围是 1000000 ，可以以此dp[]用 $f[i]$ 表示荡到横坐标为 i 的位置最少荡几次。这里要注意的是由于对称性空中停驻点的纵坐标永远是 y_1 于是可以扫一遍楼用 $f[x[i]-j]$ 更新 $f[x[i]+j]$

```

#include<iostream>
#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<cmath>
#include<algorithm>
#define LL long long
#define INF 0x3f3f3f3f
using namespace std;
const int N=5005;
int read()
{
    int x=0,f=1;char c=getchar();
    while(c<'0' || c>'9'){if(c=='-')f=-1;c=getchar();}
    while(c>='0' && c<='9'){x=x*10+c-'0';c=getchar();}
    return x*f;
}
LL x[N],y[N];
int f[1000005];
LL p(LL a){return a*a;}
int main()
{
    int T=read();
    while(T--)
    {
        memset(f,0x3f,sizeof(f));
        int n=read();
        for(int i=1;i<=n;i++)x[i]=read(),y[i]=read();
        f[x[1]]=0;
        for(int i=2;i<=n;i++)
        {
            int d=sqrt(p(y[i])-p(y[i]-y[1]));
            for(int j=1;j<=d;j++)
            {
                if(x[i]-j<x[1])continue;
                if(x[i]+j>x[n])f[x[n]]=min(f[x[i]-j]+1,f[x[n]]);
                else f[x[i]+j]=min(f[x[i]-j]+1,f[x[i]+j]);
            }
        }
    }
}

```

```
    }  
    }  
    printf("%d\n", f[x[n]]==INF?-1:f[x[n]]);  
}  
return 0;  
}
```

比赛

本周推荐

From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai_milk:weekly3&rev=1590248751

Last update: 2020/05/23 23:45

