

题意

滑动解锁是智能手机一项常用的功能。你需要在3x3的点阵上，从任意一个点开始，反复移动到一个尚未经过的"相邻"的点。这些划过的点所组成的有向折线，如果与预设的折线在图案、方向上都一致，那么手机将解锁。两个点相邻当且仅当以这两个点为端点的线段上不存在**尚未经过的点**。此外，这条折线还需要至少经过4个点。

为了描述方便，我们给这9个点从上到下、从左到右依次编号1-9。那么1->2->3是不合法的，因为长度不足。1->3->2->4也是合不合法的，因为1->3穿过了**尚未经过的点**2。2->4->1->3->6是合法的，因为1->3时点2已经被划过了。

作为一个爱逛知乎的好少年，小Hi已经知道[一共有389112种不同的解锁方案](#)。不过小Hi不满足于此，他希望知道，当已经瞥视到一部分折线的情况下，有多少种不同的方案。

遗憾的是，小Hi看到的部分折线不一定是连续的，也不知道方向。例如看到1-2-3和4-5-6，那么1->2->3->4->5->6，1->2->3->6->5->4，3->2->1->6->5->4->8->9等都是合法的方案。

分析

预处理出中间存在其他点的点对

按题目要求[搜索](#)

代码

```
1 #include<algorithm>
2 #include<bitset>
3 #include<cstdio>
4 #include<cstring>
5 #include<cstdlib>
6 #include<cmath>
7 #include<deque>
8 #include<iostream>
9 #include<map>
10 #include<queue>
11 #include<set>
12 #include<stack>
13 #include<string>
14 #include<vector>
15 #include<list>
```

```

16 #define For(i,a,b) for(int i=(a); i<=(b) ; i++)
17 #define _For(i,a,b) for(int i=(a); i>=(b) ; i--)
18 #define Memset(a,b); memset((a),(b),sizeof((a)));
19 #define Cout(a,b); printf("%d",(a));printf(b);
20 #define Coutc(a,b); printf("%c",(a));printf(b);
21 #define Couts(a,b); printf("%s",(a));printf(b);
22 using namespace std;
23 const int INF = 0x3f3f3f3f;
24 typedef long long LL;typedef unsigned long long ULL;typedef long double
    LDB;
25 inline LL CinLL(){LL x=0,f=1;char ch=getchar();while(ch<'0' || ch>'9'){if(ch=='-'
    ')f=-1;ch=getchar();}while(ch>='0'&&ch<='9'){x=x*10+ch-
    '0';ch=getchar();}return x*f;}
26 inline int Cin(){int x=0,f=1;char ch=getchar();while(!isdigit(ch)){if(ch=='-'
    ')f=-1;ch=getchar();}while(isdigit(ch))x=x*10+ch-'0',ch=getchar();return f*x;}
27 int a[10][10];
28 int n;
29 int book[10];
30 int ans;
31 int pass[10][10];
32 void ycl()
33 {
34     pass[1][3] = 2;pass[1][7] = 4;pass[1][9] = 5;
35     pass[2][8] = 5;
36     pass[3][1] = 2;pass[3][9] = 6;pass[3][7] = 5;
37     pass[4][6] = 5;
38     pass[6][4] = 5;
39     pass[7][1] = 4;pass[7][3] = 5;pass[7][9] = 8;
40     pass[8][2] = 5;
41     pass[9][1] = 5;pass[9][7] = 8;pass[9][3] = 6;
42 }
43 void dfs(int pp,int now,int tot,int cs)
44 {
45     if(tot == n) {
46         if(cs >=4)
47             ans++;
48     }
49     For(i,1,9)
50     {
51         if(i == now) continue;
52         if(book[i] == 0)
53         {
54
55             if(book[pass[now][i]] == 1)
56             {
57                 book[i] = 1;

```

```

58         if(a[now][i] == 1) dfs(now,i,tot+1,cs+1);
59         else dfs(now,i,tot,cs+1);
60         book[i] = 0;
61     }
62 }
63 }
64 }
65
66 int main()
67 {
68     ios::sync_with_stdio(false);
69     ycl();
70     int _;
71     cin>>_;
72     while(_--)
73     {
74         cin>>n;
75         int x,y;
76         Memset(a,0);
77         For(i,1,n){
78             cin>>x>>y;
79             a[y][x] = a[x][y] = 1;
80         }
81         ans = 0;
82         For(i,1,9)
83         {
84             Memset(book,0);
85             book[0] = book[i] = 1;
86             dfs(-1,i,0,1);
87         }
88         cout<<ans<<endl;
89     }
90 }

```