

```

1.  /*****      kosaraju final by Shadman      *****/
2.  #include <bits/stdc++.h>
3.  using namespace std;
4.  vector<int>v[50001],v1[50001],v2;
5.  stack<int>s;
6.  map<int,int>m,m1;
7.  bool vis[50001];
8.  int mf[50001],maf[50001];
9.  void dfs2(int a)
10. {
11.     vis[a]=true;
12.     int b=v1[a].size();
13.     for(int i=0;i<b;i++)
14.     {
15.         int d=v1[a][i];
16.         if(!vis[d])
17.         {
18.             dfs2(d);
19.         }
20.     }
21.     v2.push_back(a);
22. }
23. void dfs(int a)
24. {
25.     vis[a]=true;
26.     int b=v[a].size();
27.     for(int i=0;i<b;i++)
28.     {
29.         int d=v[a][i];
30.         if(!vis[d])
31.         {
32.             dfs(d);
33.         }
34.     }
35.     s.push(a);
36. }
37. int dfs1(int a)
38. {
39.     vis[a]=true;
40.     int c=maf[a];
41.     int b=v[a].size();
42.     for(int i=0;i<b;i++)
43.     {
44.         int d=v[a][i];
45.         if(!vis[d])
46.         {
47.             c=c+dfs1(d);
48.         }
49.     }
50.     return c;
51. }
52. int dfs3(int a)
53. {
54.     vis[a]=true;
55.     int c=maf[a];
56.     int b=v1[a].size();
57.     for(int i=0;i<b;i++)
58.     {

```

```

59.         int d=v1[a][i];
60.         if(!vis[d])
61.         {
62.             c=c+dfs3(d);
63.         }
64.     }
65.     return c;
66. }
67. void dfs_reverse(int a,int p)
68. {
69.     mf[a]=p;
70.     maf[p]++;
71.     vis[a]=true;
72.     int b=v1[a].size();
73.     for(int i=0;i<b;i++)
74.     {
75.         int d=v1[a][i];
76.         if(!vis[d])
77.         {
78.             dfs_reverse(d,p);
79.         }
80.     }
81. }
82. int main()
83. {
84.     int a,b,c,d,e,i,j,k,p,q,x,y,T;
85.     scanf("%d",&T);
86.     for(i=1;i<=T;i++)
87.     {
88.         v2.clear();
89.         p=0;
90.         q=0;
91.         scanf("%d",&a);
92.         for(j=0;j<=50000;j++)
93.         {
94.             v[j].clear();
95.             v1[j].clear();
96.             mf[j]=-1;
97.             maf[j]=0;
98.         }
99.         memset(vis,false,sizeof(vis));
100.        m.clear();
101.        m1.clear();
102.        for(j=1;j<=a;j++)
103.        {
104.            scanf("%d %d",&c,&d);
105.            v[c].push_back(d);
106.        }
107.        for(j=1;j<=a;j++)
108.        {
109.            if(!vis[j])
110.                dfs(j);
111.        }
112.        memset(vis,false,sizeof(vis));
113.        for(j=1;j<=a;j++)
114.        {
115.            c=v[j].size();
116.            for(k=0;k<c;k++)

```

```

117.         {
118.             d=v[j][k];
119.
120.             v1[d].push_back(j);
121.         }
122.     }
123.     while(!s.empty())
124.     {
125.         c=s.top();
126.         s.pop();
127.         if(!vis[c])
128.         {
129.             dfs_reverse(c,p);
130.             p++;
131.         }
132.     }
133.     x=-1;
134.     q=-1;
135.     for(j=0;j<=50000;j++)
136.         v1[j].clear();
137.     for(j=1;j<=a;j++)
138.     {
139.         c=v[j].size();
140.         for(k=0;k<c;k++)
141.         {
142.             e=mf[j];
143.
144.             d=v[j][k];
145.             d=mf[d];
146.             if(d!=e)
147.                 v1[e].push_back(d);
148.         }
149.     }
150.     d=0;
151.     e=0;
152.     if(p==1)
153.     {
154.         cout<<"Case "<<i<<": "<<1<<endl;
155.     }
156.     else
157.     {
158.         memset(vis, false, sizeof(vis));
159.         for(j=0;j<p;j++)
160.         {
161.             if(!vis[j])
162.                 dfs2(j);
163.         }
164.         memset(vis, false, sizeof(vis));
165.         for(j=0;j<=50000;j++)
166.             v[j].clear();
167.         for(j=1;j<=a;j++)
168.         {
169.             if(v[mf[j]].size()==0)
170.             {
171.                 v[mf[j]].push_back(j);
172.             }
173.         }
174.         for(j=p-1;j>=0;j--)
175.         {
176.             memset(vis, false, sizeof(vis));

```

```
175.         if(!vis[v2[j]])
176.         {c=dfs3(v2[j]);
177.         if(c>x)
178.         {
179.             x=c;
180.             q=v[v2[j]].front();
181.
182.         }
183.         else if(c==x && v[v2[j]].front()<q)
184.         {
185.             x=c;
186.             q=v[v2[j]].front();
187.         }
188.     }
189. }
190. cout<<"Case "<<i<<": "<<q<<endl;
191. }
192. }
193. return 0;
194.
195. }
```