

省选基础算法

雷宇辰

2017 年 2 月 12 日

目录

1 day1 图论	1
1.1 有向图强连通分量的 Tarjan 算法	1
2 day2	5
3 day3	6
4 day4	7
5 day5	8
6 day6	9
7 day7	10
8 day8	11
9 day9	12
10 day10	13
11 day11	14
12 day12	15
13 day13	16
14 day14	17
15 day15	18

1 day1 图论

1.1 有向图强连通分量的 Tarjan 算法

定义 在有向图 G 中, 如果两个顶点 u, v 间存在一条路径 u 到 v 的路径且也存在一条 v 到 u 的路径, 则称这两个顶点 u, v 是**强连通的 (strongly connected)**。如果有向图 G 的每两个顶点都强连通, 称 G 是一个**强连通图**。有向非强连通图的极大强连通子图, 称为**强连通分量 (strongly connected components)**。若将有向图中的强连通分量都缩为一个点, 则原图会形成一个 DAG (有向无环图)。

极大强连通子图 G 是一个极大强连通子图当且仅当 G 是一个强连通子图且不存在另一个强连通子图 G' 使得 G 是 G' 的真子集。

Tarjan 算法 定义 $dfn(u)$ 为结点 u 搜索的次序编号, 给出函数 $low(u)$ 使得

$low(u) = \min$

```
{
     $dfn(u)$ ,
     $low(v)$ ,    $(u, v)$  为树枝边,  $u$  为  $v$  的父结点
     $dfn(v)$     $(u, v)$  为后向边或指向栈中结点的横叉边
}
```

当结点 u 的搜索过程结束后, 若 $dfn(u) = low(u)$, 则以 u 为根的搜索子树上所有还在栈中的结点是一个强连通分量。

代码

tarjan - SCC

```
1 void tarjan(int u)
2 {
3     dfn[u] = low[u] = ++idx;
4     st[top++] = u;
5     for (Edge cur : G[u])
6         if (!dfn[cur.to])
7             tarjan(cur.to),
8             low[u] = min(low[u], low[cur.to]);
9         else if (!scc[cur.to])
10            low[u] = min(low[u], dfn[cur.to]);
11 if (dfn[u] == low[u] && ++cnt)
12     do scc[st[--top]] = cnt;
13     while (st[top] != u);
14 }
```

练习题

POJ2186/ BZOJ1051 - Popular Cows 双倍的快乐

Popular Cows

```
1 #include <cstdio>
2 inline int min(int a, int b) { return a < b ? a : b; }
3 int head[10010], next[50010], to[50010], ecnt;
4 int dfn[10010], low[10010], stk[10010], scc[10010], top, idx, scccnt;
5 bool instk[10010];
```

```

6  int deg[10010];
7  inline void addEdge(int f, int t)
8  {
9      ecnt++;
10     next[ecnt] = head[f];
11     head[f] = ecnt;
12     to[ecnt] = t;
13 }
14 void tarjan(int x)
15 {
16     dfn[x] = low[x] = ++idx;
17     instk[stk[top++] = x] = true;
18     for (int cur = head[x]; cur; cur = next[cur])
19         if (!dfn[to[cur]])
20             tarjan(to[cur]), low[x] = min(low[x], low[to[cur]]);
21         else if (instk[to[cur]])
22             low[x] = min(low[x], dfn[to[cur]]);
23     if (dfn[x] == low[x])
24     {
25         scccnt++;
26         do
27         {
28             top--;
29             scc[stk[top]] = scccnt;
30             instk[stk[top]] = false;
31         } while (stk[top] != x);
32     }
33 }
34 int main()
35 {
36     int n, m;
37     scanf("%d%d", &n, &m);
38     for (int i = 0, x, y; i < m; i++)
39     {
40         scanf("%d%d", &x, &y);
41         addEdge(x, y);
42     }
43     for (int i = 1; i <= n; i++)
44         if (!dfn[i])
45             tarjan(i);
46     for (int i = 1; i <= n; i++)
47         for (int cur = head[i]; cur; cur = next[cur])
48             if (scc[i] != scc[to[cur]])
49                 deg[scc[i]]++;
50     int zcnt = 0, id = 0;
51     for (int i = 1; i <= scccnt; i++)
52         if (deg[i] == 0)
53             zcnt++, id = i;
54     if (zcnt != 1)
55         putchar('0');
56     else
57     {
58         int ans = 0;
59         for (int i = 1; i <= n; i++)
60             if (scc[i] == id)
61                 ans++;
62         printf("%d", ans);
63     }
64     return 0;
65 }

```

POJ3180 - The Cow Prom The N ($2 \leq N \leq 10,000$) cows are so excited.

The Cow Prom

```

1  #include <stdio>
2  inline int min(int a, int b) { return a < b ? a : b; }
3  const int maxn = 100010;
4  int head[maxn], next[maxn << 1], to[maxn << 1], ecnt, n, m;
5  int dfn[maxn], scc[maxn], cnt[maxn], scccnt, stk[maxn], low[maxn], idx, top;
6  inline void addEdge(int f, int t)
7  {
8      ecnt++;
9      next[ecnt] = head[f];
10     head[f] = ecnt;
11     to[ecnt] = t;
12 }
13 void tarjan(int x)
14 {
15     dfn[x] = low[x] = ++idx;
16     stk[top++] = x;
17     for (int i = head[x]; i; i = next[i])
18         if (!dfn[to[i]])
19             tarjan(to[i]), low[x] = min(low[x], low[to[i]]);
20         else if (!scc[to[i]])
21             low[x] = min(low[x], dfn[to[i]]);
22     if (dfn[x] == low[x])
23     {
24         scccnt++;
25         do
26             scc[stk[--top]] = scccnt;
27         while (stk[top] != x);
28     }
29 }
30 int main()
31 {
32     scanf("%d%d", &n, &m);
33     for (int i = 0, x, y; i < m; i++)
34     {
35         scanf("%d%d", &x, &y);
36         addEdge(x, y);
37     }
38     for (int i = 1; i <= n; i++)
39         if (!dfn[i]) tarjan(i);
40     int ans = 0;
41     for (int i = 1; i <= n; i++) cnt[scc[i]]++;
42     for (int i = 1; i <= scccnt; i++)
43         if (cnt[i] > 1) ans++;
44     printf("%d", ans);
45     return 0;
46 }

```

POJ1236 - Network of Schools 强连通分量缩点求出度为 0 的和入度为 0 的分量个数

Network of Schools

```

1  #include <stdio>
2  inline int min(int a, int b) { return a < b ? a : b; }
3  const int maxn = 110, maxm = 10100;
4  int head[maxn], next[maxm], to[maxm], ecnt, f[maxn], g[maxn];
5  inline void addEdge(int f, int t)
6  {
7      ecnt++;

```

```

8     next[ecnt] = head[f];
9     head[f] = ecnt;
10    to[ecnt] = t;
11 }
12 int dfn[maxn], low[maxn], stk[maxn], scc[maxn], scccnt, top, idx;
13 void tarjan(int x)
14 {
15     dfn[x] = low[x] = ++idx;
16     stk[top++] = x;
17     for (int i = head[x]; i; i = next[i])
18         if (!dfn[to[i]])
19             tarjan(to[i]), low[x] = min(low[x], low[to[i]]);
20         else if (!scc[to[i]])
21             low[x] = min(low[x], dfn[to[i]]);
22     if (dfn[x] == low[x])
23     {
24         scccnt++;
25         do
26             scc[stk[--top]] = scccnt;
27         while (stk[top] != x);
28     }
29 }
30 int main()
31 {
32     int n;
33     scanf("%d", &n);
34     for (int i = 1, x; i <= n; i++)
35         for (scanf("%d", &x); x; scanf("%d", &x))
36             addEdge(i, x);
37     for (int i = 1; i <= n; i++)
38         if (!dfn[i]) tarjan(i);
39     for (int i = 1; i <= n; i++)
40         for (int j = head[i]; j; j = next[j])
41             if (scc[i] != scc[to[j]])
42                 f[scc[i]]++, g[scc[to[j]]]++;
43     int ans1 = 0, ans2 = 0;
44     if (scccnt == 1)
45         printf("1\n0");
46     else
47     {
48         for (int i = 1; i <= scccnt; i++)
49             ans1 += f[i] == 0, ans2 += g[i] == 0;
50         printf("%d\n%d", ans2, ans1 > ans2 ? ans1 : ans2);
51     }
52     return 0;
53 }

```

2 day2

3 day3

4 day4

5 day5

6 day6

7 day7

8 day8

9 day9

10 day10

11 day11

12 day12

13 day13

14 day14

15 day15