

# ACM 模板

钱智煊，黄佳瑞，车昕宇

2024 年 9 月 13 日

# 目录

<b>1</b>	<b>图论</b>	<b>3</b>
1.1	连通性相关 . . . . .	3
1.1.1	tarjan . . . . .	3
1.1.2	tarjan 求 LCA . . . . .	3
1.1.3	割点、割边 . . . . .	3
1.1.4	圆方树 . . . . .	3
1.2	同余最短路 . . . . .	3
<b>2</b>	<b>字符串</b>	<b>4</b>

# 1 图论

## 1.1 连通性相关

### 1.1.1 tarjan

```

1 void tarjan(int x)
2 {
3     dfn[x]=low[x]=++Time,sta[++tp]=x,ins[x]=true;
4     for(int i=hea[x];i;i=nex[i])
5     {
6         if(!dfn[ver[i]])
7             ↪ tarjan(ver[i]),low[x]=min(low[x],low[ver[i]]);
8         else if(ins[ver[i]])
9             ↪ low[x]=min(low[x],dfn[ver[i]]);
10    }
11    if(dfn[x]==low[x])
12    {
13        do
14        {
15            x=sta[tp],tp--,ins[x]=false;
16            } while (dfn[x]!=low[x]);
17    }
18 }

```

### 1.1.2 tarjan 求 LCA

实现均摊  $O(1)$ 。就是用 tarjan 按照顺序遍历子树的特点加上并查集即可。

```

1 inline void add_edge(int x,int y){
2     ↪ ver[++tot]=y,nex[tot]=hea[x],hea[x]=tot; }
3 inline void add_query(int x,int y,int d)
4     { qver[++qtot]=y,qnex[qtot]=qhea[x],qhea[x]=qtot,
5       ↪ qid[qtot]=d; }
6 int Find(int x){ return (fa[x]==x)?x:(fa[x]=Find(fa[x])); }
7 void tarjan(int x,int F)
8 {
9     vis[x]=true;
10    for(int i=hea[x];i;i=nex[i])
11    {
12        if(ver[i]==F) continue;
13        tarjan(ver[i],x),fa[ver[i]]=x;
14    }
15    for(int i=qhea[x];i;i=qnex[i])
16    {
17        if(!vis[qver[i]]) continue;
18        ans[qid[i]]=Find(qver[i]);
19    }
20 }
21 for(int i=1;i<=n;i++) fa[i]=i;
22 for(int i=1,x,y;i<=n;i++)
23     ↪ x=rd(),y=rd(),add_edge(x,y),add_query(y,x,i);
24 for(int i=1,x,y;i<=m;i++)
25     ↪ x=rd(),y=rd(),add_query(x,y,i),add_query(y,x,i);
26 tarjan(s,s);
27 for(int i=1;i<=m;i++) printf("%d\n",ans[i]);

```

### 1.1.3 割点、割边

注意这里的  $dfn$  表示不经过父亲，能到达的最小的  $dfn$ 。割点：

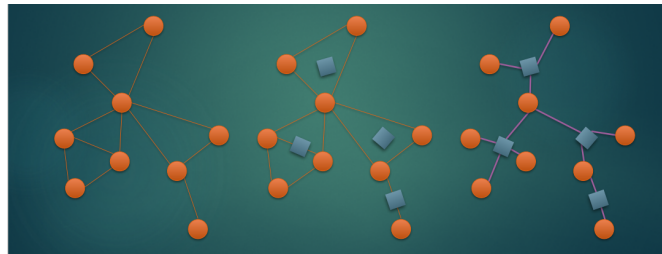
- 若  $u$  是根节点，当至少存在 2 条边满足  $low(v) \geq dfn(u)$  则  $u$  是割点。
- 若  $u$  不是根节点，当至少存在 1 条边满足  $low(v) \geq dfn(u)$  则  $u$  是割点。

割边：

- 当存在一条边满足  $low(v) > dfn(u)$  则边  $i$  是割边。

注意：记录上一个访问的边时要记录边的编号，不能记录上一个过来的节点（因为会有重边）!!! 或者在加边的时候特判一下，不过注意编号问题。（用输入顺序来对应数组中位置的时候，重边跳过，但是需要  $tot+=2$ ）

### 1.1.4 圆方树



圆方树会建立很多新的点，所以不要忘记给数组开两倍！

```

1 void tarjan(int u)
2 {
3     dfn[u]=low[u]=++Time,sta[++tp]=u;
4     for(int v:G[u])
5     {
6         if(!dfn[v])
7         {
8             tarjan(v),low[u]=min(low[u],low[v]);
9             if(low[v]==dfn[u])
10            {
11                int hav=0; ++All;
12                for(int x=0;x!=v;tp--)
13                    ↪ x=sta[tp],T[x].pb(All),T[All].pb(x),hav++;
14                T[u].pb(All),T[All].pb(u);
15                siz[All]++;hav;
16            }
17        }
18        else low[u]=min(low[u],dfn[v]);
19    }
20 }

```

## 1.2 同余最短路

形如：

- 设问 1：给定  $n$  个整数，求这  $n$  个整数在  $h(h \leq 2^{63} - 1)$  范围内能拼凑出多少的其他整数（整数可以重复取）。
- 设问 2：给定  $n$  个整数，求这  $n$  个整数不能拼凑出的最小（最大）的整数。

设  $x$  为  $n$  个数中最小的一个，令  $ds[i]$  为只通过增加其他  $n-1$  种数能够达到的最低楼层  $p$ ，并且满足  $p \equiv i \pmod{x}$ 。

对于  $n-1$  个数与  $x$  个  $ds[i]$ ，可以如下连边：

```

1 for(int i=0;i<x;i++) for(int j=2;j<=n;j++)
2     ↪ add(i,(i+a[j])%x,a[j]);

```

之后进行最短路，对于：

- 问在  $h$  范围内能够到达的点的数量：答案为（加一因为  $i$  本身也要计算）：

$$\sum_{i=0}^{x-1} [d[i] \leq h] \times \frac{h-d[i]}{x} + 1$$

- 问不能达到的最小的数：答案为  $(i - 1) \times x + 1$ （注意： $ds$  与  $h$  范围相同，一般也要开 `long long` !  
 $d[i] = s \times x + i$ ，则  $(s - 1) \times x + i$  一定不能被表示出来）；

$$\min_{i=1}^{x-1} \{d[i] - x\}$$

## 2 字符串