

图的连通

2020年1月20日

黄哲威 hzwer

北京大学16级计算机科学



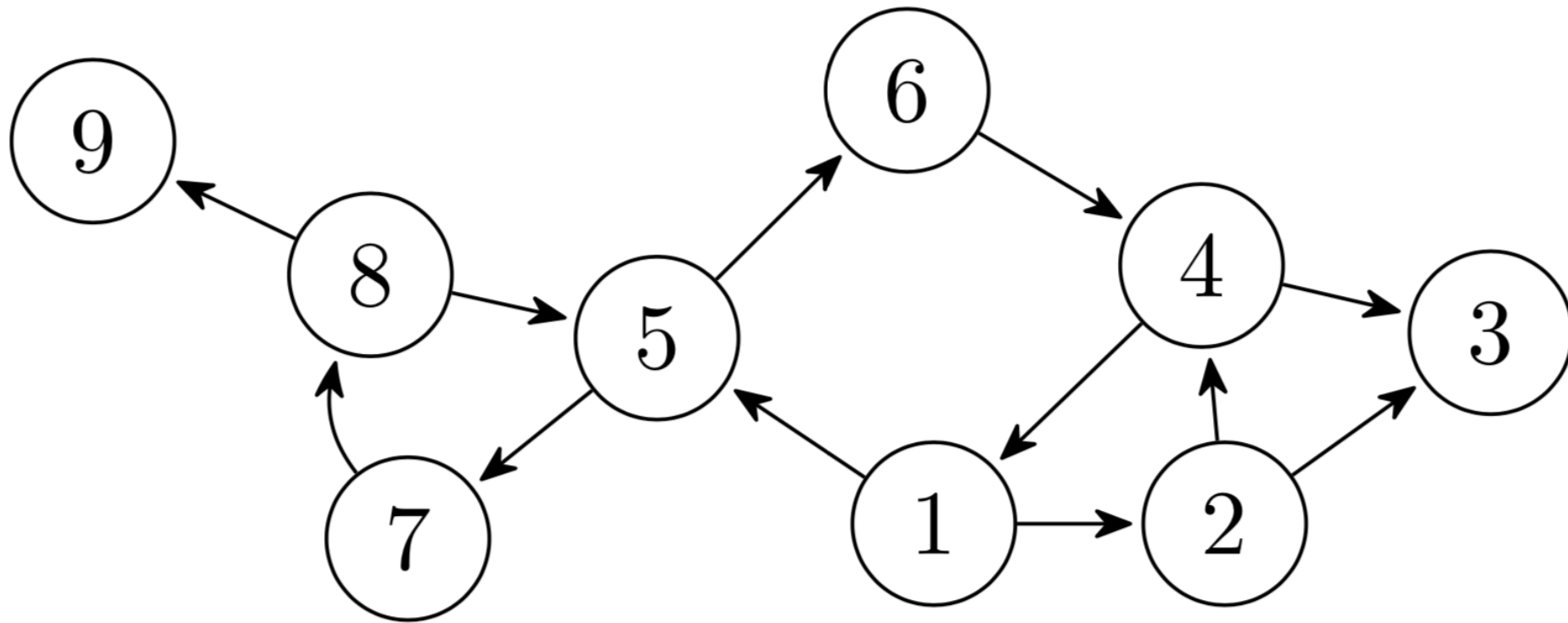
课程安排 3

Tarjan 算法

强连通分量缩点重建图

有向图强连通分量

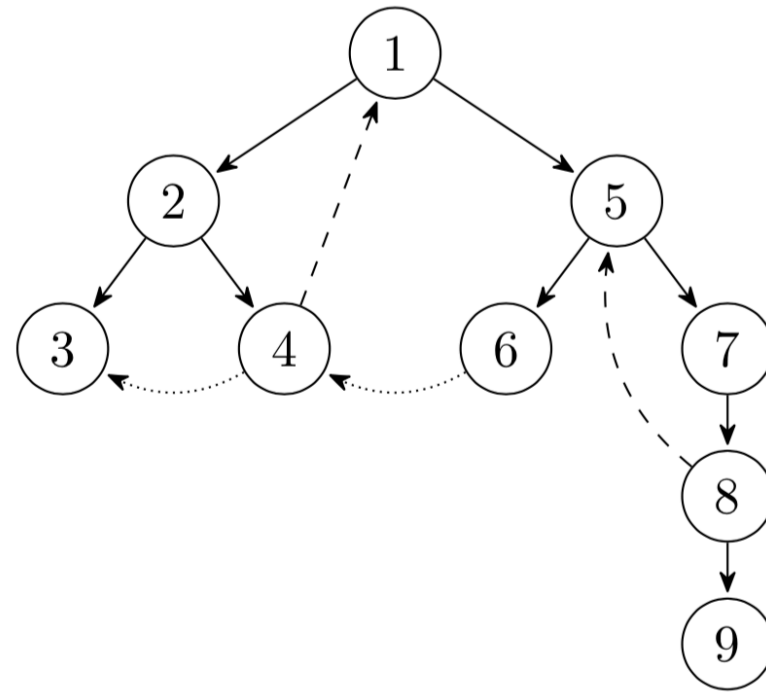
在图论中，一个有向图被成为是强连通的当且仅当每一对不相同结点 u 和 v 间既存在从 u 到 v 的路径也存在从 v 到 u 的路径。有向图的极大强连通子图(这里指点数极大)被称为强连通分量。



比如说这个有向图中，点 1,2,4,5,6,7,8 和相应边组成的子图 就是一个强连通分量，另外点 3,9 单独构成强连通分量。

搜索树

在介绍求解强连通分量的算法之前先来介绍一下搜索树。



有向图的搜索树主要有 4 种边(这张图只有三种)，其中用实线画出来的是树边，每次搜索找到一个还没有访问过的结点的时候这就形成了一条树边。用长虚线画出来的是回边。用短虚线画出来的是横叉边，它主要是在搜索的时候遇到了一个已经访问过的结点，但是这个结点并不是当前节点的祖先时形成的。除此之外，像从 1 到 6 这样的边叫做前向边，它是在搜索的时候遇到子树中的结点的时候形成的。

Tarjan 算法

Tarjan 算法是由 Robert Tarjan 提出的用于寻找有向图的强连通分量的算法。它可以在 $O(n + m)$ 的时间内得出结果。

Tarjan 算法主要是利用 DFS 来寻找强连通分量的。现在我们来看看在 DFS 的过程中强连通分量有什么性质。

很重要的一点是如果结点 u 是某个强连通分量在搜索树中遇到的第一个结点(这通常被称为这个强连通分量的根)。

Tarjan 算法

Tarjan 算法主要是在 DFS 的过程中维护了一些信息:dfn、 low 和一个栈。

- 栈里的元素表示的是当前已经访问过但是没有被归类到任一强连通分量的结点。
- $dfn[u]$ 表示结点 u 在 DFS 中第一次搜索到的次序，通常被叫做时间戳。
- $low[u]$ 稍微有些复杂，它表示从 u 或者以 u 为根的子树中的结点，再通过一条反祖边或者横叉边可以到达的时间戳最小的结点 v 的时间戳，并且要求 v 有一些额外的性质: v 还要能够到达 u 。

可以证明，结点 u 是某个强连通分量的根等价于 $dfn[u]$ 和 $low[u]$ 相等。

Tarjan 算法

当通过 u 搜索到一个新的节点 v 的时候可以有多种情况:

- 结点 u 通过树边到达结点 v
 $\text{low}[u] = \min(\text{low}[u], \text{low}[v])$
- 结点 u 通过反祖边到达结点 v , 或者通过横叉边到达结点 v 并且满足 low 定义中 v 的性质

$$\text{low}[u] = \min(\text{low}[u], \text{dfn}[v])$$

Tarjan 算法

```
int n, m, scc, ind, top;
int bl[N], inq[N], low[N], dfn[N], size[N], q[N];
vector<int> e[N];
void tarjan(int x)
{
    dfn[x] = low[x] = ++ind;
    q[++top] = x; inq[x] = 1;
    for(int i = 0; i < e[x].size(); i++)
        if(!dfn[e[x][i]])
        {
            tarjan(e[x][i]);
            low[x] = min(low[x], low[e[x][i]]);
        }
        else if(inq[e[x][i]])
            low[x] = min(low[x], dfn[e[x][i]]);
    if(low[x] == dfn[x])
    {
        int now = -1;
        scc++;
        while(now != x)
        {
            now = q[top]; top--;
            inq[now] = 0;
            size[scc]++;
            bl[now] = scc;
        }
    }
}
```


无向图的割点与桥

割点：无向连通图中，去掉一个顶点及和它相邻的所有边，图中的连通分量数增加，则该顶点称为割点。

桥（割边）：无向联通图中，去掉一条边，图中的连通分量数增加，则这条边，称为桥或者割边。

割点与桥（割边）的关系：

- 1) 有割点不一定有桥,有桥一定存在割点
- 2) 桥一定是割点依附的边。

无向图的割点与桥

朴素的做法就是枚举删掉一个点或者边，然后判断整张图是否连通。

非根节点 u 是图 G 的割点，当且仅当 u 存在一个子节点 v ，使得 v 及其后代都没有反向边连向 u 的祖先。

即非根节点 u 是图 G 的割点当 u 存在一个子节点 v ，使得 $\text{low}[v] \geq \text{dfn}[u]$ 。

严格大于就是桥的情况。

这里要注意有向图和无向图 Tarjan 算法的区别，有向图 Tarjan 需要 inq 数组来判断一条边是不是反祖边，无向图则不需要，因为横叉边也会成环。

练习题

HAOI2006. 受欢迎的牛

n 头牛， m 条关系，形如 (A, B) ，表示 A 认为 B 受欢迎

如果 A 认为 B 受欢迎， B 认为 C 受欢迎， A 也认为 C 受欢迎

问有多少头牛被认为被所有其它牛欢迎

$n, m \leq 10^5$

HAOI2006. 受欢迎的牛

如果 A 认为 B 受欢迎，A 向 B 连边。

Tarjan 求强连通分量缩点后的图，出度为 0 的点若只有一个则输出其代表的强连通分量的大小，否则无解。

HDU2767. Proving Equivalences

n 个点， m 条边的有向图，问至少添加多少条边，使得整张图强连通。

$n, m \leq 10^5$

HDU2767. Proving Equivalences

答案是 $\max\{\text{入度为0的连通块个数}, \text{出度为0的连通块个数}\}$

特判连通块为1的情况

每个连通块，出度0的点，向其它入度为0的连边，使得形成一个环

中山市选2011. 杀人游戏

n 个人里，有一个人是杀手。 m 个关系 (A, B) ，表示查证 A 的话，他会告诉警察 B 是不是杀手。

查证杀手的话，警察就会死。如果每个人是杀手的概率相同，问最优策略下警察死亡概率？

$n, m \leq 10^5$

中山市选2011. 杀人游戏

查的人越少越好，一个强连通块最多只要查一个人。

如果查了 p 个人，死掉的概率是 p / n 。

发现可以只查入度为 0 的强连通块。

如果存在某个入度为 0 的强连通块只有一个点，并且在新图上它指向的点都有其它点能到达（入度都大于 1），就可以少查一个人。

POI2008. BLO

n 个点， m 条边的无向图，问删掉第 i 个点之后，有多少对点不互相连通？

$n, m \leq 10^5$

POI2008. BLO

把某个割点去掉以后，会出现几个连通块，它们之间不能互相到达

即会分成上面一棵树，下面若干子树

子树之间不互通，所有子树和上面那个树不互通，通过记录树的大小统计答案（两两相乘再相加）

CF1276B. Two Fairs

n 个点， m 条边的无向图，有两个关键点 A ， B ，问有多少个点对 (s, t) ，从 s 走到 t 的时候必须通过 A ， B 。

$n, m \leq 10^5$

CF1276B. Two Fairs

把 A 点从图中删掉，从 B 开始 dfs。

把 B 点从图中删掉，从 A 开始 dfs。

答案是只被 A dfs 到的点的个数乘以只被 B dfs 到的点的个数。

UOJ67. 新年的毒瘤

n 个点， m 条边的无向连通图，删去某一个结点以后整张图会变成一棵树，问有多少个这样的结点。

$n, m \leq 10^5$

UOJ67. 新年的毒瘤

只需找出不是割点，且度数为 $m - (n - 2)$ 的点即可。

BZOJ1589. Trick or Treat on the Farm

n 个点，每个点有一个后继，从第 i 个点出发，会一直往它的后继走，问能走多少个不同的点。

$n \leq 10^5$

BZOJ1589. Trick or Treat on the Farm

每个连通块是基环树。

重建图以后，会变成若干条链，dfs 递推求答案。

HNOI2012. 矿场搭建

矿场是 n 个挖煤点， m 条边的无向图，问至少要在多少个挖煤点设计救援出口，使得当某一个挖煤点坍塌时，其它挖煤点都有道路通向救援出口。

$n \leq 500$

HNOI2012. 矿场搭建

只要考虑坍塌在割点的情况

删去所有割点得到若干连通块，如果图中只有一个连通块，一共只需要设两个

若一个连通块可以到达两个割点的话则不用设置救援出口，否则要设置一个

全都暴力处理，复杂度不超过 $O(n^3)$

SDOI2010. 所驼门王的宝藏

一个 $R * C$ 的网格图，其中有 N 个宫殿，每个宫殿有一个传送门。传送门是 3 种其中的一种，横天门可以传送到同一行的任意一个，纵寰门可以传送到同一列任意一个，自由门可以传送到周围 8 格任意一个（如果有）。

可以从某个宫殿开始寻宝，问最多可以去多少个宫殿？

$N \leq 10^5$, $R, C \leq 10^6$

SDOI2010. 所驼门王的宝藏

每一行找一个横天门，向同一行的横天门连双向边，向同一行其它门连单向边

纵上同理

自由门用 map 判断周围 8 个格子有没有宫殿，有则连边

一个强连通分量可以完全遍历。

Tarjan 缩点以后，在拓扑图中求点权的最长路径，DP。