Tarjan算法

XJTU
Information and Computing Sciences
mg
xjtumg.me
wmg_1007@163.com

- SCC - 有向图强连通分量
- 无向图割点集&割边集
- 双连通分量

- 定义dfn[u]为u在dfs搜索树中被遍历到的次序号

- 定义low[u]为u或u的子树中能通过非父子边追溯到的最早的节点，即dfn[]最小的节点

- 首次到达某点(p)时，dfn与low均为此时的时间
- 每搜索到一个点，将其压进堆栈
- 当点p有与点p’相连时，p’不在栈中，p的low值为两点的low值中较小的一个
- 当点p有与点p’相连时，p’在栈中，p的low值为p的low值和p’的dfn值中较小的一个
- 每当搜索到一个点经过以上操作后（也就是子树已经全部遍历）的low值等于dfn值，则将它以及在它之上的元素弹出栈。这些出栈的元素组成一个强连通分量
- 继续搜索（或许会更换搜索的起点，因为整个有向图可能分为两个不连通的部分），直到所有点被遍历

- 强连通分量一定是有向图的某个深搜树子树
- 可以证明，当一个点既是强连通子图Ⅰ中的点，又是强连通子图Ⅱ中的点，则它是强连通子图Ⅰ∪Ⅱ中的点。
- 这样，我们用low值记录该点所在强连通子图对应的搜索子树的根节点的Dfn值。注意，该子树中的元素在栈中一定是相邻的，且根节点在栈中一定位于所有子树元素的最下方
- 强连通分量是由若干个环组成的。所以，当有环形成时（也就是搜索的下一个点已在栈中），我们将这一条路径的low值统一，即这条路径上的点属于同一个强连通分量
- 如果遍历完整个搜索树后某个点的dfn值等于low值，则它是该搜索子树的根。这时，它以上（包括它自己）一直到栈顶的所有元素组成一个强连通分量

- 时间复杂度 O(V + E)
- 空间复杂度 O(V + E)

- 在一个无向连通图中，如果有一个顶点集合，删除这个顶点集合，以及这个集合中所有顶点相关联的边以后，原图变成多个连通块，就称这个点集为割点集合
- 在一个无向连通图中，如果有一个边集合，删除这个边集合以后，原图变成多个连通块，就称这个点集为割边集合
- 一个图的点连通度的定义为，最小割点集合中的顶点数
- 一个图的边连通度的定义为，最小割边集合中的边数
- 如果一个无向连通图的点/边连通度大于1，则称该图是点/边双连通的(biconnected)，简称双连通或重连通

- 割点u，当且仅当满足(1)或(2)
- (1) u为树根，且u有多于一个子树
- (2) u不为树根，且满足存在(u,v)为树枝边(或称父子边，即u为v在搜索树中的父亲)，使得dfn[u]<=low[v]
- 桥无向边(u,v)，当且仅当(u,v)为树枝边，且满足dfn[u]<low[v]

- POJ 1236
- 给定一个有向图，求：
- 1) 至少要选几个顶点，才能做到从这些顶点出发，可以到达全部顶点
- 2) 至少要加多少条边，才能使得从任何一个顶点出发，都能到达全部顶点

- POJ 3694
- 给定一个连通无向图
- 每次在图上加一条边or查询图上有多少桥
- V, E <= 200000, Q <= 1000