有向图的强连通分量

数据结构：以邻接表存储得到图，数组，栈；

算法：targan算法

tarjan算法基于dfs算法，同一强连通分量内的所有顶点均在同一棵深度优先搜索树中。也就是说强连通分量一定是有向图的某个深度优先搜索生成树。

用low值记录点u所在强连通子图对应的搜索子树的根节点的dfs值。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，所以当有环形成时，我们将这一条路径的low值统一，即这条路径上的所有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。

运行流程：

开始

创建一个邻接表存储的图

定义两个数组dfn和low分别存放遍历到达时间和由该节点可以遍历到的最早祖先的标号

数组初始化：当首次搜索到点u时，dfn和low数组的值都为到该点的时间。

遍历所有节点，换用不同的始点进行DFS

压栈：每遍历到一个未被标记的点，将它入栈。

当点u可以到达点v时，点v不在栈中，那么low[u] = min{low[u],low[v]}；如果点v在栈中，那么low[u] = min{low[u],dfn[v]}。

每当搜索到一个点并经过以上步骤后，其low值等于dfn值，则将它以及在它之上的元素弹出栈。这些出栈元素组成一个强连通分量。