1. **证明：给定有向图G，且点u，点v分别在G的两个强连通分量中，那么u在G中可达v，则u在G的逆向图中不可达v**

证明：因为点u，点v分别在G的两个强连通分量中，且u在G中可达v。即u、v所在的两个强连通分量联通。所以在G中任意的u、v互达。即G中包含了所有u、v互达的路径。则在G的逆向图中，不存在任意一条u、v可互达的路径。即u在G的逆向图中不可达v。

1. **证明：给定有向图G，且点u，点v均在G的某个强连通分量中，那么u在G中可达v，则u在G的逆向图中也可达v**

证明：因为u、v均在G的某个强连通分量中，则u在G中必定可达v，同时v在G中必定可达u。即u、v在G中必定同在一个环上。则u、v在G的逆向图上同样必定同在一个环上。即。u在G的逆向图中也可达v。

**3. 基于1、2给出Korasaju算法的正确性证明  
提示：利用强连通图的性质定理**

证明：首先用反证法证明“每个和s强连通的顶点v都会在构造函数调用的dfs(G,s)中被访问到”。假设有一个和s强连通的顶点V不会在构造函数调用的dfs(G,s)中被访问到。因为存在从s到v的路径，所以v肯定在之前就已经被标记过了。但是，因为也存在从v到s的路径，在dfs(G,v)调用中s肯定会被标记，因此构造函数应该是不会调用dfs(G,s)的。矛盾。

其次证明：“构造函数dfs(G，s)所到达的任意顶点V都必然是和S强联通的。”

在G中的dfs(s)，假设有s可达v且dfs(v)发生在dfs(s)调用过程中，即dfs(s)直接或间接调用了dfs(v)。此时在遍历顺序中，v应该排在s之后。反证法证明：如果v排在s之前，在dfs(s)遍历到v时，由于dfs(v)在之前已执行过，即marked[v]已经为true，所以dfs(s)将不会直接或间接调用dfs(v)。  
由于在G中某个强连通图中存在一条s->v的路径，所以也应存在一条v->s的路径。即在G(R)中，应存在一条s->v的路径。根据上面第一点，由于v排在s之后，所以G(R)中求reversePost时dfs(v)应该在dfs(s)结束前结束(因为reversePost是在每个dfs遍历结束后将元素入栈的，入栈时该dfs结束，先入栈的节点排在后面，即先进后出)，所以dfs(v)在dfs(s)结束之前结束。  
由于在G中存在一条路径s->v，所以在G(R)中存在一条路径v->s。因此在dfs(v)发生的时候，dfs(v)将会检查s是否被遍历过，如果未被遍历过则触发dfs(s)。但是如果dfs(v)触发了dfs(s)，则s将排在v之后，与v排在s之后矛盾。由此可知dfs(v)没有触发dfs(s)。  
由于dfs(v)没有触发dfs(s)，且在dfs(s)结束之前结束。即在dfs(v)发生时，marked[s]已经被标为true了，但是s还未被push到reversePost中。所以在dfs(v)进行时，dfs(s)已发生但还未结束(发生时marked对应下标被标为true,结束时reversePost对应节点入栈)。即dfs(v)发生在dfs(s)的调用过程中。因此dfs(s)肯定直接或间接调用了dfs(v)。即从s存在一条路径到v。即G(R)中存在一条路径s->v。

1190201411蒋婷婷