

# 算法exercise08

homework

## 1. 证明G的凝聚图是有向无环图

实在不会，这一问放弃了。

## 2. G邻接表存储状态下的转置

```
1.  #define maxn 100 //最大顶点个数
2.  int n, m; //顶点数, 边数
3.
4.  struct arcnode //边结点
5.  {
6.      int vertex; //与表头结点相邻的顶点编号
7.      int weight = 0; //连接两顶点的边的权值
8.      arcnode * next; //指向下一相邻接点
9.  };
10.
11. struct vernode //顶点结点, 为每一条邻接表的表头结点
12. {
13.     int vex; //当前定点编号
14.     arcnode * firarc; //与该顶点相连的第一个顶点组成的边
15. }Ver[maxn];
16.
17. struct Graph
18. {
19.     vernode* Adj[maxn];
20.     int n;
21.     int m;
22. };
23.
24. //插入边
25. void insertEdge(Graph g, int start, int end)
26. {
27.     vernode* vn = new vernode();
28.     vn->vex = end;
29.     if (g.Adj[start]->firarc == NULL)
30.     {
31.         vernode* sn = new vernode();
32.         sn->vex = start;
```

```
33.         g.Adj[start] = sn;
34.     }
35.     vernode* temp = g.Adj[start];
36.     while(temp->firarc)    temp = temp->firarc;
37.     temp->firarc = vn;
38. }
39.
40. void transpose(Graph origin, Graph des, int n)
41. {
42.     if ( (origin == NULL) || (des == NULL)) return;
43.     for (int i = 0; i < n; ++i)
44.     {
45.         vernode* vn = origin.Adj[i];
46.         while(vn)    insertEdge(des, vn->vex, i);
47.     }
48. }
```