对包含 **n** 个顶点的 **AOV** 网进行拓扑排序

```
void Graph_List:: TopoOrder( )
{
            int n = graphsize;
           int* count = new int[n];
           //计算 count 数组
          for( int i=0 ; i<n ; i++ ) count[i] = 0;
          for( i=0 ; i<n ; i++ )
              {
                 Edge* p = Head[i].adjacent;
                while (p!=NULL)
                 {
                       count[p→VerAdj]++;
                       p = p→link;
                 }
              }
         int top = - 1 ;                                          //  初始化“栈顶指针”
         for( int i = 0 ; i < n ; i ++ )                               // (1)  将入度为 0 的顶点入栈
                 if( count[i] = = 0 )
                     { count[i] = top ; top = i ; }
         for( int i = 0 ; i < n ; i ++ )                             // (*) AOV 网中最多有 n 个顶点
                 //  若循环体尚未被执行 n 次，栈顶指针已为-1，说明有回路，终止程序
                 if( top = = - 1 )
                      { cout << " There is a cycle in network ! " << endl ; return ; }
                 else
                 {
                       int j = top ; top = count[top] ;//  从栈中弹出一个顶点 j
                       cout << j << endl ;                        //  输出该顶点
                       Edge *p = Head[j].adjacent ;//  令 p 为 j 的边链表头指针
                       while( p ! = NULL )                        // (2)  从当前的图中删除与 j 关联的边
                       {
                             int k = p→ VerAdj ;              // k 为边<j,k>的终点
                             // k 的入度减 1，若入度为 0，则 k 入栈
                             if( --count[k] = = 0 ) { count[k] = top ; top = k ; }
                             p = p→ link ;
                       }
                 }
         delete[] count;
}
```