计算连通分量的算法

```
void Graph_List::Find_all_Component()
{
	int n=graphsize;
	int *markedList=new int[n];   //记录节点是否加入到连通分量
	int **WSM;
	WSM=Warshall( ); //求图的可及矩阵
	//int *scList=new int [graphsize];
	struct Node{
		int Vername;
		Node * link;
	};
	for(int i=0;i<n;i++)//初始化，
	{
		markedList[i]=0;
	}
	int t=0;
	for(int i=0;i<n;i++)
	{
		if(markedList[i]==0)
		{
			Node *scList=new Node; //一条连通分量链表的头结点
			t=t+1;
			markedList[i]=1;
			scList->Vername=i;
			scList->link=NULL;
			Node *q=new Node;
			q=scList;
			for(int j=0;j<n;j++)
			{
				if(i!=j&&WSM[i][j]==1&&WSM[j][i]==1)
				{
					markedList[j]=1;
					Node *p=new Node;
					p->Vername=j;
					p->link=NULL;
					q->link=p;
					q=p;
				}
			}
			//q=scList;
			cout<<"第"<<t<<"个连通分量：";
			while(scList!=NULL){
				cout<<scList->Vername<<" ";
				q=scList;
				scList=scList->link;
				delete q;
			}
			cout<<endl;
		}	}}
```