

计算可及矩阵的算法

```
int **Graph_List::Warshall()
{
    int n=graphsize;
    int **WSM=new int*[n]; //二维数据存储删除的边，删除的边为1，没删除的边为1
    Edge* p = new Edge;
    for(int j=0;j<n;j++) //二维数组初始化
    {
        WSM[j]=new int[n];
        p=Head[j].adjacent;
        for(int k=0;k<n;k++)
        {
            if(j==k){
                WSM[j][k]=1;
            }else{
                WSM[j][k]=0;
            }
        }
        while(p!=NULL)
        {
            WSM[j][p->VerAdj]=1;
            p=p->link;
        }
    }
    for(int k=0;k<n;k++)
    {
        for(int i=0;i<n;i++)
        {
            if(WSM[i][k]==1)
            {
                for(int j=0;j<n;j++)
                {
                    WSM[i][j]=(WSM[i][j]||WSM[k][j]);
                }
            }
        }
    }
    cout<<"可及矩阵是: "<<endl;
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            cout<<WSM[i][j]<<' ';
        }
        cout<<endl;
    }
    delete p;
    return WSM;
}
```