

Floyd求最小环

2010-11-17 16:07:51 | 分类： 图论 | 标签： 最小环

FLOFTER

订阅 | 写字 | 相册

我的照片书 |

关于floyd求最小环的一点心得

//Floyd 的 改进写法可以解决最小环问题，时间复杂度依然是 $O(n^3)$ ，储存结构也是邻接矩阵

```
int mincircle = infinity;
Dist = Graph;

for(int k=0;k<nVertex;++k){
    //新增部分:
    for(int i=0;i<k;++i)
        for(int j=0;j<i;++j)
            mincircle = min(mincircle,Dist[i][j]+Graph[j][k]+Graph[k][i]);
    //通常的 floyd 部分:
    for(int i=0;i<nVertex;++i)
        for(int j=0;j<i;++j){
            int temp = Dist[i][k] + Dist[k][j];
            if(temp < Dist[i][j])
                Dist[i][j] = Dist[j][i] = temp;
        }
}
```

以上为网上流传的Floyd求最小环的主代码。我们发现，最下面两重循环就是Floyd原来的代码，新增的就是上面那个判环部分。一开始，我不明白，为什么要把新增的放在前面，两者的顺序能不能调换？现在的理解是这样的：在第k层循环，我们要找的是最大结点为k的环，而此时Dist数组存放的是k-1层循环结束时的经过k-1结点的最短路径，也就是说以上求出的最短路径是不经过k点的，这就刚好符合我们的要求。为什么呢？假设环中结点i,j是与k直接相连，如果先求出经过k的最短路径，那么会有这样一种情况，即：i到j的最短路径经过k。这样的话就形成不了环，显然是错误的。当时还有一个问题，就是为什么要多开一个Dist数组呢，一个Graph不是足够了吗？其实好好想想，出现的问题和前面是一个道理。如果只开Graph，那么它里面的值就会不断改变，也会存在路径覆盖的情况，导致形成不了环或不是最小环。举个例子：假设现在进行第k层循环，i,j为枚举出来与k直接相连的边。由于此时Graph是动态的，原来根本不存在i到k的一条边，现在可能经过其它结点形成了“边”，但它未必是与k直接相连的边。以上两个问题花了我半天时间来弄懂，由于网上也没有找到关于这些问题的（可能我比较笨吧），所以要写这些东西，但又写得挺乱……