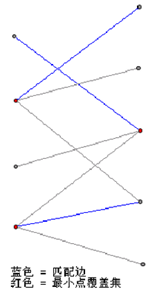
[【最小边覆盖 & 最小路径覆盖 & 最小顶点覆盖 & 最大独立集 & 最大团】](http://blog.csdn.net/leolin_/article/details/7199688)

**最小边覆盖 = 最大独立集 = |V| - 最大匹配数**  
这个是在原图是二分图上进行的  
最小路径覆盖和最小边覆盖不同，不要求给的图是二分图，而是要求是N x N的有向图，不能有环，然后根据原图构造二分图，构造方法是将点一分为二，如，i分为i1和i2然后如果i和j有边，那么就在i1和j2之间连一条边。由此构成二分图

然后**最小路径覆盖 = n-m，n为原图的点的个数，m为新造二分图的最大匹配**。证明也是特别简单的，根据定义最小路径覆盖里要求同一个点只可以属于一条路径，即路径时不可以开叉的，如果在二分图里选两条有公共点的边那么反应在原图上就是路径有岔路了，所以二分图里选的边必须是无公共交点的，这就是转化到最大匹配了。

**总结：**

**【无向图的最大独立数】: 从V个顶点中选出k个顶，使得这k个顶互不相邻。 那么最大的k就是这个图的最大独立数。**

**【无向图的最大团】:  从V个顶点选出k个顶，使得这k个顶构成一个完全图，即该子图任意两个顶都有直接的边。**

**【最小路径覆盖(原图不一定是二分图，但必须是有向图，拆点构造二分图)】：在图中找一些路径，使之覆盖了图中的所有顶点，且任何一个顶点有且只有一条路径与之关联。最小路径覆盖 = |V| - 最大匹配数**

**【最小边覆盖(原图是二分图)】：在图中找一些边，使之覆盖了图中所有顶点，且任何一个顶点有且只有一条边与之关联。最小边覆盖 = 最大独立集 = |V| - 最大匹配数**

**【最小顶点覆盖】：最大匹配。用最少的点（左右两边集合的点）让每条边都至少和其中一个点关联。**

PS: 原来学二分匹配时就整理过这些数字，他们之间关系是很多。如: 最小边覆盖数+最大独立数 = 顶点数。 虽然求出他们都是np问题。但对于特殊的图还是有好的算法的，如:

在二分图中，最小点覆盖数等于最大匹配数，而最大独立数又等于顶点数减去最小覆盖数(=最大匹配数)，所以可以利用匈牙利求出最大独立数等等。

a.点覆盖集：无向图G的一个点集，使得该图中所有边都至少有一点端点在该集合内。  
b.点独立集：无向图G的一个点集，使得任两个在该集合中的点在原图中不相邻。  
c.最小点覆盖集：无向图G中点数最少的点覆盖集  
d.最大点独立集：无向图G中，点数最多的点独立集  
e.最小点权覆盖集：带点权的无向图中，点权之和最小的点覆盖集  
f.最大点权独立集：实在带点权无向图中，点权之和最大的点独立集

**性质：**

**最大团 = 补图的最大独立集**

**最小边覆盖 = 二分图最大独立集 = |V| - 最小路径覆盖**

**最小路径覆盖 = |V| - 最大匹配数**

**最小顶点覆盖 = 最大匹配数**

**最小顶点覆盖 + 最大独立数 = |V|**

最小割 = 最小点权覆盖集 = 点权和 - 最大点权独立集

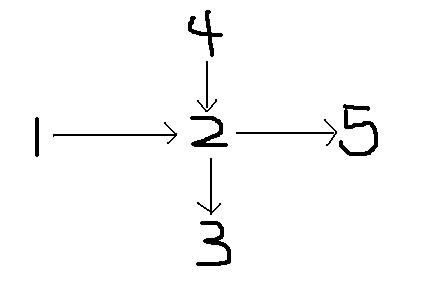
**[最小路径覆盖问题值得注意的地方](http://www.cnblogs.com/ka200812/archive/2011/07/31/2122641.html)**

首先，最小路径覆盖=总节点数-最大匹配数。这个应该已经是路人皆知了。

所谓最小路径覆盖，是指在一个有向图中，找出最少的几条路径，用它们来覆盖全图

这里说的值得注意的地方，如果有向图的边有相交的情况，那么就不能简单的对原图求二分匹配了

举个例子，假设有图：1->2     2->5     2->3      4->2，事实上，这其实就是两条边：1->5  4->3 ，节点2只是他们的一个交点



如果只是简单的在原图的基础上求二分匹配，那么得到的匹配答案是2，最小路径覆盖答案便是5-2=3。

可是随便一看都能看看出端倪，这个图中，只需要两个点便可以探索完整个地图，这里最小路径覆盖数明显是2。

问题究竟出在哪里呢？其实就和这个交点2有关。既然边有相交，那么他们的连通性也应该连通下去。

解决的办法是对原图进行一次闭包传递(也就是flody)，于是便增加了四条边:1->3   1->5   4->3  4->5

这时再求最大匹配数，匹配答案便是3，最小路径覆盖值为2，这是正确答案！

完全图：若一个图的每一对不同顶点恰有一条边相连，则称为完全图。完全图是每对顶点之间都恰连有一条边的[简单图](http://baike.baidu.com/view/93116.htm" \t "_blank)。n个端点的完全图有n个端点及n(n − 1) / 2条边，以Kn表示。它是(k − 1)-[正则图](http://baike.baidu.com/view/566158.htm" \t "_blank)。所有完全图都是它本身的团（[clique](http://baike.baidu.com/view/3174641.htm" \t "_blank)）

# [最大团](http://www.cnblogs.com/pushing-my-way/archive/2012/08/08/2627993.html)

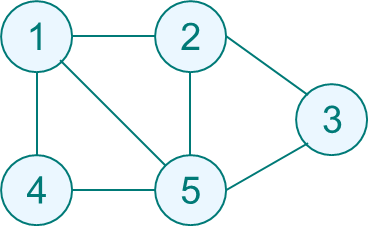
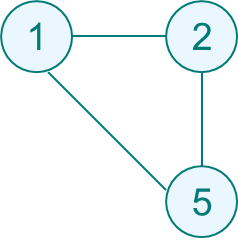
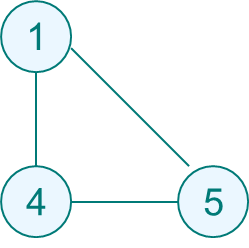
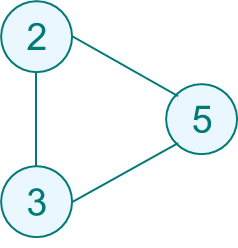
问题描述：团就是最大完全子图。

给定无向图G=(V,E)。如果Uhttp://pic002.cnblogs.com/images/2012/414141/2012080811500571.pngV，且对任意u，vhttp://pic002.cnblogs.com/images/2012/414141/2012080811500571.pngU 有(u，v) http://pic002.cnblogs.com/images/2012/414141/2012080811500571.png E，则称U 是G 的完全子图。

G 的完全子图U是G的团当且仅当U不包含在G 的更大的完全子图中，即U就是最大完全子图。

G 的最大团是指G中所含顶点数最多的团。

例如：

                 (a)                                        (b)                             (c)                            (d)

图a是一个无向图，图b、c、d都是图a的团，且都是最大团。