## 染色法判断二分图

如果存在奇数环，则该图不是二分图。如果不存在奇数环，则该图是二分图。

bool dfs(int u, int c)

{

    color[u] = c; // 当前点先染色

    for (int i = h[u]; ~i; i = ne[i])

    {

        int j = e[i]; // 对于这个点连接的所有的点

        if (color[j])

        { // 如果已经被染过色了

            if (color[j] == c)

                return false;

            // 就需要判断一下，如果两点颜色一样，染色就冲突了

        }

        else if (!dfs(j, 3 - c))

            return false;

        // 否则dfs去染下一个结点，赋予的颜色肯定要跟 c 不一样

        // 3 - 1 == 2，3 - 2 == 1

        // 同时传回染色成功与否的信息

    }

    return true;

}

bool check()

{

    memset(color, 0, sizeof color); // 0 —— 未染色，1 —— 黑色，2 —— 白色

    for (int i = 1; i <= n; i++)

        if (color[i] == 0) // 一旦某个点没染过色，dfs去染色

            if (!dfs(i, 1))

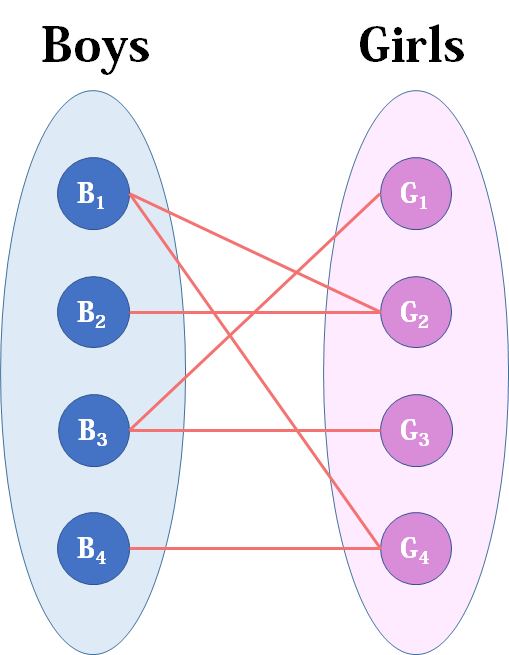
                return false; // 如果传回false显然失败，此图不是二分图

    return true;

    // 否则true

}

## 匈牙利算法求最大匹配数



**最大匹配数：假如你是红娘，可以撮合任何一对有暧昧关系的男女，那么你最多能成全多少对情侣**？（数学表述：在二分图中最多能找到多少条没有公共端点的边）

## 最小点覆盖问题

我们想找到**最少**的一些**点**，使二分图所有的边都**至少有一个端点**在这些点之中。倒过来说就是，删除包含这些点的边，可以删掉所有边。

**最小点数和最大匹配数相等证明：**

最小覆盖点n和最大匹配数m的关系：  
因为有m条实线边，最少要删除m个点才能删除这些实线，此时n＝m。  
ⅰ至于虚线，假设虚线的两个端点均没有连接实线，那么这两个端点就应该配对(连接成实线)，这显然和匈牙利算法不符，故假设不成立，所以虚线至少有一个端点已经连接了实线。  
ⅱ接下来讨论两个端点只有一个端点连接了实线(lineAB)的虚线，删除lineAB和这条虚线的交点(pA)即可(这里没有开销，算作删除实线时的端点的选择条件)，不用担心删除pA点后还要删除pB点导致开销＞m，这里讨论pB的所有情况：  
①pB没有任何虚线连接，此时删除pA即可删除pB的所有连线，pB没有删除的必要。  
②pB有虚线相连，且这条虚线另一端没有连接实线，那我们就又发现了一条增广路径，和匈牙利算法不符，故不成立。  
③pB有虚线连接，且虚线的另一端也有实线连接。那么问题递归，可以把pB当做没有实线连接的点(实线随着pA的删除同步删除了)，又回归到ⅱ。  
ⅲ虚线的两端均有实线连接，那么此时只要实线任一端点被删除，虚线均被删除，没有任何开销。  
综上所述，删除所有连线，只需要删除实线的端点，端点的选择受ⅱ①约束，故开销为m。

原博客：<https://zhuanlan.zhihu.com/p/96229700>