二部图基础

Assign	
Status	
create	
source	https://blog.nowcoder.net/n/0e9a713d93f54bc79739588e928f091a
source2	https://zhuanlan.zhihu.com/p/270333968

基于这篇博客讲解二部图的一些基础知识，以及常用算法

什么是二部图

二部图定义：

设$G=(V,E)$是一个无向图，如果顶点V可分割为两个相互不相交的子集(A,B)，且图中每条边(i,j)所关联的两个顶点i和j分别属于这两个不同的顶点集$(i\in A, j\in B)$，则称图G为一个二部图。

简单来说，就是就是一个顶点集V可以分为两个完全不相交的子集，然后对于图中任一条边的两个顶点分别属于这两个子集。但是子集内部没有边。

二部图匹配：

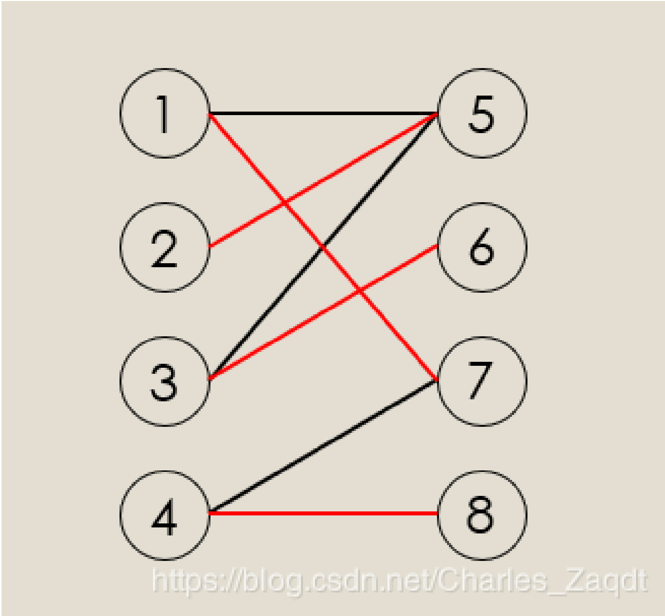
匹配：在二分图G的子图M的边集中，任意两条边都不依附于同一顶点，则称M是一个匹配

极大匹配：在当前已完成匹配的情况下，无法通过增加未完成匹配的边的当时来增加匹配的边数

最大匹配：所有极大匹配当中边数最大的一个匹配

完全匹配：图中的每个顶点都和图中某条边相关联，称为完全匹配（完备匹配）

完全匹配一定是极大匹配，但是极大匹配不一定是完全匹配。



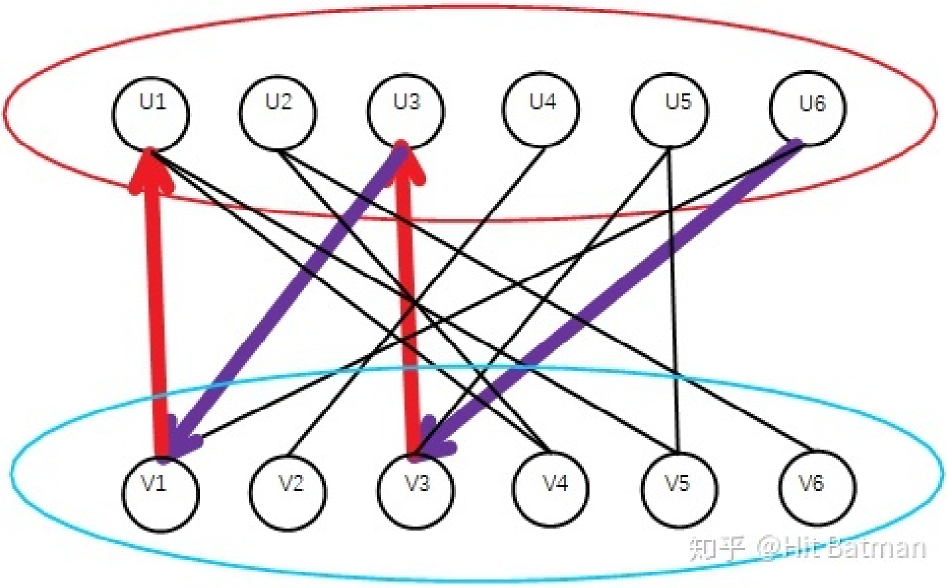
上图就是一个完全匹配

这里有一个点，既这些边都应该是客观存在的，而不是我们构造出来的。基于此，我们去寻找最大匹配的边的合集（用tuple表示）。由此引出匈牙利算法以及一些应用场景。

匈牙利算法

在了解匈牙利算法之前，需要给出两个路径的定义：

交错路径：给定图G的某一个匹配M，如果一条路径的边交替出现在匹配M中和不在M中，则称这条路径为交错路径。



如上图所示$\{(u_6,v_3),(v_3,u_3),(u_3,v_1),(v_1,u_1)\}$是一条路径，$\{(u_1,v_1),(u_3,v_3)\}$是一个匹配$M$。

这时我们发现在路径中$(u_6,v_3),(u_3,v_1)$这两个点不在匹配当中。因此这条路径就被成为交错路径。

增广路径：当一条交错路径的起始和终点都不在匹配边的顶上时，就是一条增广路径

有了这两条路径的定义，我们就可以理解匈牙利算法的思路：

首先，设定一个目标。

假设我们本次的目标是给集合U中的点尽可能找到匹配

Step 1：找到一个匹配M（可以为空）。依据这个匹配可以得到集合U中为匹配的点的集合S

Step 2：从S中取出一个未匹配的点，从这个点开始寻找增广路径，得到新的增广路径P

Step 3：更新匹配M。将M和路径P求异或，更新得到新的匹配M_1

Step 4：以M_1为**Step 1**中M，重复**Step 2**和**Step 3**，直至将为匹配的点S全部取完，则可以得到此二部图的最大匹配

KM算法（Kuhn-Munkres Algorithm）

由匈牙利算法的推导过程我们可以看到，每个人找到的最大匹配都有可能不一样（最大匹配不唯一）。当图中没有权重时这样差异不会造成什么影响，但是在有权图中，这种差异会使匈牙利算法达到局部最优而不是全局最优。

KM算法可以用贪心算法的思路解决这一问题

首先定义**相等子图：**

每一个顶点有一个顶标，选择边权等于两断电的顶标之和的边，组成的图称为相等子图

相等子图的性质：

1.在任意时刻，相等子图上的最大权匹配一定小于等于相等子图的顶标和

2.在任意时刻，相等子图的顶标即为所有顶点的顶标和

3.扩充相等子图后，相等子图的顶标和会减小

4.当相等子图的最大匹配为原图的完备匹配时，匹配边的权值和等于所有顶点的顶标和，此匹配为最佳匹配

Step 1：选择顶点数较少的为X部，初始对X部的每一个顶点设置顶标，顶标的值为该点关联的最大权值，Y部的顶点的顶标为0

Step 2：对于X部中每个顶点，在相等子图种利用匈牙利算法找到一条增广路径。如果没有找到，则修改顶标，扩大相等子图，继续找增广路径。当每个点都找到增广路径时，我们就找到了二部图的完备匹配。这一完备匹配为二部图的最佳匹配。

修改顶标的策略：

如果没有找到增广路径，但是会有交错路径，多条交错路径可以组成一个交错树。将交错树中X部的顶点顶标减去一个值d，交错树种属于Y部的顶点顶标加上一个值d（需要计算）

之后我们会发现：

- 两端都在交错树中的边(i,j)，其顶标和没有变化。既原来属于相等子图，现在仍属于相等子图
- 两端都不在交错树中的边(i,j)，其顶标也没有变化。既原来不属于相等子图，现在仍不属于相等子图
- X端不在交错树中，Y端在交错树中的边(i,j)，顶标和会增大。也就是原来不属于相等子图现在仍然不是相等子图。
- X端在交错树中，Y端不在交错树中的边(i,j)，顶标和会减小。既原来不属于相等子图，现在可能进入了相等子图，相等子图会扩大。

如何计算d值：

- 为了保证至少有一条边进入相等子图，可以在交错树的变种寻找顶标和边权之差最小的边，也就是d值。