图的匹配及其应用

清华大学 何昊天

kiana810@126.com

主要内容

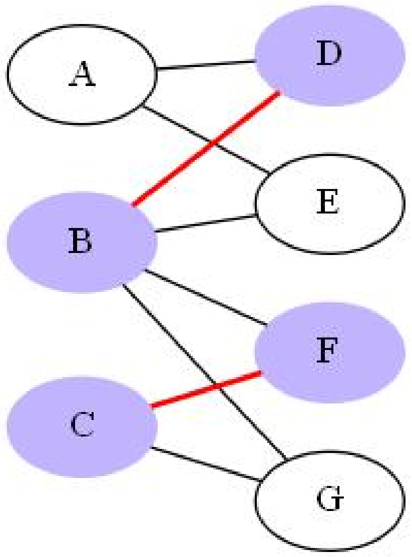
- **Theorem Proving**
- 首先会对相关内容和算法进行简单讲解，同时对用到的各种结论进行证明，最后略讲一些具体建模的应用
- 本节课内容非常基础，祝大家午睡愉快~

一些定义

- 对于一张无向图G=(V,E)，一个**匹配(matching)**M是边集的一个子集，且满足M中不存在两条边共用同一个节点
- 在G的所有匹配中，元素数量最多的匹配就称为G的**最大匹配**
- 图G被称为一张**二分图(bipartite graph)**，如果点集V可以被划分为两个集合$V=V_1\cup V_2$，且满足V_1与V_2的交集为空
- 简单连通图我们称作**一般图**，今天我们主要讨论的就是二分图与一般图的最大匹配问题及其应用

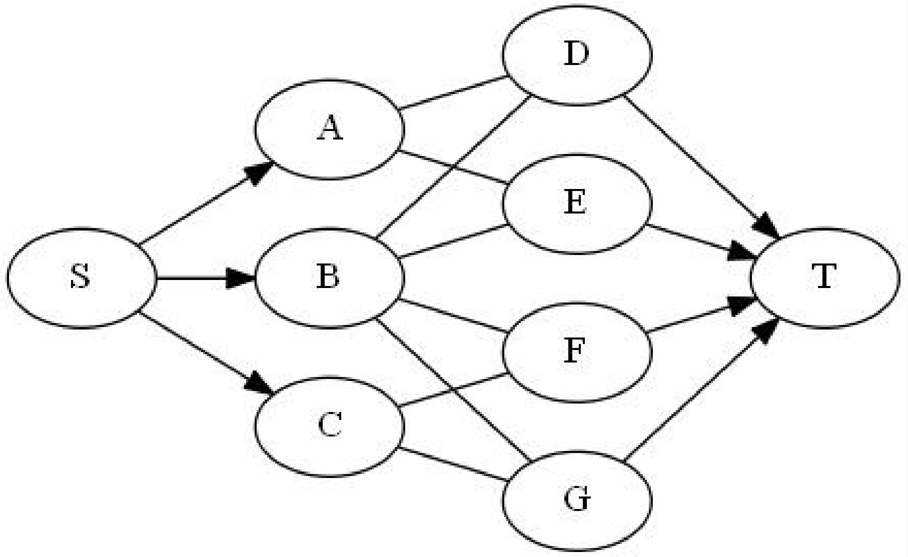
一些约定

- 在匹配中的边称为<u>**匹配边**</u>，如下图边<B,D>和<C,F>，其余边称为<u>**非匹配边**</u>
- 和匹配边相邻的节点称为<u>**匹配点**</u>，如下图节点B、C、D和F，其余节点称为<u>**未匹配点**</u>



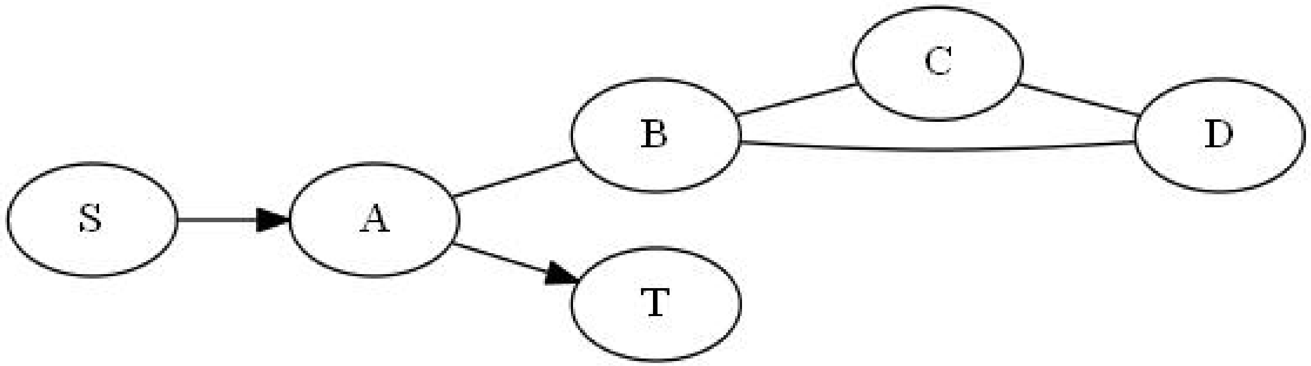
匹配与网络流

- 二分图最大匹配可以使用网络流来解决
- 我们建虚拟源点向一个点集中的点连边，另一个点集中的点向虚拟汇点连边，原图中有的边保持不动，所有边流量均为1，则此时新图的最大流的值就是原图最大匹配数



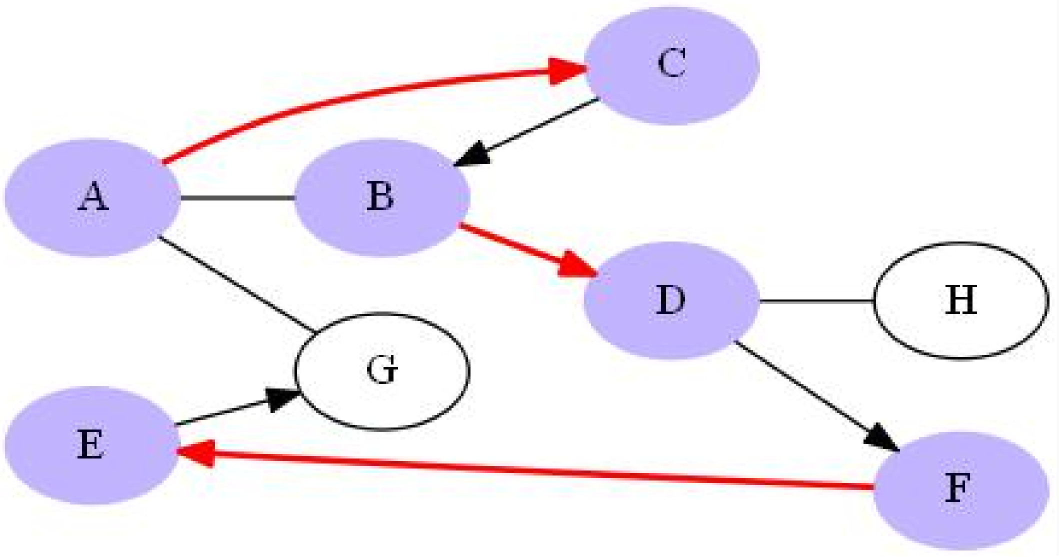
匹配与网络流

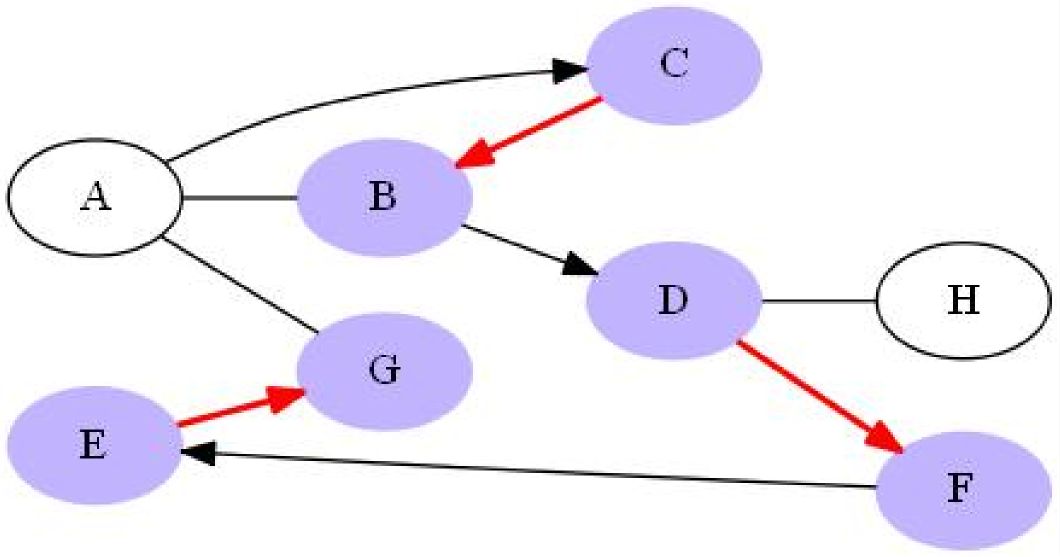
- 我们可能会想到将这个方法推广到一般图上，但由于一般图存在**奇环**的缘故，这样做答案是错误的
- 我们可能还想过将一般图匹配建模成网络流的其它方式，遗憾的是这些方法都不能奏效



交错路

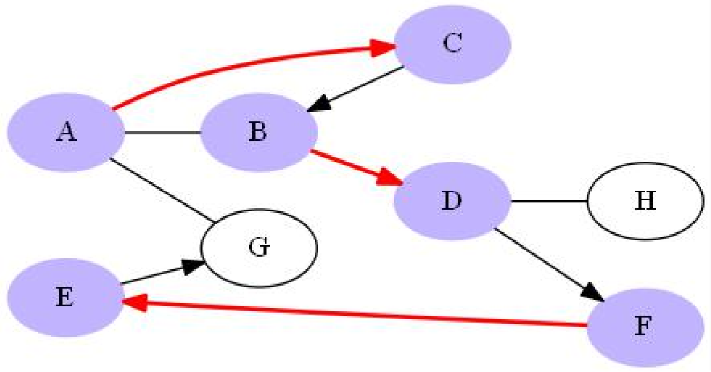
- 给出一张图和一个匹配，匹配边和非匹配边彼此相间的路径称为**交错路**
- 交错路有一个重要的性质：将其中的匹配边改为非匹配边，非匹配边改为匹配边，仍旧会得到一个合法的匹配





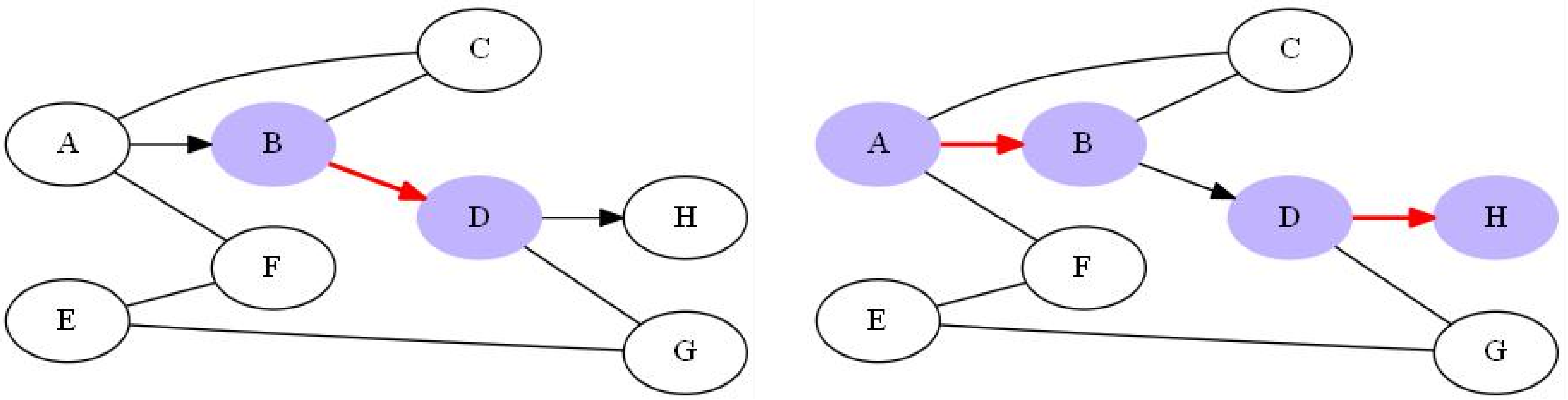
交错路性质证明

- 对于不在交错路上的边和点，显然没有影响
- 对于交错路上不是端点的节点，其恰好和一条匹配边相邻，取反后该性质仍然保留，故这些节点不会产生矛盾
- 对于交错路的端点，原先的匹配点变为未匹配点不会产生矛盾，原先的未匹配点一定不和匹配边相邻，故也不会产生矛盾，性质得证



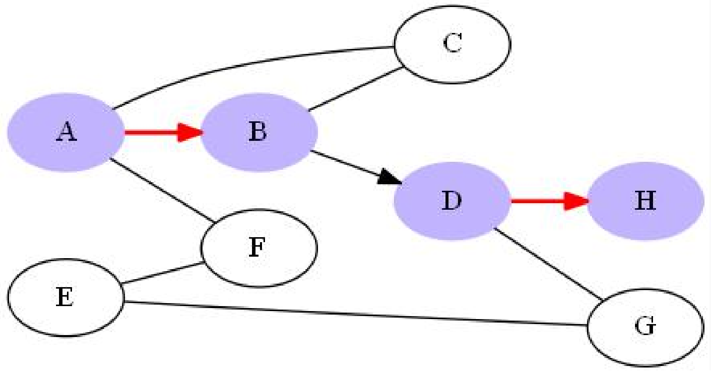
增广路

- 两个端点均为未匹配点的交错路称为**增广路(Augmenting Path)**
- 增广路有一个更重要的性质：将其中的匹配边改为非匹配边，非匹配边改为匹配边，仍旧会得到一个合法的匹配，且匹配数恰好增加1



增广路性质证明

- 匹配仍旧合法已证
- 我们通过匹配点的数量来考虑证明，在取反之前，除去增广路端点外的所有点都是匹配点，取反之后这一点没有改变，且增广路的端点也成为了匹配点，故匹配点数量增加了2，所以匹配数增加了1
- 这里用到了匹配数等于匹配点数量的一半，但这是显然的



进一步的想法

- 既然每次取反增广路会使得匹配数量加1，那么我们可以不断通过寻找并取反增广路来获得极大匹配（这一点可能不是显然的，但我们不进行证明）
- 现在的问题是，这个极大匹配是不是最大匹配？
- 答案是肯定的，这依赖于下面的定理：
- **增广路定理(Augmenting Path Theorem)**：给定一张图和一个匹配，如果图上不存在增广路，当且仅当该匹配为图的最大匹配
- 最大匹配的图上没有增广路是显然的，接下来我们给出这个定理另一部分的纯图论证明，并由此引申出一个可操作的算法

增广路引理的证明

- **增广路引理**：给定一张图和一个匹配，如果从一个未匹配点出发找不到增广路，则一定存在一个最大匹配，使得在该匹配中这个点是未匹配点，换句话说，我们删去这个点仍然能找到原图的一个最大匹配
- （这个名字是我起的，因为这个引理足够重要，所以需要一个名字）
- 证明：假设给定匹配为M，且存在一未匹配点u，使得从u出发找不到增广路，该图有一个最大匹配为M'
- 假设u在M'中不是匹配点，则引理得证，接下来我们考虑u是M'中匹配点的情况，并构造出另一个最大匹配使得u不是匹配点

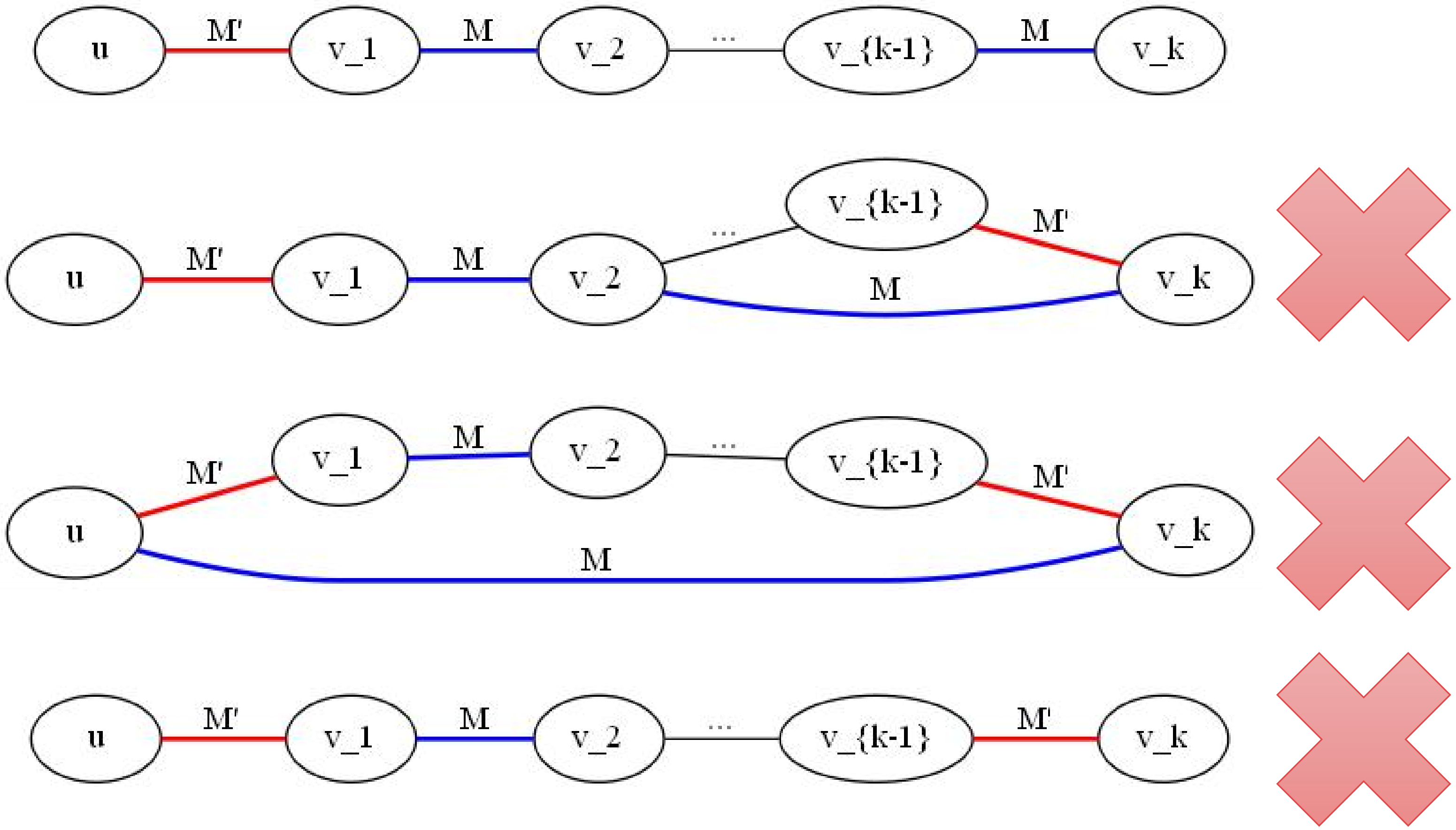
增广路引理的证明

- 考虑M和M'的对称差（$M \triangle M' = (M \cup M') - (M \cap M')$），这不一定是匹配，但其中仅有M中的匹配边和M'中的匹配边以及无关的边
- 由于u在M中不是匹配点，所以u在$M \triangle M'$中仍然保留了在M'中的匹配边，我们假设匹配边的另一端是v_1
- 显然v_1不可能与其它M'中的匹配边相邻，但是可能与M中的匹配边相邻，假如有这样的边，我们假设匹配边的另一端为v_2
- 依次类推，直到走到一个点v_k，使得v_k除了与v_{k-1}通过匹配边相连外，其余邻边都是无关的边

增广路引理的证明

- 显然刚才的路径不可能重复经过一个节点v_i两次（否则v_i在某个匹配中一定有两条匹配边，但这是不可能的）
- 其次，$v_k \neq u$，否则u不是M中的未匹配点
- 最后，v_k与v_{k-1}之间的边一定是M中的匹配边，否则从u到v_k就有一条M中的增广路，这与假设矛盾
- 现在，我们在M'中将从u到v_k的路径取反，注意这是一条交错路，所以我们会得到一个新的匹配，且匹配数没有改变，所以这是一个新的最大匹配M''，但是u不是M''中的匹配点，证毕

增广路引理图示



增广路定理的证明

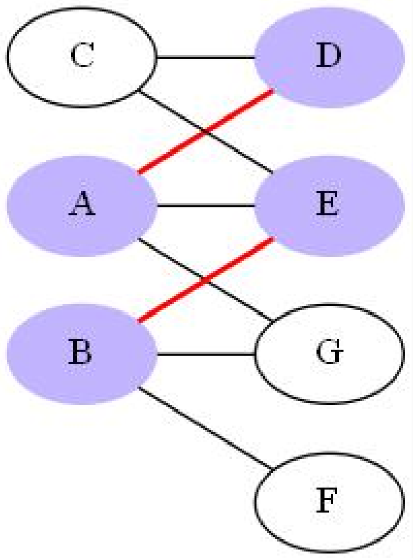
- **增广路定理(Augmenting Path Theorem)**：给定一张图和一个匹配，如果图上不存在增广路，当且仅当该匹配为图的最大匹配
- **增广路引理**：给定一张图和一个匹配，如果从一个未匹配点出发找不到增广路，我们删去这个点仍然能找到原图的一个最大匹配
- 证明：假如图中不存在增广路，则从任意未匹配点出发都找不到增广路
- 我们任选一个未匹配点，将其删去后我们依然能得到原图的最大匹配，但删完点之后原匹配仍然是匹配
- 递归使用增广路定理，直到图中只剩匹配点，增广路定理得证

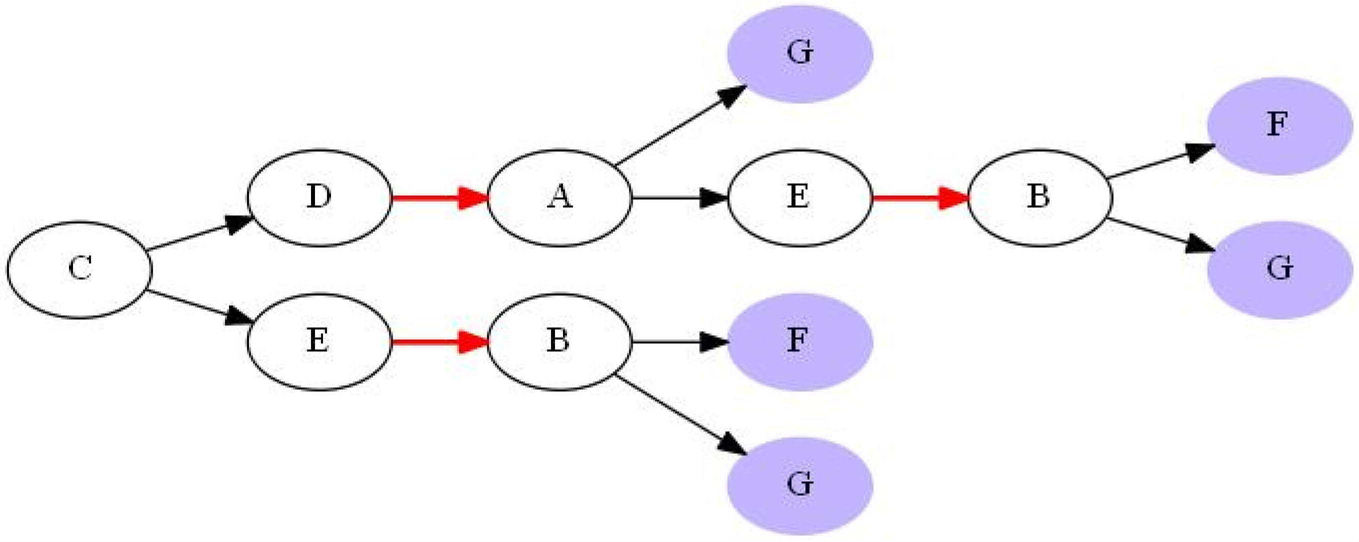
增广路算法

- 现在我们可以给出一个完整的增广路算法来求解最大匹配：
  - 对于一张图，从零匹配开始枚举每个未匹配点
  - 对于一个未匹配点，如果能够找到增广路，则我们将增广路取反，匹配数加1，该点变为匹配点
  - 否则根据增广路引理，我们直接删去这个点，之后仍然能够得到原图的最大匹配，继续枚举下一个未匹配点即可
- 注意刚才的证明全部是针对一般图进行的，所以所有结论也对一般图成立，现在的困难之处只在于如何快速寻找增广路

搜索树

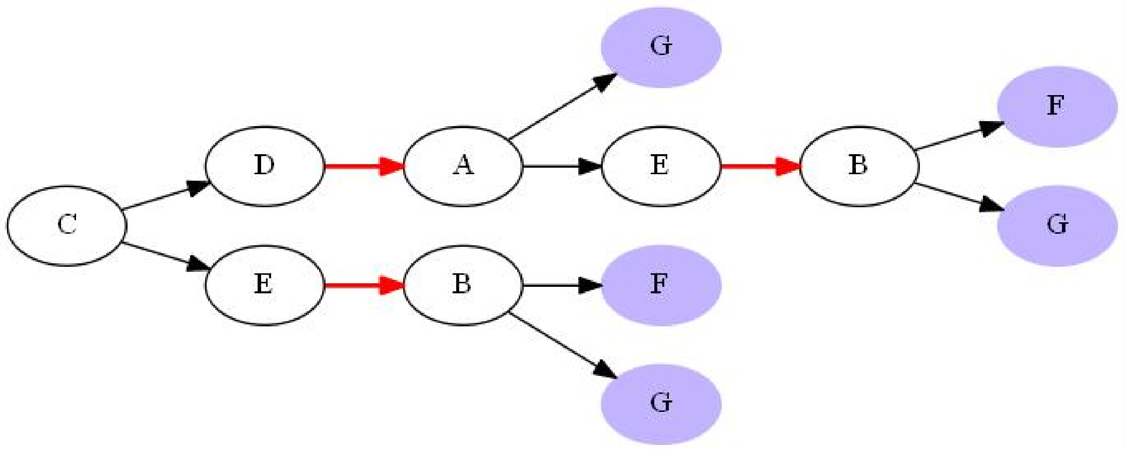
- 不知道怎么办的时候，我们考虑直接爆搜就好
- 例如对于下面的匹配，我们直接以C点为根建立搜索树：

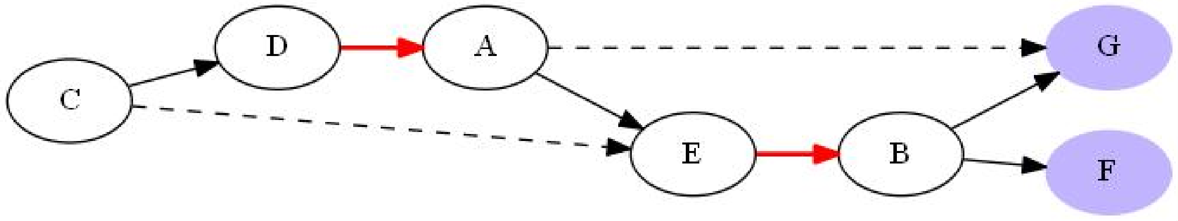




搜索树的性质

- 我们发现两条相交的交错路最终只能选择一条来增广，搜索树可以大幅简化
- 话句话说，对于图中所有的边和点，在搜索树上都只需要出现一次





二分图的搜索树

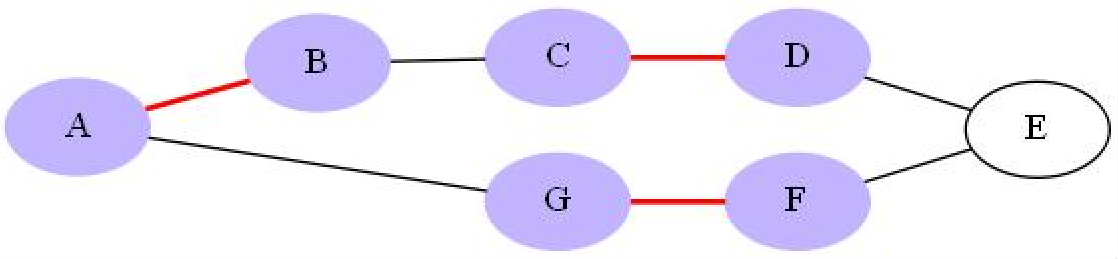
- 对于二分图，以任意一个点为根的搜索树很好建立，划分点集后，我们以其中一个集合的未匹配点为根，那么另一个集合中的未匹配点就是搜索树的叶子，且所有边都在两个集合之间来回，我们构造如下算法：
  - 对于一张二分图，从零匹配开始枚举一个集合中的每个未匹配点
  - 直接从未匹配点开始进行图的DFS遍历，如果走到了另一个集合中的未匹配点，则说明发现增广路，直接进行取反操作，否则通过匹配边走回原集合
  - 如果遍历完成后仍没有发现增广路，说明该点可以忽略，继续枚举下一个未匹配点
- 以上就是二分图匹配的**增广路算法(Augmenting Path Algorithm)**，由于图的遍历是O(E)的，所以该算法时间复杂度为O(VE)

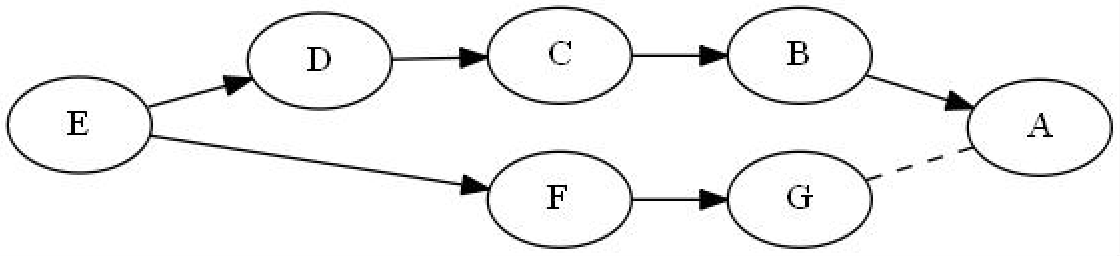
二分图与一般图的不同

- 我们将搜索树上距离根节点距离为偶数的点称为**偶点**，距离为奇数的点称为**奇点**，则二分图与一般图分别有如下性质：
  - 二分图中偶点到奇点的边都是非匹配边
  - 二分图中奇点到偶点的边都是匹配边
  - 二分图中不存在奇点到奇点、偶点到偶点的边
  - 一般图中偶点到奇点的边都是非匹配边
  - 一般图中奇点到偶点的边都是匹配边
  - 一般图中偶点到偶点的边都是匹配边
  - 一般图中不存在奇点到奇点的边，否则不形成交错路

一般图的搜索树

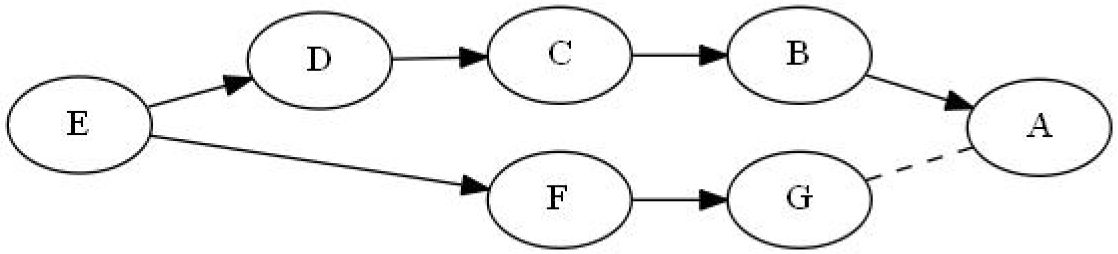
- 偶点到偶点的边，在搜索树上会形成一个环，而且一旦穿越这条偶点到偶点的边，绕过一圈以后环上所有奇点都能够成为偶点，那么它们将来可能可以延伸出更多条交错路



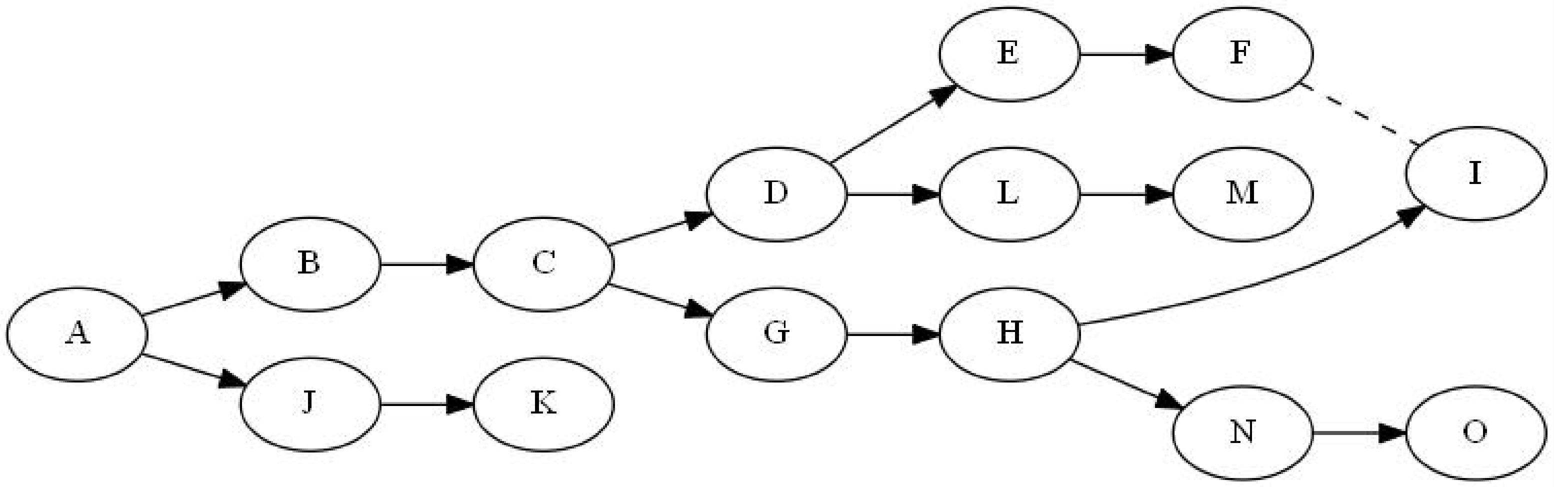


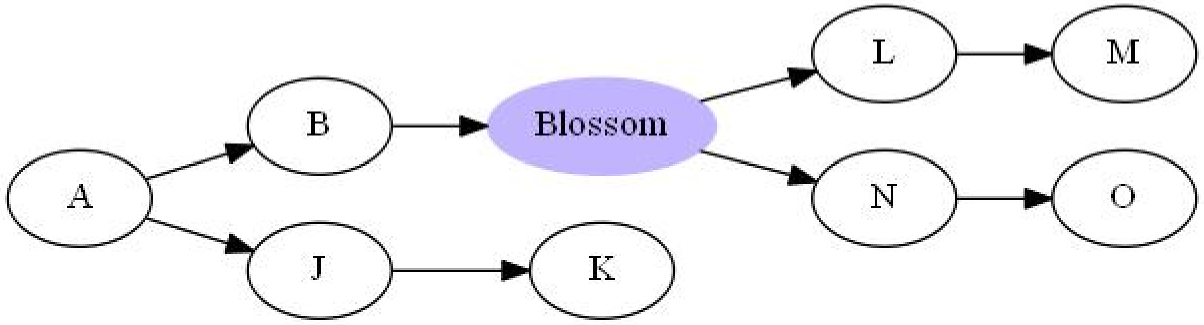
一般图的花

- 原本在搜索树上，奇点只能连向偶点，但现在奇点可以通过一种方式变成偶点，这会给我们建立搜索树带来麻烦
- 我们把搜索树上两条分岔的交错路通过偶点与偶点之间的边连成的奇数条边的环称为**花(Blossom)**
- 花的问题在于，上面的所有点都可以成为偶点，所以我们不妨把整个花直接缩成一个偶点，即**缩花(Blossom Contraction)**，这样搜索树的结构就会更清晰



一般图的缩花图示





缩花

- 在更复杂的图上不难想象各个花之间可能产生重叠，可谓是百花齐放，但注意到我们的目的其实只是化简搜索树并找出增广路，所以缩花的顺序影响并不大，我们可以按任意顺序缩花
- 实际操作时，只要遇到偶点到偶点的边我们就马上进行缩花即可
- 考虑缩花的次数，由于花上至少有三个点，所以每次缩花都会减少两个点，故总的缩花次数不会超过V/2，否则图中就没有节点了，即每次建立搜索树最多进行O(V)次缩花

一般图的增广路算法

- 现在我们可以给出一个完整的算法来求解一般图最大匹配：
  - 对于一张图，从零匹配开始枚举每个未匹配点
  - 对于一个未匹配点，我们进行图的遍历走到一个未访问过的点：如果该点是未匹配点，说明已经找到增广路，直接进行取反，否则的话扩展搜索树，走到那个点匹配对应的另一个点，并继续进行遍历
  - 如果走到一个访问过的奇点，我们什么都不必做并直接返回，否则访问到偶点就形成了花，我们立即进行缩花
  - 如果遍历完成后仍没有发现增广路，说明该点可以忽略，继续枚举下一个未匹配点
- 接下来我们具体讨论怎么进行缩花

一般图的缩花算法

- 我们定义花上两条非匹配边衔接的点为**花托(base)**
- 形成花后，首先我们找出花托（即两个偶点在搜索树上的最近公共祖先），然后将花上的点全部缩成一个新偶点，这个可以用并查集来完成
- 由于缩掉的点不会再出现，所以对于一棵搜索树的所有缩花操作，找花托的总时间复杂度仅为O(V)，之前也证明了最多缩花O(V)次，故效率瓶颈依然在遍历图上
- 以上就是一般图匹配的**带花树算法(Blossom Algorithm)**，由于图的遍历是O(E)的，加上维护并查集的复杂度，该算法时间复杂度为O(VEα(V))

算法优化

- 仔细一想，为什么我们每次只选一个未匹配点作为树根进行扩展呢?
- 以二分图匹配为例，我们不难改进出以下算法:
  - 对于一张二分图，从零匹配开始执行以下操作直到匹配无法增大
  - 以一个集合的所有未匹配点为根，使用BFS建立搜索森林，每次仅扩展一层，直到找出所有目前最短的增广路，总时间复杂度O(E)
  - 对于另一个集合中的未匹配点，如果是搜索森林的叶子节点，则DFS往回寻找具体的增广路，之后将所有增广路取反，总时间复杂度O(E)
- 接下来我们证明，上述过程至多进行O(√V)轮

算法优化的证明

- 在上述算法的前√V轮，由于每一轮我们都找到并取反了最短的增广路，所以每轮所找到的增广路长度一定是严格单调递增的，那么这√V轮执行完以后，剩余的最短增广路长度至少为√V
- 考虑当前匹配与最大匹配的对称差，根据前面的论述我们知道这是由一些两个匹配交替出现的路径或环构成的，且它们的长度均不小于√V，故它们的数量不大于√V
- 由于每取反一条增广路会让匹配数量增加1，所以最大匹配数与当前匹配数最多相差√V

算法优化的证明

- 由于每执行一轮上述算法，最大匹配数至少增加1，所以至多再执行√V轮，我们就得到了最大匹配
- 因为一共执行了2√V轮上述算法，每一轮时间复杂度为O(E)，所以总的时间复杂度为O(E√V)，这就是二分图匹配的**Hopcroft–Karp算法**
- 仔细一想，如果使用网络流求解二分图匹配并采用Dinic算法，其实就是在使用Hopcroft–Karp算法

算法优化在一般图上的扩展

- 扩展刚才的方法，在一般图匹配中也是可以做到$O(E\sqrt{V})$的
- 具体实现比较复杂，由于时间关系我们不展开讨论，感兴趣的同学请参考：
- http://www.cc.gatech.edu/~vazirani/new-proof.pdf
- 这个算法称为一般图匹配的**Micali-Vazirani算法**

一些模型

- 对于一张图G=(V,E)，一个**点覆盖**C是点集的一个子集，且满足E中任意一条边至少与C中一个点相邻
- 在G的所有点覆盖中，元素数量最少的点覆盖就称为G的**最小点覆盖**
- 对于一张图G=(V,E)，一个**独立集**D是点集的一个子集，且满足D中任意两个点之间都没有边相连
- 在G的所有独立集中，元素数量最多的独立集就称为G的**最大独立集**

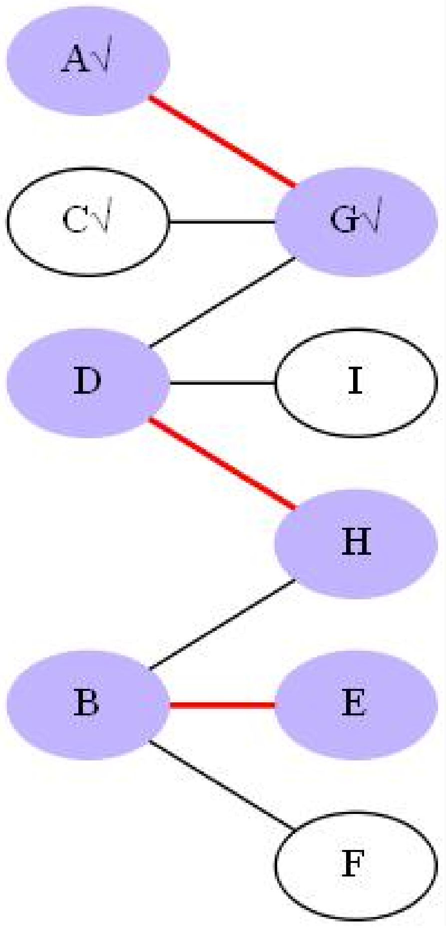
König定理及其证明

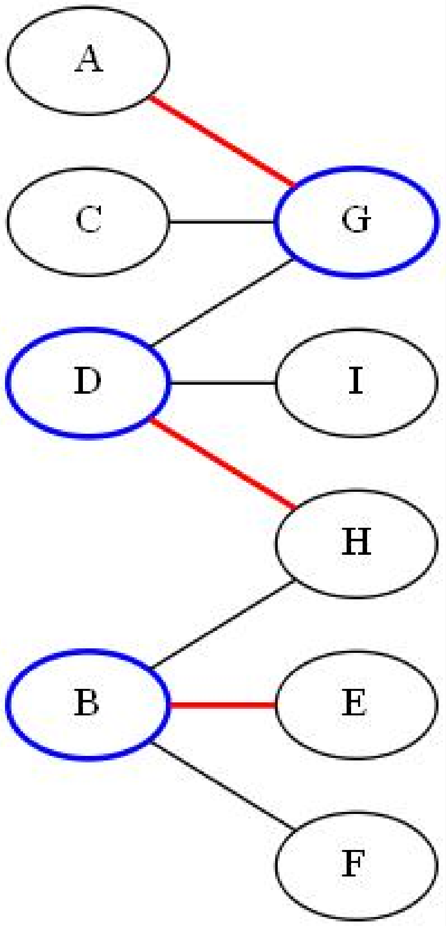
- **König定理**：二分图最小点覆盖数等于最大匹配数
- 证明：由于每条匹配边至少需要选择一个端点才能完成覆盖，所以最小点覆盖数不小于最大匹配数
- 接下来我们构造一个点覆盖数恰好等于最大匹配数的点覆盖：给一个集合的未匹配点打上标记，然后从未匹配的点出发，依次交替走过非匹配边和匹配边，并标记沿途经过的点，显然路径上的最后一条边是匹配边（否则存在增广路），重复上述过程直到初始集合未匹配点连出的边均被遍历
- 将初始集合未标记的点和另一集合被标记的点合起来得到点集C

König定理及其证明

- 由于左边被标记的点和右边未标记的点都是匹配点，所以C的大小恰好为最大匹配数，下证C是最小点覆盖
- 如果存在一条边<u,v>未被覆盖，则必然u被标记而v未被标记
- 如果<u,v>是非匹配边，则从u出发一定会标记v，否则<u,v>是匹配边，当v被标记才可能标记u，矛盾，所以所有边都被C覆盖
- 综上所述，König定理得证

König定理图示





最大独立集定理及其证明

- **最大独立集定理**：最小点覆盖数+最大独立集数=二分图顶点总数
- 证明：我们证明点覆盖的补集是独立集
- 在点覆盖的补集中，任意两个点之间不可能有边相连，否则与点覆盖的性质矛盾，所以点覆盖与独立集一一对应
- 由于顶点总数一定，所以最小点覆盖对应最大独立集，证毕

模型在一般图上的扩展

- 一般图最小点覆盖：我不会
- 一般图最大独立集：我也不会
- 希望有会在多项式时间内解决这两个问题的同学私下教一教我QwQ
- 当然我们不讨论搜索或近似算法

三分图

- 无向图G=(V,E)被称为一张**三分图(tripartite graph)**，如果点集V可以被划分为三个集合$V=V_1\cup V_2\cup V_3$，且满足V_1与V_2的交集、V_1与V_3的交集以及V_2与V_3的交集全部为空
- 对于u,v,w∈V，如果<u,v>,<u,w>,<v,w>∈E，则称<u,v,w>为一个**三元组(3-cluster)**
- 不与任何三元组相关的边称为**冗余边(redundant edge)**，在三分图问题中，一般都会先删去所有冗余边，它们往往不影响答案

三元组与匹配

- 两个三元组的交定义为他们代表的边集的交
- 如果两个三元组有交且不全等，则称这两个三元组相邻，与三元组<u,v,w>相邻的三元组个数称为该三元组的**度(degree)**
- 记σ为G所有三元组的集合，一个**匹配(matching)**M是σ的一个子集，且满足M中不存在两个相交的三元组
- 在G的所有匹配中，元素数量最多的匹配就称为G的**最大匹配**

三元组系列

- σ的连通子集∑称为**<u>三元组系列(3-cluster serials)</u>**，如果∑中的三元组度数均为2，其中∑连通表示如下含义：任意s,t∈∑，存在一列$r_1,r_2,\ldots,r_k\in\sum$，使得$s\cap r_1$、$r_1\cap r_2$、…、$r_k\cap t$均不为空
- 对于G的两个匹配M和M'，∑称为(M,M')的**<u>交错三元组系列(alternating 3-cluster serials)</u>**，如果有s∈M，则与之相邻的t∈M'、如果s∈M'，则与之相邻的t∈M，且任意r∈∑，若有u∈r且u在∑中仅出现一次，则不存在$r'\in(M\cup M')\backslash\sum$使得$u\in r'$
- 如果还有{r|r∈∑∩M}的大小小于{r|r∈∑∩M'}，则称∑为**<u>增广三元组系列(argumenting 3-cluster serials)</u>**

三分图增广路定理

- **增广路定理(Augmenting Path Theorem)**：M为G的最大（三分图）匹配的充要条件是G中不存在M的增广三元组系列
- 这个定理的证明没有本质困难，留给大家作为练习

三分图最大匹配

- 放心，我不会
- 希望有会在多项式时间内解决这个问题的同学私下教一教我QwQ
- 当然我们不讨论拉格朗日松弛

谢谢大家！