浅谈图的匹配算法及其应用

长郡中学 陈胤伯

摘 要

图的匹配算法是信息学竞赛中常用的算法，有着广泛的应用。本文对图的匹配算法进行了总结，总体分为最大匹配和最大权匹配两部分，每部分先介绍二分图上的做法，再进一步提出一般图上的做法，并提出了一些拓展与应用。

1 匹配的有关定义

在图论中，设图 $G = (V, E)$，其中 V 是点集，E 是边集。

一组两两没有公共点的边集 $M(M \in E)$ 称为这张图的一个**匹配**。

定义一个匹配的大小为其中边的数量，边数最多的匹配是**最大匹配**。

当图中边带权时，边权和最大的匹配是**最大权匹配**。

匹配中的边称为**匹配边**，不在匹配中的边称为**非匹配边**。

我们称一个点在匹配中，当且仅当它是某条匹配边的端点。在匹配中的点称为**匹配点**，不在匹配中的点称为**未盖点**。

特别地，所有点都是匹配点的匹配，称为**完备匹配**。

对于匹配中的一条边 (u, v)，我们称 u 和 v 互为配偶。

如未特殊说明，接下来均用 n 来表示图的点数，m 来表示图的边数。

2 增广路

设 P 为图中一条简单路径，如果满足：

1. P 的起点、终点都是未盖点。

2. P 经过的边中匹配边、非匹配边交错出现。

则称 P 是一条**增广路**。

特别地，我们把满足条件 2 的简单路径称为**交替路**。

增广路上非匹配边比匹配边数量多 1，如果把这条路径取反[1]，则当前匹配大小 +1 且依然合法。我们称把增广路取反这个过程为增广。

沿增广路增广，路径上的所有点都将称为匹配点，也就是增广的这个过程会不断扩充匹配点的集合。

定理 1.

一个匹配 M 是图 G 的最大匹配，充要条件是 G 中不存在增广路。

证明 1.

由于取反增广路匹配大小 +1，必要性得证。

考虑 M 和一个更大的匹配 M' 的对称差 $D = M \oplus M' = (M \cup M') \setminus (M' \cap M)$。

根据匹配的定义可以推出 D 中每个点的度数至多为 2，且度为 2 的点两条邻边分别来自 M 和 M'。

这说明 D 由若干环、链组成，且每个环、链中 M, M' 的边交替出现。每个交替环长度一定为偶数，其中 M, M' 的边的数量相同。

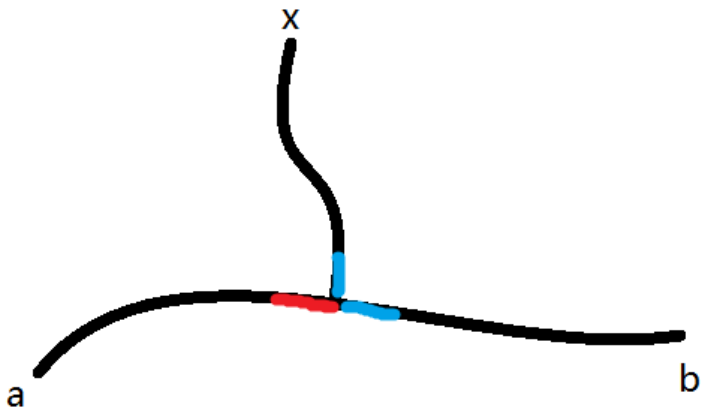
除去所有交替环，此时应该仍有 $|M'| > |M|$，则 D 中一定存在一条 M' 中的边出现得更多的交替链，该链对于 M 来说就是一条增广路。

因此，只要存在更大的匹配，就会有增广路，充分性得证。

由上述定理我们可以得出一个求最大匹配的算法框架：每次枚举图中所有未盖点，找增广路并增广，直到找不到增广路。

其实，如果某一次找不到从未盖点 x 出发的增广路，则增广多次后，依然没有从 x 出发的增广路。

[1]把路径上的匹配边变成非匹配边、非匹配边变成匹配边。为了描述方便，后文中的“把路径取反”、“把路径异或”均指这个意思。



假设某一轮沿增广路 $a-b$ 增广后，出现了以 x 为起点的增广路 P_x，则 P_x 一定是和路径 $a-b$ 有交。考虑 P_x 第一次碰上路径 $a-b$，由于 $a-b$ 是交替路，则触点两端的边类型不同，这意味着在增广前 x 就已经能走到 a,b 中的某个点，即某个未盖点，说明在此之前就已经存在从 x 出发的增广路，矛盾。

用这个结论可以对上述算法框架进一步优化：枚举每个未盖点，各找一次从它出发的增广路。

3 二分图最大匹配

3.1 基本算法

3.1.1 增广路算法

为了描述方便，我们把二分图的点分成左、右两部分，所有边都是横跨左右的。

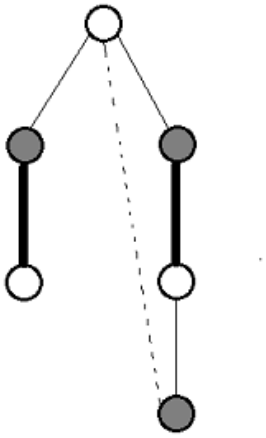
因为增广路长度为奇数，路径的起点终点一定分居左右两侧，不妨从左边的每个未盖点找增广路。

注意到增广路上第奇数次走非匹配边、第偶数次走匹配边，这说明左到右走的都是非匹配边、右到左走的都是匹配边。

因此我们可以给二分图按上述方式定向，问题变成了：有向图中，从给定起点找一条简单路径走到某个未盖点。不难发现有向图中“能通过简单路径走到”等价于“能够走到”，因此只需关心可达性问题。

给二分图定向后，枚举左边一个未盖点 u，从 u 开始 DFS，每个点记录一个访问标记，以保证至多访问一次，当找到某个未盖点时结束遍历。

因此找一次增广路的复杂度为 $O(m)$。

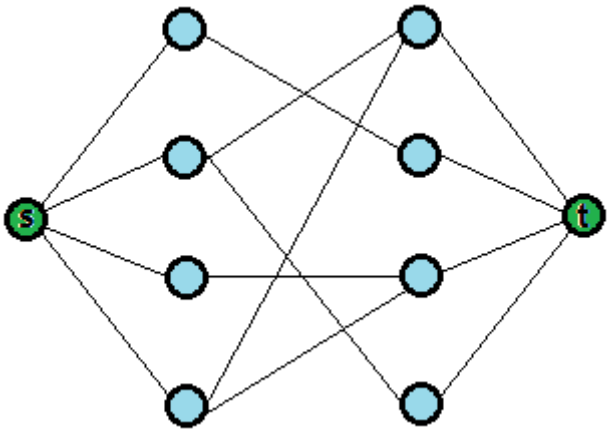


未找到增广路时，我们拓展出的 DFS 树也叫 **交错树**。

因为要枚举 $O(n)$ 个未盖点，上述算法的总复杂度是 $O(nm)$ 的。

3.1.2 Dinic 求最大匹配

二分图最大匹配可以转化为网络流的模型。



新建源汇，源向左边每个点连容量为 1 的弧，右边每个点向汇连容量为 1 的弧，原来的每条边从左往右连容量为 1 的弧。

最大流即为最大匹配，中间的满流边对应的是匹配边。

我们使用 Dinic 算法[1]，可以在 $O(m\sqrt{n})$ 的时间复杂度内求得最大匹配。

这个复杂度是如何得出的呢？

Dinic 算法的每一轮分为两步。第一步是用 $O(m)$ 的时间 BFS 建立层次图，第二步是用 $O(nm)$ 的时间 DFS 进行多路增广。

对于二分图模型，由于每条边容量都是 1，第二步 DFS 的复杂度可以做到 $O(m)$。

对于前 $\sqrt{n}$ 轮，总复杂度为 $O(m\sqrt{n})$。

$\sqrt{n}$ 轮后，因为 Dinic 算法每一轮重建层次图后 S 到 T 的最短增广路距离是严格递增的[2]，那么每条增广路的长度都不短于 $\sqrt{n}$。

换个视角，考虑最大匹配和当前匹配的对称差，即若干个互不相交的交替环、交替链。每一条交替链对应一条增广路，长度至少为 $\sqrt{n}$，因此这样的链的数量不超过 $\sqrt{n}$ 条。

这意味着当前匹配还剩 $O(\sqrt{n})$ 条增广路，也就是接下来 Dinic 的只会最多进行 $O(\sqrt{n})$ 轮，总复杂度为 $O(m\sqrt{n})$。

[1]一种网络流算法，考虑到大多数读者都会该算法，限于篇幅不在此展开详细介绍，不了解的读者可以参考其他资料自行学习。

[2]用反证法可以证明沿最短路增广一次不会产生更短的路，而 Dinic 一轮完毕意味着当前长度的增广路已全部增广完，接下来就只会有更长的了。

3.2 一些拓展

任意图中，设 M, M' 为两个不同的最大匹配，令 D 为 M, M' 的对称差。

则 D 由若干交替环、交替链组成，且由于 M, M' 都是最大匹配，环、链长度都为偶数（长为奇数的交替链代表有增广路）。

这说明，任意两个最大匹配，一个都可以通过异或上若干不相交的偶链、偶环得到另一个。

特别地，两个完备匹配的对称差只有若干不相交的交替环。因为某个匹配异或长度为偶数的交替链后，链的某端点将成为未盖点，说明不是完备匹配。

3.2.1 二分图最大匹配关键点

关键点是指的一定在最大匹配中的点。

由于二分图左右两侧是对称的，我们只考虑找左侧的关键点。

先求任意一个最大匹配 M，要找的关键点此时一定都是匹配点。考虑 M 中的一个匹配点 p，设 M' 为某个不包含 p 的最大匹配，对称差 $D = M \oplus M'$，则 D 中一定存在一条以 p 为端点的偶交替链，这条链另一端不在 M 中。

那么一个匹配点 s 能变成非匹配点，当且仅当从这个点出发能找一条以匹配边出发的交替链，使得终点是某个未盖点 t。由于链长为偶数，t 和 s 属于同一侧（左侧）。

我们倒过来考虑，先给二分图定向：匹配边从右到左、非匹配边从左到右，从左侧每个未盖点出发 DFS，给到达那些点打上标记。最终左侧每个没有标记的匹配点即为关键点。因为只关心可达性，显然每个点只需访问至多一次，复杂度 $O(n + m)$。

3.2.2 二分图最大匹配关键边

我们需要找到哪些边一定在最大匹配中。

同样地，先求任意一个最大匹配 M。

关键边一定是匹配 M 中的边，对于一条边 $e(e \in M)$，假设存在另一个不包含 e 的最大匹配 M'。

M, M' 的对称差中，e 一定存在，则要么属于一个偶交替链、要么属于一个偶交替环。

对于偶交替链的情况，链的某一端一定是未盖点，说明 e 的某个端点能通过交替路走到未盖点，即某个端点是非关键点。因此只需判定 e 两端是否存在非关键点。

对于偶交替环的情况，我们给二分图定向：匹配边从左到右、非匹配边从右到左，再检测 e 是否在某个环中（因为不存在奇环）。用 Tarjan 算法求强连通分量即可。

考虑复杂度，需要先 $O(n+m)$ 求关键点，再 $O(n+m)$ 用 Tarjan 算法求强连通分量。

3.2.3 二分图最大独立集

最大独立集是说，选最多的点，满足两两之间没有边相连。

我们求出二分图的最大匹配 M，根据匹配的定义，M 中的边两两没有公共点，而每条边两端点至少有一个不在独立集中，因此最大独立集的大小至多为 $n-|M|$。

接下来介绍一种构造出大小为 $n-|M|$ 的最大独立集的方法。

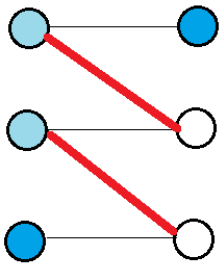
假设已经求出了最大匹配 M，先把所有未盖点加入最大独立集。

考虑一条匹配边 $e=(u,v)$，为了让独立集大小达到上界，e 的两端点 u,v 有且仅有一个要加入到最终的独立集中，不妨把这个“取左还是取右”的决策看做一个变量 x_e，通过枚举 u,v 的出边我们可以得到 x_e 对其他匹配边 e' 的 $x_{e'}$ 的决定关系。

把 $x_{1\sim|M|}$ 之间的决定关系图建出来后，再枚举每个未盖点 t，由于 t 已经被钦定到最大独立集中了，那么与 t 有边相连的所有匹配点所在的匹配边 e 的变量 x_e 的值将被确定。

这样建出来的推导图是否会产生矛盾呢？

如果产生矛盾，那么一定是如图所示的情况：



红色的表示匹配边，蓝色的表示被钦定到最大独立集中的未盖点，他们的值发生矛盾，是通过中间的匹配边们的决定关系来传递的，浅蓝色表示选择下面那个蓝点推导出来匹配边的 x 的取值，于上面发生矛盾。

这意味着找到了一条增广路，与 M 是最大匹配这一条件不符。

因此，我们建立的 $x_{1\sim|M|}$ 的推导图不存在矛盾。这样就可以赋值了，从每个被未盖点决定了 x_e 的匹配边 e 开始 DFS，把能确定的 x 都确定了，最后剩下的还未确定的 x_e 自由取值。

我们的推导图中点数是 $O(|M|) = O(n)$ 的；推导边是通过枚举原图中的边建出来的，所以是 $O(m)$ 的。

因此上述求二分图最大独立集的时间复杂度为 $O(n + m)$。

其实最大独立集也可以直接用网络流建模后最小割求，左边的点在 S 集表示选、右边的点在 T 集表示选，那么中间有边相连的自然就需要割掉了。

这也说明：二分图中，最大独立集 = n - 最大匹配。

3.2.4 二分图最小点覆盖

最小点覆盖是说，选最少的点，满足每条边都至少有一个端点被选。

根据定义不难发现：任意点覆盖的补集都是独立集、任意独立集的补集都是点覆盖。

那么在所有点中去掉最大独立集，得到的就是最小点覆盖。

这也说明：二分图中，最小点覆盖 = n - 最大独立集。

3.2.5 二分图最小字典序完备匹配

最小字典序是指：令 $match_i$ 表示与左边第 i 个点匹配的右边的点的编号，最小化 $match_{1\sim n_L}$ 的字典序。

之前提到过，任意两个完备匹配的对称差一定是若干个交替环。那么我们先求任意一个最大匹配，即一个完备匹配。

根据字典序贪心的思想，按编号从小到大枚举左边每个点 u 来决定 $match_u$。

我们从小到大枚举右边每个和 u 有边相连的点 v，每次需要检验边 (u, v) 能否成为匹配边。若其为匹配边则显然能；否则，给二分图定向后等价于求边

(u,v) 是否在一个环里，即询问 u,v 是否属于同一强连通分量，如果是则说明能、否说明不能。

对于上述每个点 u，找到最小的能匹配的 v 即可确定匹配关系。决定了 $match_u = v$ 后，为了防止再次被修改，我们不妨把 u、v 从二分图中删去。

匹配关系只会变动 $O(n)$ 次，因此只需求 $O(n)$ 次强连通分量。对于某个 u，找到合法的 v 后，找交替环用 DFS 是 $O(m)$ 的。

因此时间复杂度为 $O(nm)$。

3.3 应用

我们知道了在二分图中，最大匹配 = 最小点覆盖 = n - 最大独立集。

那么二分图模型的上述问题都能解决。

常见的二分图模型有棋盘的行列、人为的黑白染色，或者能证明出无奇环的图等。

这一类题目十分常见，相信读者已经见过很多，在此只进行总结而不加赘述。

下面介绍一些思路较为巧妙的二分图匹配的题目。

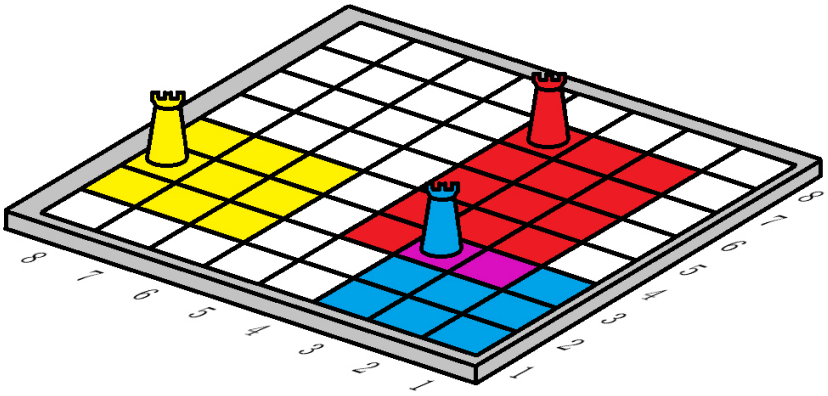
3.3.1 Problem 1. Chessboard

题意 [1]

有一个 $n \times n$ 的棋盘和 n 个车，你要将 n 个车放进棋盘中使得每行每列都至多有一个车。$n \leq 10^5$。

出于某些原因，每个车的坐标范围限制在一个矩形区域内。

[1] 来源：POJ 挑战赛 Round 2 - by sumix173



请你求出哪些车的摆放位置是唯一确定的。

题解

首先题中的 x, y 坐标是独立的，我们只看一维。

问题变成一个排列 P，数字 x 出现的位置限定在 $[L_x, R_x]$，求哪些数字出现位置固定。

我们建立二分图，左边对应数字，右边对应排列位置，左边每个点向右边一个区间的点连边。我们等于询问哪些边一定在最大匹配中。

这是之前提到过的“关键边”问题。

注意为了保证有解，这个二分图一定有完备匹配。也就是说“求关键点”这一步可以省略。那么我们只需要像之前说的，给二分图定向（匹配边向左、非匹配边向右），然后判定每条边是否在环里。

我们用 Tarjan 求强连通分量，如果一个表示数字的点所在的强连通分量大小大于 1，说明这个点的出边在环内，说明这条出边不是“关键边”，即这个点所代表的数字的位置不确定。

由于是区间连边，又只关心连通性，可以用对右边的点建立线段树结构来优化连边，这样边数是 $O(n \log n)$ 的。

至于本题的求完备匹配，这是一个经典的贪心问题，即所有数字按 L 排序然后从左到右枚举每个位置，每次填入 R 最小的数字，即得到一组方案。

时间复杂度 $O(n \log n)$。

3.3.2 Problem 2. Minimum Diameter

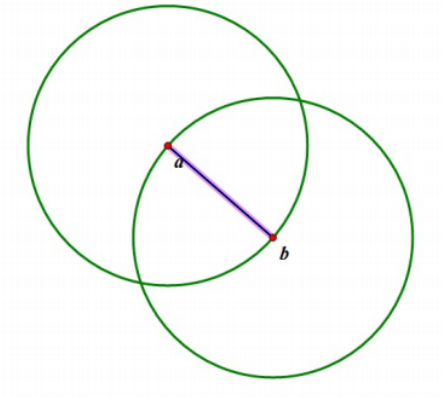
题意 [1]

给平面上 $n(n \le 1000)$ 个点，删除其中 $k(k < n, k \le 30)$ 个点，使得剩下的点中相距最远的点对最近。

题解

枚举剩下来的点里相距最远的两个点 a, b。

以 $|ab|$ 为半径，a, b 为圆心画圆。



那么剩下的点 x 需要满足 $|xb|, |xa| \le |ab|$，也就是一定在两圆的交集内部。

接下来，我们在两圆交集内的点中找到所有距离 $> |ab|$ 的点对，连上边，表示不能同时选。最大独立集即为内部最多能保留的点，检查一下删除的点总数是否 $\le k$ 即可。

注意到每条边一定是跨越了两圆交集的两瓣的中线 ab，因为同一瓣内不存在距离 $> |ab|$ 的点对。所以这实际上是一个二分图最大独立集问题。

由于最大独立集 = n - 最大匹配，当发现最大匹配 $> k$ 时，说明最大独立集 $< n - k$，说明需要拿走超过 k 个点，这是不可行的，因此可以直接终止这轮匹配。

找一次增广路是 $O(m) = O(n^2)$ 的，至多找 k 次增广路就会终止算法。再考虑之前枚举 a, b 的复杂度，总时间复杂度 $O(kn^4)$。

尽管这个复杂度不能通过此题，但也是一个多项式级别的不错的算法，思想十分巧妙。

[1] 来源：CodeForces VK Cup 2012 Round 3 D

完整做法由于和主题关系不大且限于篇幅在此不深入讨论，有兴趣的读者可以参考陈老师 2012 年自选题解题报告（见参考文献）。

3.3.3 Problem 3. The Square Div One

题意 [1]

一个 $n \times m (n, m \leq 50)$ 的矩阵，你需要给每个格子填上 $1 \sim \max(n, m)$ 的整数，使得每行每列不存在重复数字。

求字典序最小的方案。（字典序比较按先行后列的顺序）

题解

根据字典序的定义，一个直接的想法是：从上到下一行行确定数字。

考虑这一行每个位置，根据所在列已出现的数字，将每个位置与可以填的数字之间连边，求字典序最小的匹配。

遗憾的是，这样做可能导致之后几行无解。因为一个数字必须出现 $\max(n, m)$ 次，如果前几行填得太少，后几行可能行数不足以让它填完所需次数。

如何修复这个问题呢？

考虑正在填某一行，令：$P = \max(n, m)$（数字种数）、r_x 表示数字 x 还需要出现多少次、h 表示还剩多少行未确定。

则有解当且仅当：

1. $\sum_{i=1}^{P} r_x = h \times m$
2. 对于任意 x 满足 $r_x \leq h$

必要性是显然的。充分性的证明：当 $h = 1$ 时显然成立；当 $h > 1$ 时，我们强制填 $r_x = h$ 的那些 x，由于 $\sum_{i=1}^{P} r_x = h \times m$，那么 $r_x = h$ 的 x 一定不超过 m 个，也就是说这一行位置是充足的。既然如此，填完到下一行时，发现条件依然成立，递归证明即可。

所以每次只要保证把 $r_x = h$ 的 x 填一次即可。

考虑二分图模型。右边 P 个点表示数字，左边 m 个点表示位置。

一个数字能填在某个位置就连一条边。

[1] 来源：Topcoder - SRM 459

若 $P > m$ 则在左边补一些点表示选择不填，右边满足 $r_x < h$ 的 x 向新建的点连边。

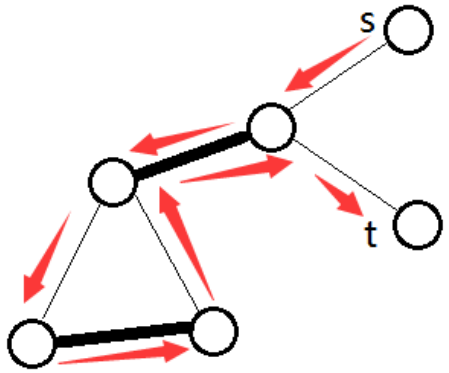
用之前说的做法求这个二分图的最小字典序完备匹配。

4 一般图最大匹配

4.1 带花树算法

一般图和二分图的唯一区别在于可能存在奇环，这导致我们不能转换成有向图可达问题。

能不能还是只关心走交替边的可达性呢？



很遗憾并不行，如上图所示，虽然 s 到 t 可达，但由于不是简单路径，取反后将使得某些点在两条匹配边上，这是非法的。

我们发现问题出在奇环。如果上图的环是偶环，即使只关心可达性我们也不会绕回同一个点，因为绕偶数步回到同一个点不改变下一步走的边的类型。

下面考虑一般图的增广路算法。

不妨先当做二分图来做，每次枚举一个未匹配点 s，从 s 开始按交替的规则 DFS 找增广路。

我们对走出的这棵交错树中的点奇偶标号，其中根为偶点。不难发现偶点是被奇点直接用匹配边“拉进来”的，每次我们枚举的是偶点的出边。

设当前所在点为 u，枚举的下一个点为 v。

如果 v 未访问过：若 v 是未盖点则找到增广路；否则从 v 的配偶开始 DFS。

如果 v 已被访问：则当前枚举的出边是交错树的非树边，找到了一个环。

如果找到的是偶环，直接忽略。因为只有偶环的图就是二分图，算法正确性之前已经证明过了。

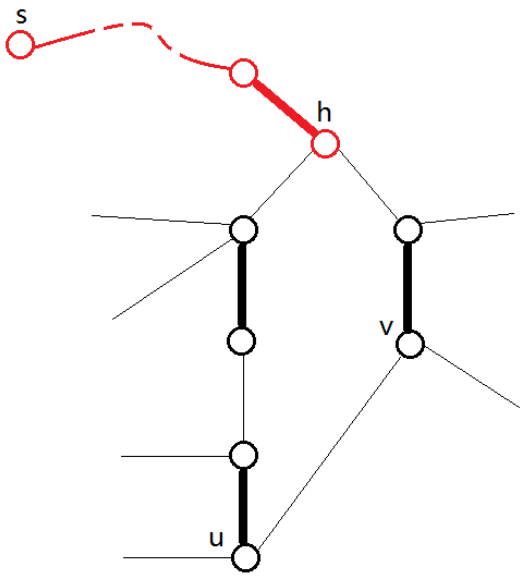
如果找到的是奇环，我们把环上所有边缩掉，即把整个环缩成一朵花（一个点）。然后在新图中重新找增广路。

下面来分析这样做的理由。

设原图为 G，缩点后得到的新图为 G'，我们只需要证明以下两点：

1. 若 G 存在增广路，则在 G' 中也存在。
2. 若 G' 存在增广路，则在 G 中也存在。

在证明之前，我们先画出交错树和一条非树边，即找到的那个奇环。



设非树边为 (u,v)，定义花根 $h = LCA(u,v)$。

奇环一定是交替的，有且仅有 h 的两条邻边类型相同，都是非匹配边。

那么进入 h 的树边一定是条匹配边，环上除了 h 外其他点往环外的边都是非匹配边。

观察可知，沿匹配边进入 h 可以到环上其他任意点沿非匹配边出去（顺时针、逆时针总有一种走法可行），反之亦然。

我们称缩成的点为新点，被缩的环为旧环。

若 G 存在增广路，则在 G' 中也存在。

设之前图中 $s-h$ 这一段红色路径为 S。

令 G_S, G'_S 分别表示 G, G' 中把路径 S 取反后得到的新图，显然 G_S, G'_S 相对于自己的原图来说匹配合法且大小不变。

既然 G 存在增广路，说明 G 中的匹配不是最大的，也就说明了 G_S 中的匹配不是最大的。根据增广路定理，G_S 中也应该存在增广路。

设 G_S 中的增广路为 P，若 P 没有经过旧环，则 G'_S 中也存在 P。

否则，让 P 刚走上旧环就径直沿着环走到 h，在 G_S 中 h 是未盖点，那么这样走也是一条合法的增广路，把这条增广路对应到 G'_S 中，相当于直接走到了新点，也是合法的增广路。

又因为 G'_S 和 G' 的匹配大小相同，既然 G'_S 存在增广路，那么 G' 也应该存在增广路。

若 G' 存在增广路，则在 G 中也存在。

G' 中的增广路，如果没走新点，那么 G 中显然存在一条一模一样的。

如果走了新点，无非是一进一出，进出分别是一条匹配边和非匹配边。

考虑把这两条边对应到 G 中去，匹配边一定是和 h 相连的那一条。

我们之前已经分析过了 h 可以到旧环上任意一点，反之亦然。那么 G' 中对新点的一进一出，就可以根据进出边在旧环上的位置，还原出在 G 中旧环上的走法，得到 G 中的增广路了。

进一步地可以发现：如果当前是从 s 出发找增广路碰到了奇环，那么如果存在从 s 出发的增广路，缩环之后一定还存在从 s 出发的增广路。因此缩点之后，找增广路的起点不变。

在找到奇环的时候，为了实现方便，我们不“显式”地缩点，而是用并查集来记录每个点现在在以哪个点为根的花中。在记录匹配关系的同时，用一个 *next* 数组记录最终增广的时候每个点的后继，每朵花上的 *next* 实际组成了一个双向链表来描述花内部的走法。主程序用一段 BFS 来拓展交错树，队列中存所有偶点，在碰到奇环时，用并查集完成缩点并将环上所有奇点标记为偶点加入 BFS 的队列。这样可以做到一次找增广路的时间为 $O(n^2)$。

一共要找 $O(n)$ 次增广路，因此总复杂度为 $O(n^3)$。

4.2 应用

4.2.1 Problem 1. 无向图游戏

题意

Alice 和 Bob 在一张 $n(n \le 200)$ 个点的简单无向图上玩游戏，他们轮流操作，其中 Alice 先手。

最初，有一个小人在某个点 s 上。每次，设小人在点 u，玩家可以让小人走到和 u 相邻的某个点 v 上去。小人走过的点会被炸掉，从图中永久删去。不能操作者输。

已知 Alice 和 Bob 足够聪明，求有多少个 s 是先手必胜。

题解

结论是：若存在一种最大匹配，满足 s 不是匹配点，则先手必败；否则先手必胜。

首先，若存在一种最大匹配满足 s 不是匹配点，那么先手每走一步（设走到了 v），后手只需走到 v 的匹配点 u 即可。这样后手总是有路可走，除非走到某个未盖点，但注意如果这种情况发生，我们实际上找到了一条从未盖点到未盖点的交替路，即增广路，这与之前所说的最大匹配的条件矛盾。

若不存在，即 s 一定在最大匹配中，那么任意找一个最大匹配，然后先手使用上一段描述中后手的策略。这样若先手无路可走，说明到了某个未盖点，走出了一条只有一端是未盖点的交替路，但这条路一旦取反，s 将不再在匹配中而匹配大小不变，说明 s 并非一定在最大匹配中，与之前的条件矛盾。

现在问题变成了：在一般图里，判定哪些点一定在最大匹配中。

我们任意找一个最大匹配，枚举每个匹配点，把它从图中删去，看看是否存在增广路，如果是则说明该点不一定在最大匹配中。

注意删去一个点 u 后，如果存在增广路，u 的配偶 v 一定是增广路的一端。否则，考虑删点时匹配数 -1，增广后匹配数 $+1$，此时 u, v 还能匹配起来匹配数再 $+1$，得到了一个比之前求出的最大匹配更大的匹配，矛盾。

因此对于枚举的 u，我们只需从 v 开始找增广路即可。

首先求一次最大匹配是 $O(n^3)$ 的。之后枚举一个匹配点 $O(n)$，删去后用带花树可以在 $O(n^2)$ 的时间里判定是否存在从某个点出发的增广路。

时间复杂度 $O(n^3)$。

4.2.2 Problem 2. 拓扑图拿点

题意

有一张 $n(n \le 200)$ 个点的拓扑图，每轮你可以同时拿走至多两个入度为 0 的点（入度在每一轮结束后才重新结算，拿走的两个点不能有拓扑关系），求至少多少轮能拿走所有点。

题解

考虑拓扑图上任意两个有拓扑关系的点 u,v，即 u,v 中一个可以到达另一个，它们显然不能在同一轮被拿走。那么我们建立新图 G'，在所有没有拓扑关系的点对之间无向边，表示这条边两端点可以在一轮被同时拿走。

由于点数一定，要最小化拿完全部的轮数，等价于最大化一轮拿两个的轮数 R。

考虑 G' 中的最大匹配，显然是 R 的上界。如果这个上界可以达到，也就求出了最大的 R。

下面我们证明，这个上界是一定可以达到的。

首先，考虑原拓扑图中所有入度为 0 的点，再考虑 G' 中的匹配关系。

如果某个点没有配偶，直接拿掉。

如果某个点的配偶入度也为 0，一起拿掉。

现在情况只剩下：所有入度为 0 的点配偶都被“压着”了。考虑这些配偶们之间的拓扑序，这些配偶们中处于拓扑序顶点的（即不被其他配偶压着）配偶一定有至少两个。

否则，设那个唯一处于顶点的为 u，由于 u 在原拓扑图入度不为 0，一定存在一个入度为 0 的点 v 能到达 u，既然 v 压着 u，那么 v 一定也压着了 v 自己的配偶，而根据定义 v 是不能到达 v 的配偶的，因此矛盾。

既然至少有两个，设她们为 x,y，和她们匹配的入度为 0 的点为 X,Y，原来匹配关系是 $x-X, y-Y$，我们改成 $X-Y, x-y$，其中 X,Y 都入度为 0 所以可以一并拿去，x,y 之间不存在拓扑关系所以可以匹配起来。

这样每次可以调整最大匹配使得大小不变且合法，并且总能调整出一条匹配了两个入度为 0 的点的匹配边，那么就可以做到刚好 R 轮拿走两个了。

上述证明事实上也提供了一种找合法解的方法。

时间复杂度 $O(n^3)$。

5 二分图最大权匹配

最大权匹配，是指边权和最大的匹配。

如果我们在二分图左右两边点数较少的一边补点，使得左右两边点数一样，再将不存在的边看做权值为 0，那么最大权匹配问题可以转化为最大权完备匹配问题。

接下来我们讨论最大权完备匹配问题，即找一个边权和最大的匹配，满足每个点都在匹配中。

5.1 KM 算法

KM 算法又叫匈牙利算法，可以在 $O(n^3)$ 的时间求出最大权完备匹配。

我们先介绍两个概念。

可行顶标是指给每个节点 i 分配一个权值 $l(i)$，对于所有边 (u,v) 满足

$$w(u,v) \le l(u) + l(v)$$

相等子图是指在一组可行顶标下原图的生成子图，包含所有点，只保留满足 $w(u,v) = l(u) + l(v)$ 的边。

定理 2.

对于某一组可行顶标，如果其相等子图存在完备匹配，那么该匹配一定是原二分图的最大权完备匹配。

证明 2.

考虑原二分图任意一组完备匹配 M，其边权和

$$val(M) = \sum_{(u,v)\in M} w(u,v) \le \sum_{(u,v)\in M} l(u) + l(v) \le \sum_{i=1}^{n} l(i)$$

任意一组可行顶标下的相等子图的完备匹配 M' 的边权和

$$val(M') = \sum_{(u,v)\in M'} l(u) + l(v) = \sum_{i=1}^{n} l(i)$$

即任意一组完备匹配的边权和都不会比 M' 的大，那么 M' 就是最大权完备匹配。

有了上述定理，我们的目标就是通过不断调整顶标，使得相等子图中存在完备匹配。

为了方便描述，我们作如下定义。

由于左右两边点数相等，令 n 表示一边的点数。令 lx_i 表示左边第 i 个点的顶标，rx_i 表示右边第 i 个点的顶标，$w(u,v)$ 表示左边第 u 个点和右边第 v 个点之间边的权值。

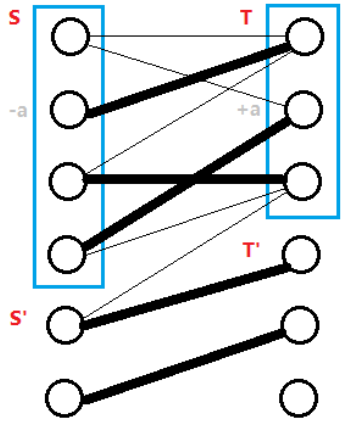
先初始化一组可行顶标，比如

$$lx_i = \max\{\, w(i,j) \mid 1 \le j \le n \,\}, rx_i = 0$$

然后每次选一个未盖点，像做最大匹配那样试着找一条增广路。

如果找到了增广路就直接增广。

否则，我们将得到一棵之前所说的交错树。令 S、T 表示二分图左边、右边在交错树中的点，S'、T' 表示二分图左边、右边不在交错树中的点。



在相等子图中：

- S 到 T' 一定没有边。因为若存在非匹配边，交错树应该会继续生长；S 中所有点要么是起点 s、要么和 T 中匹配，所以匹配边显然没有。
- S' 到 T 一定都是非匹配边，否则 T 中对应点会从匹配边遍历到 S'。

如果我们给 S 中所有点顶标 $-a$，T 中所有点顶标 $+a$，那么原来的匹配边、$S-T$ 的交错树中的边一定都还在相等子图，$S-T'$ 中可能有新的边加入相等子图。

不难推出：

$$a = \min\{\ lx_u + ly_v - w(u,v) \mid u \in S, v \in T'\ \}$$

如果 a 更大，顶标将非法；如果 a 更小，相等子图将不会变化。

当一条新的边 (u,v) 加入相等子图后，有两种情况：

1. v 是未盖点，则找到增广路。

2. v 和 S' 中某个点已经匹配，则将 v 和它的配偶都加入交错树。

每次修改顶标后，要么找到增广路，要么交错树中节点 +2。这说明至多修改 $O(n)$ 次顶标就能找到增广路。

每次修改顶标时，交错树中的边并不会离开相等子图，因此我们实质上是在扩展这棵交错树，考虑直接维护这棵树。

我们对 T' 中每个点 v，维护

$$slack(v) = \min\{\ lx_u + ly_v - w(u,v) \mid u \in S\ \}$$

以便每次 $O(n)$ 算出顶标修改值

$$a = \min\{\ slack(v) \mid v \in T'\ \}$$

在交错树新加入一个 S 节点的时候，$O(n)$ 更新一下 $slack(v)$。

在修改顶标的时候，$O(n)$ 给每个 $slack(v)$ 减去 a。

只要交错树扩展到某一个未盖点，那么就找到了增广路。

回顾一下，最初枚举一个未盖点是 $O(n)$ 的，每次为了找增广路要扩展 $O(n)$ 次交错树，每次扩展需要 $O(n)$ 维护，总复杂度 $O(n^3)$。

5.2 应用

首先可以解决最大权匹配、最大权完备匹配的问题。

通过把边权取负，也可以解决最小权匹配问题。

KM 算法的顶标模型

之前的二分图模型中，找一组 lx, ly 满足：

$$lx_i + ly_j \geq w(i,j)$$

最小化：

$$\sum_{i=1}^{n} lx_i + \sum_{i=1}^{n} ly_i$$

KM 算法求出的顶标即为答案，这是因为任意合法顶标的顶标和都会大于等于最大权完备匹配的边权和，而 KM 求出的顶标刚好是这个下限。

6 一般图最大权匹配

6.1 带权带花树

类似二分图中 KM 算法的顶标，我们给每个节点 i 定义一个非负顶标 z_i、给每个奇数点集 B 定义一个非负顶标 z_B。

对于一条边 $e=(u,v)$，令 $f_e = z_u + z_v + \sum_{B \mid u,v\in B} z_B$，$w_e$ 表示 e 的边权。

如果对于某个匹配 M，满足：

1. $f_e \geq w_e$ for $e \in E$
2. $f_e = w_e$ for $e \in M$
3. $z_i = 0$ 若 i 是未盖点，$z_B = 0$ 若 B 中匹配边不足 $\frac{|B|-1}{2}$

那么 M 一定是最大权匹配。

因为对于任意匹配 M'，其边权和一定不超过：

$$\text{LIMIT} = \sum_{i=1}^{n} z_i + \sum_{B} z_B \times \frac{|B|-1}{2}$$

我们把边权 w_e 放大成 f_e 求和，上式中前者是节点顶标贡献的上界，后者是点集顶标贡献的上界。

而注意到我们的匹配 M 的边权和刚好等于 LIMIT，因此 M 一定是最大权匹配。

（这里看起来 B 的定义十分古怪，后面会看到 z_B 实际上是对每个缩起来的花维护的一个权值。这里只需理解其正确性。）

现在看来我们的目标在于，同时找到一组顶标和一个匹配 M，满足上述条件。

称满足条件 1, 2 的一组顶标是合法的。

称所有 $f_e = w_e$ 的边 e 组成的图为**相等子图**。

初始化当前匹配 $M = \emptyset$，$z_i = \max w_e/2$。一开始所有 $z_B = 0$，我们只记录和维护那些 $z_B > 0$ 的点集。

注意到初始的状态只违背了条件 3。

我们的算法过程将在满足条件 1、2 的前提下，不断减少违背条件 3 的点的数量。

算法和求最大匹配时一样，分为 n 个阶段，每个阶段我们在相等子图中找增广路，和求最大匹配时一样，不过我们从所有未匹配点一起开始 BFS，把点奇偶染色（所有根均为偶点），用一个队列 Q 存所有偶点，缩花后花内所有点标记为偶点并把新的偶点加入队列，如果找到了增广路则直接增广，否则调整 z。

为了保证条件 3，我们在一个阶段结束后需要保护那些 $z_B > 0$ 的缩起来的花 B，也就是不展开。因此我们在 BFS 交错树的时候，把花看做点，除了偶花还可能会碰到奇花。

如果相等子图中没有找到增广路，像二分图 KM 算法中那样调整 z 的值，即：

- $z_i -= a$，如果 i 是偶点。
- $z_i += a$，如果 i 是奇点。

然而一个花内部所有点类型相同，花内的边两端都 $+a$ 或都 $-a$，势必会造成这条边离开相等子图，甚至顶标不合法，这时就要用到之前定义的 z_B 了。

- $z_B += 2a$，如果 B 是偶花。
- $z_B -= 2a$，如果 B 是奇花。

这里说的花均指最外层的花（因为花可能里面套着花），这样就可以保持花内的边依然留在相等子图了。

下面我们需要考虑 a 的取值，不难发现是下面四种取 min：

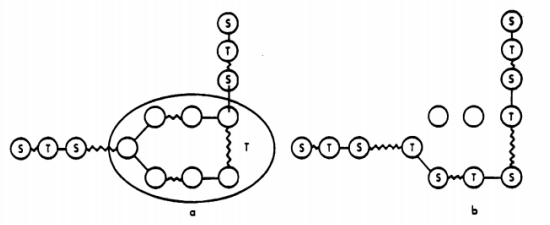
1. z_i，i 是偶点。
2. $f_e - w_e$，$e = (u, v)$、u 是偶点、v 是未访问点。
3. $(f_e - w_e)/2$，$e = (u, v)$、u, v 是不在同一个花里的偶点。
4. $z_B/2$，B 是奇花。

容易发现这样的 a 不会使花中的边、交错树上的边、匹配边从相等子图中消失，也不会使顶标非法。

我们按 a 最终是在哪种情况中取到 min 的来分类。

如果是情况 1，那么这次改变后条件 3 将可以得到满足。

如果是情况 4，那么会有若干奇花的权值变成 0。奇花肯定不是当前这一轮形成的，一定是前几轮留下的，之前保留的理由是为了满足条件 3，现在它权值为 0 了，我们就可以把它展开了。



其中 a, b 分别表示展开前、展开后。展开后这朵奇花的子花就成了当前交错树中的花。把所有新的偶点加入队列 Q。

如果是情况 2 或者 3，都将有至少一条新的边加入相等子图中。如果某两个不在同一棵树中的偶点之间加入了一条边，则找到增广路。

这样我们总会找到增广路。这一阶段结束后，再把交错树上所有权值为 0 的偶花展开，进入下一阶段。

时间复杂度 $O(mn^2)$。

用类似二分图中的维护 *slack* 变量的方法可以做到 $O(n^3)$。

6.2 应用

6.2.1 无向图中国邮递员问题

题意

给定一张 $n(n \le 200)$ 个点的简单无向图，边有边权。找一条最短的封闭路径，经过所有边至少一次。

题解

我们相当于要加最少的边，使得原图存在欧拉回路（即每个点度数都是偶数），其中新加的边的边权为两点间的最短路。

用 Floyd 算法 $O(n^3)$ 求出所有点的两两最短路，问题转化为度数为奇数的点之间的最小权完备匹配，把边权都用一个大数减去后转化为最大权匹配。

7 总结

我们讨论了四种类型的最大匹配问题，从最大匹配到最大权匹配，从二分图到一般图，每一类问题都介绍了一些多项式算法。

我们发现，匹配算法的核心在于找增广路，最大权匹配通过顶标模型一定程度可以转化为最大匹配问题从而得以解决。

事实上，图的匹配的内容远不止这些，还有很多更为高效的算法以及一些对匹配问题的研究历程本文中没有介绍，有兴趣的读者可以自行研究，希望本文能起到抛砖引玉的作用。

8 感谢

感谢中国计算机学会提供学习与交流的平台。

感谢吕凯风、黄志翱、郭晓旭、罗雨屏、余行江、胡家琛、张天扬同学在论文写作上对我的帮助。

感谢向期中教练、徐光明老师对我的栽培。

感谢长郡中学的同学们对我的帮助与支持。

感谢我的亲人对我的关心和照顾。

参考文献

[1] Wikipedia，Berge’s lemma

[2] Wikipedia，Hopcroft－Karp algorithm

[3] Matthew Kusner，《Edmonds’s Blossom Algorithm》

[4] Zvi Galil，《Efficient Algorithms for Finding Maximal Matching in Graphs》

[5] 刘汝佳，《算法竞赛入门经典》，清华大学出版社

[6] 陈立杰，《CodeForces VK Cup 2012 Round 3 D. Minimum Diameter》

[7] 黄志翱，《最大权匹配》

[8] 范浩强，《无向图匹配的带花树算法》