二分匹配基础总结

1. 夯实基础
   1. 图论点、边集和二分图的相关概念
      1. 最小点覆盖

**点覆盖集：即一个点集，使得所有边至少有一个端点在集合里。或者说是“点” 覆盖了所有“边”。**

**最小点覆盖：点最少的点覆盖，最小点覆盖的点数称为点覆盖数。**

* + 1. 最小边覆盖

**边覆盖集：即一个边集，使得所有点都与集合里的边邻接。即“边” 覆盖了所有“点”。 最小边覆盖：边最少的边覆盖，最小边覆盖的边数称为边覆盖数。**

* + 1. 最大独立集

**独立集：即一个点集，集合中任两个结点不相邻，则称V为独立集。或者说是导出的子图是零图（没有边）的点集。**

**最大独立集：点最多的独立集。最大独立集的点数称为独立数。**

* + 1. 团

**团：即一个点集，集合中任两个结点相邻。或者说是导出的子图是完全图的点集。**

**最大团：点最多的团。 最大团的点数称为团数。**

* + 1. 最小路径覆盖

**最小路径覆盖：用 “路径” 覆盖“点”，即用尽量少的不相交简单路径覆盖有向无环图G的所有顶点，即每个顶点严格属于一条路径。路径的长度可能为0(单个点)。**

* + 1. 匹配

**匹配(matching)：是一个边集，满足边集中的边两两不邻接。匹配又称边独立集(edge independent set)。相对应的有最大匹配以及匹配数。**

**在匹配中的点称为 匹配点 ；反之，称为 未匹配点。**

**最大匹配：是具有最多边的匹配。**

**匹配数： 是最大匹配的大小。**

**完美匹配：是匹配了所有点的匹配。**

**完备匹配：是匹配了二分图较小集合（二分图X，Y中小的那个）的所有点的匹配。**

**二分图带权最优匹配：对于二分图的每条边都有一个权（非负），要求一种完备匹配方案，使得所有匹配边的权和最大，记做最优完备匹配。（特殊的，当所有边的权为1时，就是最大完备匹配问题），即二分图的带权匹配就是求出一个匹配集合，使得集合中边的权值之和最大或最小。**

**交错轨：是图的一条简单路径，满足任意相邻的两条边，一条在匹配内，一条不在匹配内。**

**增广轨：是一个始点与终点都为未匹配点的交错轨。**

**增广轨定理：一个匹配是最大匹配当且仅当没有增广轨。（所有匹配算法都是基于增广轨定理）**

* 1. 二分图
     1. 二分图的判定

**二分图是这样一个图： 有两顶点集且图中每条边的的两个顶点分别位于两个顶点集中，每个顶点集中没有边直接相连接！**

**无向图G为二分图的充分必要条件是，G至少有两个顶点,且其所有回路的长度均为偶数。（即不存在奇环），常用的方法 dfs 染色法**

**易知：任何无回路的的图均是二分图。**

**所以二分图的难点就在于 建图**

* + 1. 二分图的性质

**(1) 最大匹配数 == 最小覆盖数。**

**(2) 独立数 == 顶点数 - 最大匹配数。**

**(3) 最小边覆盖 == 图中点的个数 - 最大匹配数 == 最大独立集。**

**(4) 最大匹配数 = 左边匹配点 + 右边未匹配点**

**Hall定理：**

**此定理使用于组合问题中，二部图G中的两部分顶点组成的集合分别为X, Y, X={X1, X2, X3,X4,.........,Xm}, Y={y1, y2, y3, y4 ,.........,yn},G中有一组无公共点的边，一端恰好为组成X的点的充分必要条件是：X中的任意k个点至少与Y中的k个点相邻。（1≤k≤m)**

**本论还有一个重要推论：二部图G中的两部分顶点组成的集合分别为X,Y, 若∣X∣=∣Y∣,且G中有一组无公共端点的边，一端恰好组成X中的点，一端恰好组成Y中的点，则称二部图G中存在完美匹配。若图G的每个点度数为t，则称二部图G为t---正则的二部图存在完美匹配。**

1. 方法归纳
   1. 二分图的判定

常用 **DFS** 进行 **染色**，**不存在奇环就有二分图**

只有分析出来它可以是一个二分图（看我们如何构造的图），才能用二分图的一些性质来解题。

* 1. 最大匹配，最小点覆盖，最大独立集,最大团，最优匹配
     1. 常用的解决方法

1. **最大匹配**模型的**‘0’，‘1’**要素（即节点分为两个独立的集合，每个集合内部有**0** 条边，每个节点只能与 **1** 条边相连）
2. **最小点覆盖**模型的 **‘2’**要素：即每一条边 有 **2** 个端点，二者**至少选择一个**
3. **最大独立集**：与最小点覆盖集 **互补**
4. **最大团**：**建立** G 的 **补图** G’ ，**求** 得的 **G’ 的 最大独立集** 就是 G 的最大团
5. **一对多的匹配**：
   1. 方法一：int match[N] ---> vector<int> match[N]，match[v] != 0 --->match[v].size() < limit[v]（增大match 的容量来，来增多匹配）
   2. 方法二：**将 集合内的每一个点拆成 多个点 来用**
6. **最优匹配**：
7. 通常用 **KM**算法，**网络流** 算法
8. KM 算法 运行的**首要条件**是**必须存在一个完备匹配**，如果求一个最大权匹配(不一定完备)可以把不存在的边权值赋为0。（**建立虚拟边**）
9. 求二分图的**最大权**完备匹配： KM 算法
10. 求二分图的**最小权**完备匹配：只需将所有的边权值取其相反数，求最大权完备匹配后，得到的值再取相反数即可
11. 求二分图的**最大积**完备匹配：每条边权取自然对数，然后求最大和权匹配，求得的结果a再算出e^a就是最大积匹配
12. 完备匹配的这一条件很重要
    * 1. 常用的**建图**方法

基本也都发现了，匹配问题，难就难在建图上，给出几个建图的角度：

1. **奇偶染色法**：常用于 第一类 填格子
2. **行列拆分法**：常用于 第二类 填格子
3. **行列组合拆分**：常用于 第二类 填格子
4. 从**所求最优化问题**入手， 将 所求问题 看做 **实体** 去找 关系（就是谁能跟其是一个集合，谁不能和 其 一个组合），**根据关系去连边建图**
5. 遇到有向图（或者无向图）时，通常就是将 每一个点分边作为 X，Y集合，按照给定的边去**连边建图**，找规律，然后猜方向，在去证明。（想不到就试一试呗）
6. 有时需要用到 **反建法**：所给条件的正面建出来的图没什么意义，就考虑用条件反面建图
   1. 最小路径覆盖
      1. 每一个点只能经过一次（DAG上不可相交路径）
7. **拆点**：**将每一个顶点i拆成两个顶点Xi和Yi。然后根据原图中边的信息，从X部往Y部引边。所有边的方向都是由X部到Y部。**
8. 因此，所转化出的二分图的最大匹配数则是原图G中最小路径覆盖上的边数。
9. 因此由最小路径覆盖数 ＝ 原图G的顶点数 － 二分图的最大匹配数便可以得解。
   * 1. 每一个点可以经过多次（DAG上可相交路径）
10. 先对DAG **求传递闭包**，得到新图 **G’**
11. 再求 **G’** 的 “1.3,1”
    * 1. 无向图的最小路径覆盖
12. 和 1.3.1 中一样的 **拆点建图**
13. 无向图最小路径覆盖数 = 顶点数 - 二分图最大匹配数 / 2。
    1. 题型分析
       1. 第一类填格子

主要方法：**奇偶染色法**、**拆行列法①**

* **例1、骑士放置（CH6901）**

题意：在没有“别马脚”且有些格子不能放置的情况下，在棋盘下最多能放多少 “马”。

方法：**奇偶染色法**

画个图：

|  |  |  |  |
| --- | --- | --- | --- |
| (1,1) | (1, 2) |  |  |
|  |  | (2, 3) | (2,4) |
| (3,1) | (3, 2) |  |  |
|  |  |  |  |

所谓奇偶染色法，就是理论按上图的方式将 棋盘格子 P(x, y) 中**按 (x + y) 的奇偶**进行染色，不妨偶数的为白格子（作为二分图的X部），奇数的为黑格子（作为二分图的 Y 部）。

**适用条件：** 主要适用于 第一类填格子，所谓第一类填格子，**模型发生在棋盘上（或类似棋盘）**，且棋盘的范围不大（N,M <= [50,1000]），且 **每一个棋子 的放置 只受某一小些特殊棋子的影响**

**解题方法：** **随意考察几个特殊位置的棋子，看每一个棋子能放置的条件，或者放完该棋子之后产生什么影响，或者该棋子的放置与哪些棋子是有矛盾的与哪些棋子是没有矛盾的。建图的时候通常只需要考虑X部或者Y部就行，建的是单向边从 X ---> Y**

回到这个题上：我们考虑 (1, 1) 这个点，(1, 1)与 (2, 3)、(3, 2)两个点是不能同时放棋子的，（1， 2）与 （3， 1）、（2， 3）是不能同时放置的，任意一点 P(x, y) 不能与

P1(X ± 1，Y ± 2)，P2(X ± 2, Y ± 1) 不能同时放置，而 考虑 P 与 P1、P2所处格子的颜色，因为 X + Y ± 1 ± 2 = X + Y ± 1或者 X + Y ± 3，一个数加减奇数一定与原数的奇偶性不相同，故P 与 P1、P2所处格子的颜色一定是不相同的。故只需将 P(不妨是百格的) 与 其**相矛盾**且**可以放置棋子**的格子连一条有向边，这样建图发现，题目所求的就是求一个最大的不互相矛盾格子集合，即相当于求 **最大独立集**。

* **例2、POJ 3041-Asteroids**

题意：有一个N\*N的网格,该网格有K个障碍物.你有一把武器,每次你使用武器可以清除该网格特定行或列的所有障碍.问你最少需要使用多少次武器能清除网格的所有障碍物?

方法：拆行列法①

画个图：

|  |  |  |  |
| --- | --- | --- | --- |
| P(1,1) | √ | √ | √ |
| √ |  |  |  |
| √ |  |  |  |
| √ |  |  |  |

所谓拆行列法①，就是在奇偶覆盖法上的延伸，将行、列 分别 拆成 X、Y 两部， 然后按照 满足题意的格子P(X, Y) 按 X ---> Y 进行 连边建图。

适用条件： 模型发生在棋盘上（或类似棋盘），且棋盘的范围不大（N,M <= [50,1000]），且 每一个棋子的放置 会影响 到同行同列 的其它元素，即**单个棋子只能属于某一特定的 一整行 或者 一整列。**

解题方法：将行、列 分别 拆成 X、Y 两部， 然后按照 满足题意的格子P(X, Y) 按 X ---> Y 进行 连边建图，在 题目的意思下，看求的是二分图的哪个东西。

回到这个题上：在 P(1, 1) 上放置一个 武器，就相当于 第1行不用在放上 武器了，或者第一列不用在放上武器了。故，我们在 连一条 1 ---> 1的边， 将每一个可行点（x, y）（有障碍物的点）连 一条 x ---> y的有向边，连边建图后发现，就想相当于每一条边（对应着一个格子）的两个点（对应其所处的行和列），那所有的边（障碍物）都得选上，所选的点（所在 行或列上的 武器尽可能的少），也即是求最小顶点覆盖。

* + 1. 第二类填格子

概述：所谓第二类填格子，就是在第一类填格子上的进一步延伸，在第一类填格子中，是 一个棋子 的放置会影响一整列或者一整行，而第二类填格子中加入新的限定，一个棋子的放置 也影响到一行或者一列，但不是影响一整行或一整列，只是影响到这一行的或这一列的某一端点

画个图：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (1,1) |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| (4,1) |  |  |  |  |
| (5,1) |  | (5,3) |  |  |

假设黑色的格子为端点，即 可以想成 (1,1) 和 (4,1) 上可以同时放置一个 “車”，(5,1) 和（5，3）也可以同时放上一个“車”

**解决方法**：**拆行列法②**

将上图转化为两个图

1、表示同类编号的格子只能放一个“車”（即相当于将有矛盾的格子放在一个集合里）

Flagx[ ][ ]

(按行划分)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 2 |  | 3 | 3 |
|  | 4 |  | 5 | 5 |
| 6 | 6 | 6 | 6 |  |
| 7 |  | 8 | 8 | 8 |

2、表示同类编号的格子只能放一个“車”（即相当于将有矛盾的格子放在一个集合里）

Flagy[ ][ ]

(按列划分)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 6 | 7 |
| 1 | 3 |  | 6 | 7 |
|  | 3 |  | 6 | 7 |
| 2 | 3 | 5 | 6 |  |
| 2 |  | 5 | 6 | 8 |

3、例 P（4， 2）就被映射成 P’(flagx[4][2], flagy[4][2]) = P’(6, 3)，在建图时 建一条

6 ---> 3 的有向边，同理 对于其他合理的点，进行连边建图。

* **例3、A-HDU 1045-Fire Net**

题意：是给出一张图，图中'X' 表示wall，‘ **.** ’表示空地,可以放置blockhouse

同一条直线上只能有一个blockhouse，除非有wall隔开，问在给出的图中最多能放置多少个blockhous？

解题思路：按照**行列拆分法②** 进行建图。可以发现这就是最小顶点覆盖问题。

* + 1. 第三类填格子

概述：所谓第三类填格子，就是不是第一、二类填格子的填格子，即对填格子更一步延伸。

解决方法：这一类题一般都是有点像，第一类或第二类，但又不全是，需要逻辑转换一下，**联系着第一，二类来建图**，然后在来**凑答案**

* **例4、E-HDU 2819-Swap** 题意：给定一个n\*n的矩阵，格子数字0或者1，通过交换两行或两列使对角线都是1。若不能，输出-1；若可以，输出交换次数，并且输出交换的行或者列。

解题思路：首先分析 设计到 行列交换，肯定就不属于第一、二类填格子了，它还是一个格子（矩阵）的问题。题目最后要求是将 矩阵变为对角线全为1的矩阵，可知最后的矩阵一定是一个满秩矩阵，又因为交换行列不会影响到秩，故 只交换行或者只交换列都是可以换出来 对角线全为1，若只交换行都不行，那么行列组合在一起交换也无法成功。 若只考虑交换列，那就是将 第i行对应的第j列交换到 第i列，使P(i,i) = 1。

故可以发现 “0”要素是 每两行之间没有啥影响， “1”要素是 i---> j (p[i][j] == 1)

连边建图后，发现就是一个最大匹配，最后输出路径时，需要注意一下

* **例5、H-POJ 3020-Antenna Placement**

题意：一个矩形中，有N个城市’\*’，现在这n个城市都要覆盖无线，若放置一个基站，那么它至多可以覆盖相邻的两个城市。问至少放置多少个基站才能使得所有的城市都覆盖无线？

解题思路：分析完之后，发现好像束手无策，那只能试一试常用的建图方法，这个题和 **奇偶染色法** 还挺相似的，按照那尝试建图，建好图之后强行凑一下，ans = size(X) + size(Y) – 无向图的最大匹配/2（为什么是无向图是对的， 因为同一个图的无向图的形式的最大匹配 不一定 正好是有向图最大匹配的二倍），然后发现这与无向图的最小路径覆盖的公式一致，最后在自圆其说。

* + 1. 多重匹配

概述：多重匹配有三种解决方法，本专题就不用网络流的算法了，分别对其他两种进行举例分析就行。

* **例6、CH 6803 导弹防御塔**

题意：给定 N 座塔，M 个怪物，每座塔一次可以发射一枚导弹，发射导弹有发射时间和冷却时间，每座塔和每只怪物有自己的二维坐标，所有导弹有一个共同的速度V求至少需要多长时间才能将所有怪物消灭。

思路：求最短时间，很显然可以二分时间，在给定的时间，若在规定时间t下，每座塔所能发射的炮弹集合作为X部，怪物们作为 Y 部。若 Y 部能被全部匹配， 则t 是满足条件的。而建图的时候只需要，将每一个炮塔根据t，拆成新的顶点，然后与t内能打到的怪物连边建图。跑一个最大匹配就行。

* **例7、M-POJ 2289-Jamie's Contact Groups**

题意：一个人通讯录中好友有许多，然后需要分组，现在告诉你不同的的人能分进小组的编号，然后问你怎么分配使小组中人最多的人尽可能小，输出最小值。

思路：二分最小值，判断是否能完全匹配

* + 1. 最优带权匹配

概述：带权匹配解决方法也不多，就着KM算法建图讲一个例题

* **例8、Q-HDU 3488-Tour**

题意：题目告诉我们一个有向图，现在问将图中的每一个点都划分到一个环中的最少代价是多少？每条边都有一个代价。

解法：由于要成环，那么将这个图进行拆点，就变成了单向的二分图了，此时一个**完备匹配就是一种连线策略**，只要保证没有边是和自己相连，就能够满足题目中要求的每个点至少属于一个环。证明也是很简单的。因为我们总可以从一个完备匹配中找出起点，然后再从匹配点作为起点找......

* + 1. 优秀题目清单（附有代码）

1. **例1~8**
2. **B-HDU 2444-The Accomodation of Students**
3. **D-HDU 1281-棋盘游戏**
4. **F-HDU 2389-Rain on your Parade**
5. **J-HDU 1151-Air Raid**
6. **K-POJ 2594-Treasure Exploration**
7. **L-HDU 3829-Cat VS Dog**
8. **V-POJ 2771-Guardian of Decency**
9. 常用模板
10. 二分图的判定（B-HDU 2444）
11. 匈牙利算法（B-HDU 2444）
12. HK算法（F-HDU 2389）
13. 扩展匈牙利算法（例7）
14. KM算法（例8）
15. 带点权的最优匹配
16. 带花树
17. 总结

二分图匹配，主要是解决 最优化 的 问题，一般数据规模不大。

建图的时候，主要是思考，题目中的各实体，以及各实体之间的关系，将待求实体合理地放入X部、Y部，按实体间的关系连边建图，寻找规律。

有时候也实在有不知所措的时候，就把常用的建图方法套上来，然后强行推出答案，最后在自圆其说。

二分匹配还是难在建图！