数据结构

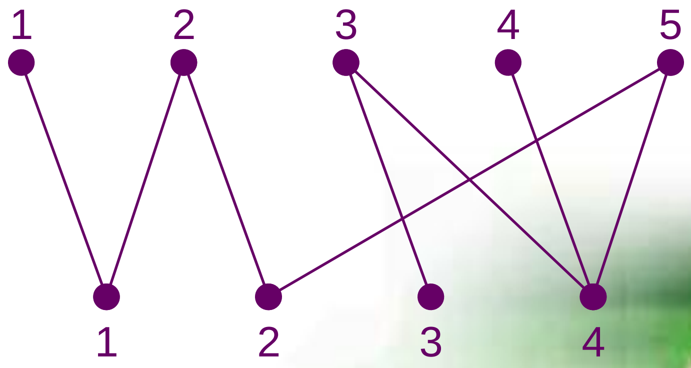
授课人：长沙市一中 周祖松

问题引入:设有M个未婚男青年x1, x2, …, xm，和N个未婚女青年y1, y2, …, yn，现已经知道每个人所喜欢对象（可以一对多），如何安排才能使尽可能多的人结为夫妻。

二部图的概念

§ 二部图又称作二分图，是图论中的一种特殊模型。

§ 设G=(V,{R})是一个无向图。如顶点集V可分割为两个互不相交的子集，并且图中每条边依附的两个顶点都分属两个不同的子集。则称图G为二分图。



最大匹配

- 给定一个二分图G，在G的一个子图M中，M的边集{E}中的任意两条边都不依附于同一个顶点，则称M是一个匹配。
- 选择这样的边数最大的子集称为图的**最大匹配问题**(maximal matching problem)
- 如果一个匹配中，图中的每个顶点都和图中某条边相关联，则称此匹配为**完全匹配**，也称作**完备匹配**。

匈牙利算法

- 求最大匹配的一种显而易见的算法是：先找出全部匹配，然后保留匹配数最多的。但是这个算法的复杂度为边数的指数级函数。因此，需要寻求一种更加高效的算法。
- 增广路的定义(也称增广轨或交错轨)：
- 若P是图G中一条连通两个未匹配顶点的路径，并且属M的边和不属M的边(即已匹配和待匹配的边)在P上交替出现，则称P为相对于M的一条增广路径。

匈牙利算法

§ 由增广路的定义可以推出下述三个结论：

§ 1－P的路径长度必定为奇数，第一条边和最后一条边都不属于M。

§ 2－P经过取反操作可以得到一个更大的匹配M’。

§ 3－M为G的最大匹配当且仅当不存在相对于M的增广路径。

基本思想

从图 G 的任意一个对集 M 开始，若 M 饱和 S 的所有点，则 M 是 G 的最大匹配；否则，由 S 的 M- 非饱和点出发，用一个系统方法搜索一条 M- 增广路 P。若 P 存在，则通过交换 P 在 M 和不在 M 中的边，便得到一个其基数增加 1 的对集，然后从新的对集开始，继续迭代。若 P 不存在，则现行的对集就是 G 的最大匹配。

匈牙利算法

算法轮廓：

§ (1)置M为空

§ (2)找出一条增广路径P，通过取反操作获得更大的匹配M'代替M

§ (3)重复(2)操作直到找不出增广路径为止

算法步骤

第 1 步 (开始)给定二分图 G=(S,T,E)，令 M 是一个任意对集，可能是空对集，这时没有点被标号。

第 2 步 (标号) (2.0) 在 S 中，每个非饱和点给以标号“0”。

(2.1) 如果不存在未检查的标号点，转向第 4 步；否则，找一个具有未检查的标号点 i，如果 $i \in S$，转向（2.2）；如果 $i \in T$，转向（2.3）

(2.2) 检查点 i 的标号如下：对每个同点 i 关联的边$\{i,j\}$，除非 j 已经被标号；否则，给点 j 标号“i”，转向(2.1)。

(2.3) 检查点 i 的标号如下：如果点 i 是非饱和点，转向第 3 步；否则，辨认同点 i 关联的属于 M 的唯一边$\{i,j\}$，给点 j 标号“i”，转向(2.1)。

第 3 步 (增广)

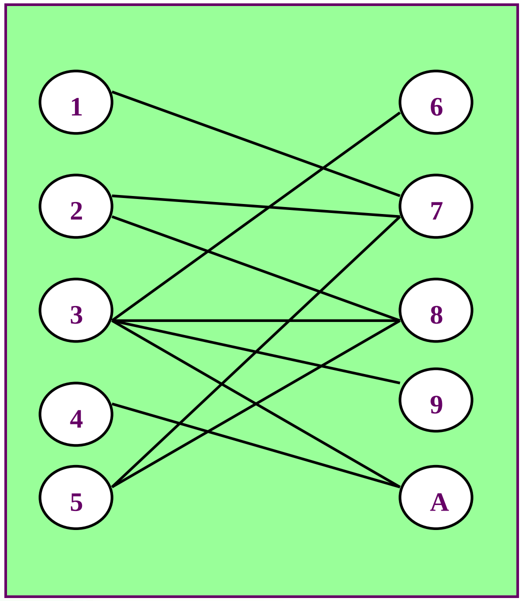
终止在 i 的一条增广路被找到，通过方向追踪辨认在路上点 i 的前点，通过把路上不在 M 中的边加入 M，而把路中在 M 中的边从 M 中除去来增广 M，抹掉所有标号，转回(2.0)。

第 4 步 (匈牙利标号)

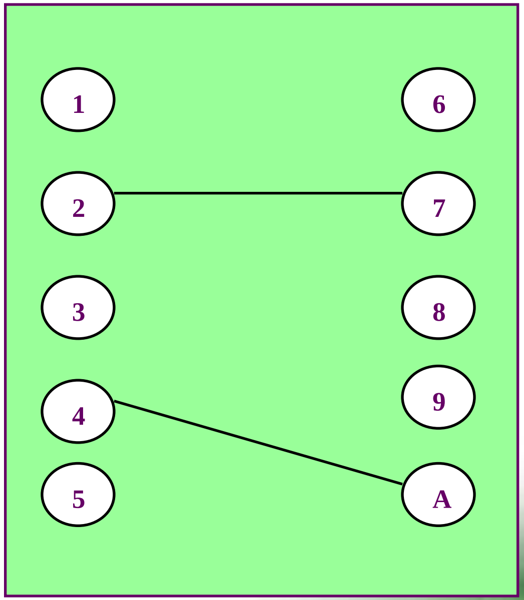
标号是匈牙利的，这时没有增广路存在，M 是最大基数对集。令 $L \subseteq S \cup T$，表示所有标号点的集合，则 $K=(S-L)\cup(T\cap L)$ 是对偶于 M 的最小覆盖。

算例

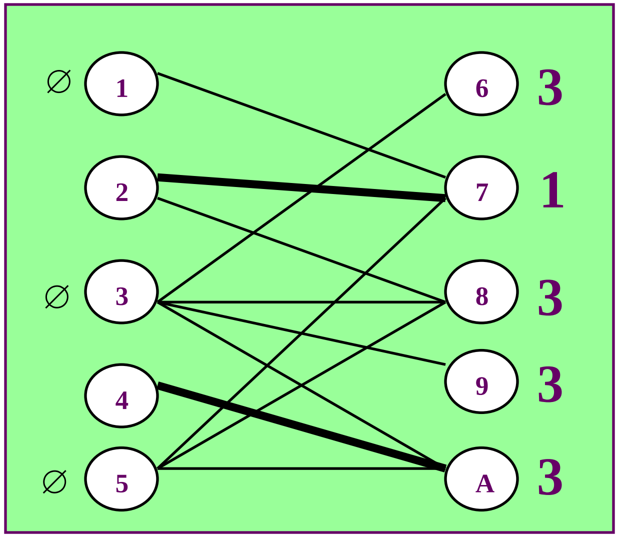
求下图a所示的二分图G的最大基数对集，若初始解如下图b所示



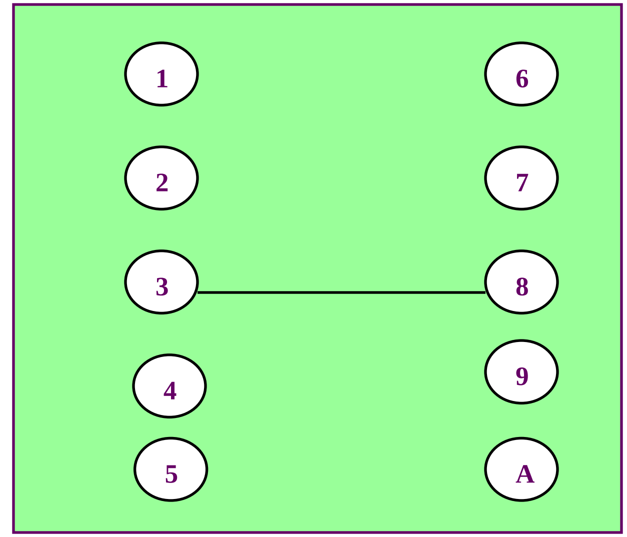
a



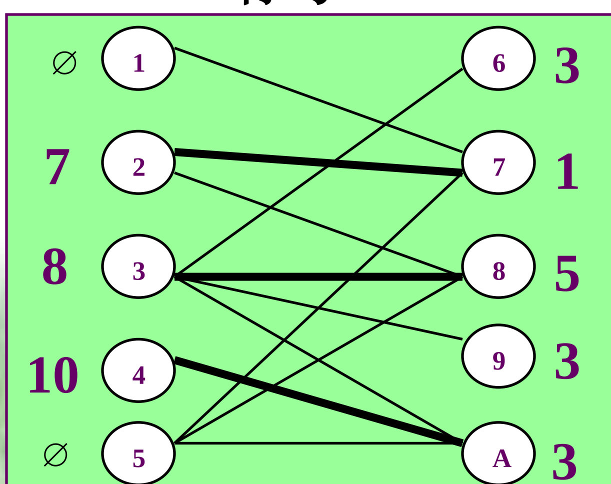
b


∅ 1
2
∅ 3
4
∅ 5
6 3
7 1
8 3
9 3
A 3

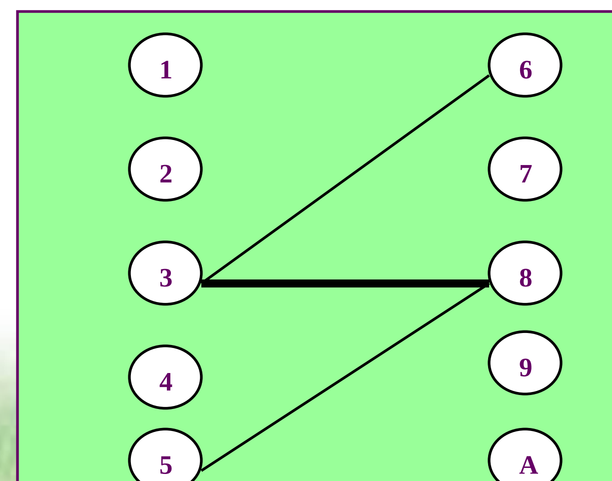
标号


1
2
3
4
5
6
7
8
9
A

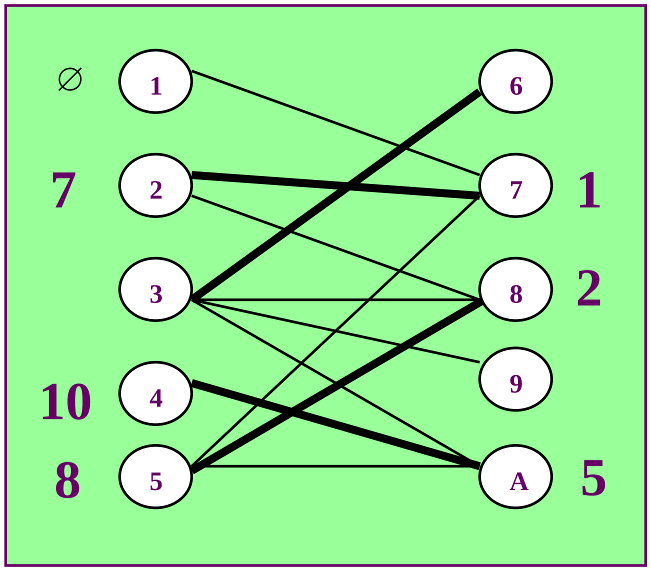
增广路


∅ 1
7 2
8 3
10 4
∅ 5
6 3
7 1
8 5
9 3
A 3

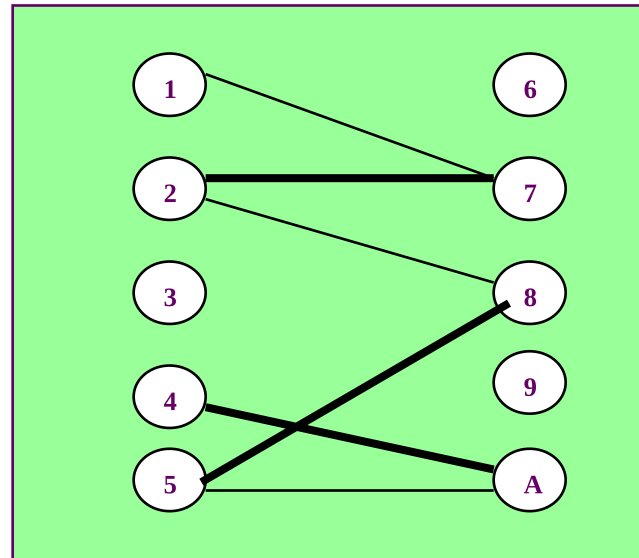
标号


1
2
3
4
5
6
7
8
9
A

增广路



标号



增广路

```
int find(int i);  //从i出发，找增广链
{  int  k, q;
   for(k=1;k<=M;k++)
       if( map[i,k] && !vis[k])
       {    vis[k] = 1;
            if (link[k]==0|| find(link[k]))
             {       link[k] = i; return (1) ;}
       };
    return (0);
}
main()
{  int map[ maxn, maxn],link[ maxn] ;     //保存匹配边
   int i,n,res=0; vis[ maxn]   init;  //输入
   memset(link,0,sizeof(link)) ;
   for(i=1;i<=n;i++)
   {    memset(vis,0,sizeof(vis)) ;   //标记一个点是否为被访问
        if (find(i))   res++;
    }
   print;   //输出
   return 0;
}
```

算法复杂性

若令$|S|=m$，$|T|=n$，且$m\le n$。

在找到一个匈牙利树或找到一条增广路之前，

标号程序最多进行$O(mn)$次；

在求出所需的对集之前，初始对集最多能增广 m 次；

所以，总的计算量为$O(m^2n)$。

最小点覆盖数

- 选取最少的点，使任意一条边至少有一个端点被选择

最佳匹配

§ 如果边上带权的话，找出权值和最大的匹配叫做求最佳匹配。

§ 实际模型：某公司有职员$x_1,x_2,\ldots,x_n$,他们去做工作$y_1,y_2,\ldots,y_n$,每个职员做各项工作的效益未必一致，需要制定一个分工方案，使得人尽其才，让公司获得的总效益最大。

§ 数学模型：G是加权完全二分图，求总权值最大的完备匹配。

算法步骤

第 1 步 (开始)给定二分网络 $G=(S,T,E,W)$，令 $M=\phi$， $w=\max\{w_{ij}\}$，对每个 $i\in S$，令 $u_i=w$，对每个 $j\in T$，令 $v_i=0$ 和 $\pi_j=+\infty$，这时没有点被标号。

第 2 步 (标号)(2.0) 给 S 中每个非饱和点标号“ϕ”。

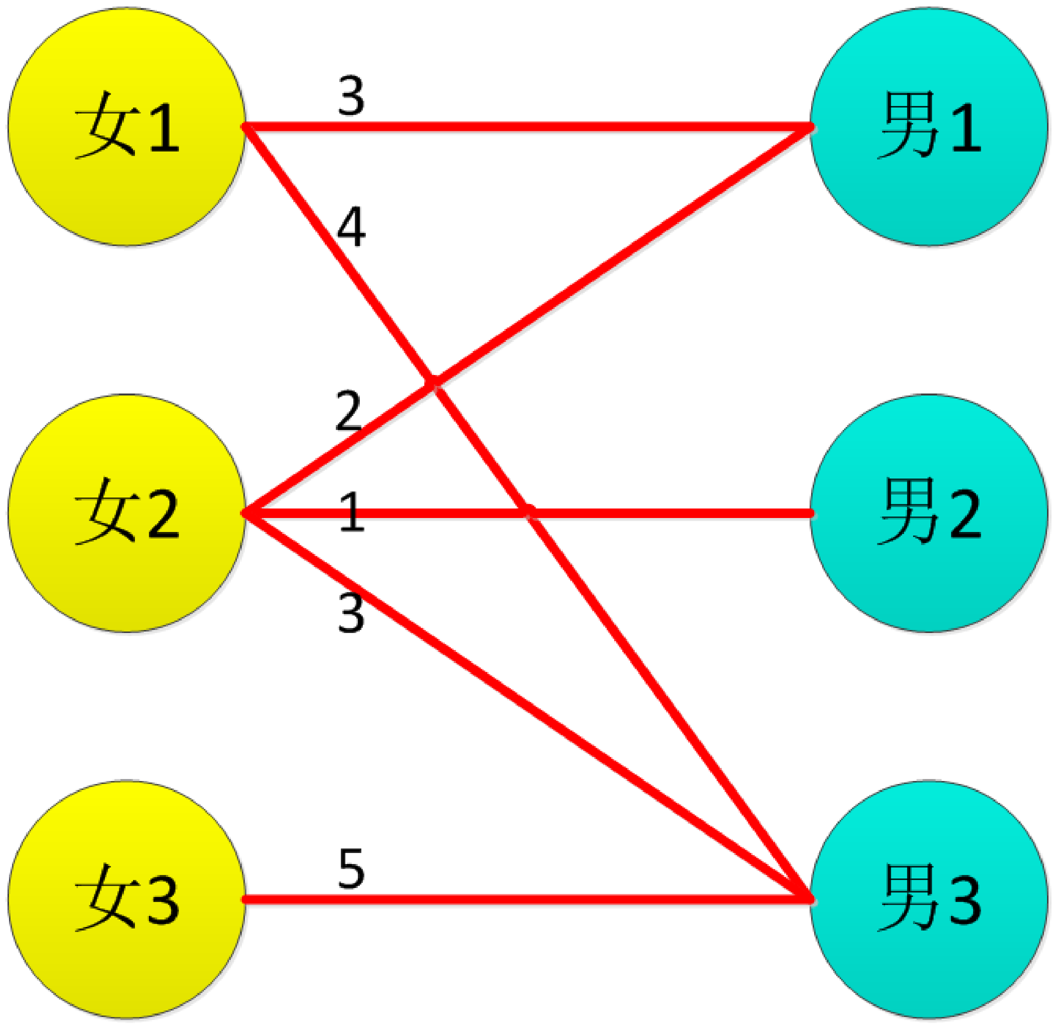
(2.1) 如果不存在未检查的标号点或者存在未检查的标号点，但每个未检查的标号点 $i\in T$ 有 $\pi_i>0$，则转向第 4 步。

(2.2) 找一个未检查的标号点 i，其中或者 $i\in S$，或者 $i\in T$ 且 $\pi_i=0$，如果 $i\in S$，则转向(2.3)；如果 $i\in T$，则转向(2.4)。

(2.3) 检查点 i 的标号如下：对每条边 $\{i,j\}\notin M$，如果 $u_i+v_j-w_{ij}<\pi_j$，给点 j 标号“i”，并令 $\pi_j=u_i+v_j-w_{ij}$，转回(2.1)。

(2.4) 检查点 i 的标号如下：如果点 i 是非饱和点，转向第 3 步；否则，辨认唯一边 $\{i,j\}\in M$，给点 j 标号“i”，转向(2.1)。

第 3 步 (增广)终止在 i 的一条增广路被找到，通过方向追踪辨认在路上点 i 的前点，通过把路上不在 M 中的边加入 M，而把路中在 M 中的边从 M 中除去来增广 M。对每个点 $j\in T$，令 $\pi_j=+\infty$。抹掉所有标号，转回(2.0)。


女1
女2
女3
男1
男2
男3
3
4
2
1
3
5

BZOJ1191HNOI2006、

BZOJ1562**[NOI2009]变换序列**

BZOJ1520、BZOJ1937