一

一

1. 给出K_5-e与$K_{3,3}-e$的一种平面嵌入，其中e 为K_5或 $K_{3,3}$的任意一条边
2. 证明：K_5不是平面图

利用平面图的性质:

若G是连通的简单平面图，则有$v-\varepsilon+\phi=2$，$\varepsilon\leq 3v-6$，$\delta\leq 5$

K_5中，$10=\varepsilon>3v-6=3*5-6=9$，矛盾

一

> 拓展：证明：$K_{3,3}$不是平面图

$4\phi \leq \sum deg(f) = 2\varepsilon = 18 \Rightarrow \phi \leq 4$

$2 = v - \varepsilon + \phi \leq 6 - 9 + 4 = 1$，矛盾

平面图

1. 平面图性质
2. 极大平面图定义和性质

> G是极大平面图当且仅当G的平面嵌入的每个面都是三角形

> G是极大平面图当且仅当$\varepsilon = 3v - 6$

> G是极大平面图，则$\delta \geq 3$

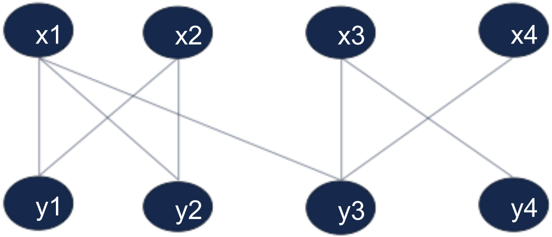
3. 图的厚度定义和性质

二 二

某公司有4个员工x_1, x_2, x_3, x_4，四份工作y_1, y_2, y_3, y_4，每个人适合做的工作分别为：

$$
\begin{aligned}
&x_1: y_1, y_2, y_3 \\
&x_2: y_1, y_2 \\
&x_3: y_3, y_4 \\
&x_4: y_3
\end{aligned}
$$

1. 请将该问题抽象为二分图；
2. 以$M = x_1 y_1$为初始匹配求出最佳工作分配方案，给出计算过程



$$\begin{cases} \text{一些定义，包括匹配，覆盖，可增广轨道} \\ Hall\text{定理，}Tutte\text{定理} \\ \text{二分图中,最大匹配}=\text{最小覆盖} \\ \text{求最大匹配的匈牙利算法} \end{cases}$$

三

证明：任意图G的顶点着色的色数$\leq \Delta(G)+1$，其中$\Delta(G)$为G的最大度数

考虑给G的任意一个顶点着色时，v最多有$\Delta(G)$个相邻顶点，至多会使用$\Delta(G)$种颜色，则对v使用第$\Delta(G)+1$种颜色可保障不会有同颜色顶点相邻

四

我们将图G中所有顶点的度数按照从大到小的顺序排列，称为图G的度数序列。证明：

1. $6, 6, 5, 4, 3, 3, 2$ 和 $6, 6, 5, 4, 3, 3, 1$ 都不是简单图的度数序列
2. 若$d_1, d_2, \ldots, d_n$是简单图的度数序列，则$d_1 + d_2 + \cdots + d_n$是偶数

五

假设$P = v_1 v_2 \dots v_k$是简单图G中的最长轨道，且$deg(v_1) + deg(v_k) \geq k$。证明：存在$2 \leq i \leq k$，使得$v_1 v_i$ 与$v_{i-1} v_k$简单图G的边

反证法。

由最长轨道可知，v_1和v_k只能和$v_1, \dots v_k$中的点相连，否则会与最长轨道矛盾

则v_1, v_k会与$v_1, \dots v_k$中的点至少连了k条边

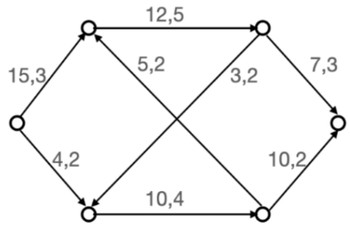
然后再考虑$(v_1 v_2, v_1 v_k), (v_1 v_3, v_2 v_k), \dots (v_1 v_k, v_{k-1} v_k)$，这$k-1$个二元组，如果二元组中的两个都包含在这$k$条边里面，矛盾，所以每个二元组中两个里面只能取一个。又由于只有$k-1$个二元组，但要取出k条边，所以必定会有一个二元组里面的两条边都取到。

六

1. 下图给出了一个网络以及该网络上的一个流函数，请求出一条可增载轨道P，求出可增载的流量 $l(p)$，并且更新流函数
2. 任给网格N的流函数 f，P为关于 f 的可增载轨道， 定义N上新的边权函数 f^* 为：

$$f^*(e)=\begin{cases} f(e) & \text{e不是P上的边} \\ f(e)+l(p), & \text{e是P上的正向边} \\ f(e)-l(p), & \text{e是P上的反向边} \end{cases}$$

证明：f^* 为 N 上的流函数



六

利用流函数的定义

$$(1)\text{任给}e \in E(D),\text{都有}c(e) \ge f(e) \ge 0;$$
$$(2)\text{任给}v \in V(D)-\{s,t\},\text{都有}\sum_{e\in\alpha(v)} f(e) - \sum_{e\in\beta(v)} f(e) = 0$$

利用可增载量的定义

$$l(e) = \begin{cases} c(e)-f(e), e\text{是正向边} \\ f(e), e\text{是反向边} \end{cases}$$
$$l(P) = min_{e\in E(P)} l(e)$$

证明时，

对于流函数定义中的(1)，对于e，讨论e不在P上，e是P上的正向边，e是P上的反向边三种情况

对于流函数定义中的(2)，对于v，讨论v不在P上，v在P上。对于v在P上时，要讨论P上与v相邻的边$e1$和$e2$，$e1 \in \alpha(v)$，$e2 \in \beta(v)$是正向边或者反向边共四种情况（情况还是很多的）

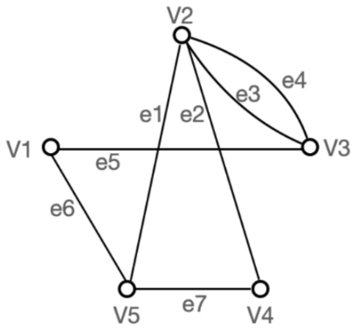
六

由于e_1和e_2都是正向边，$e_1 \in \alpha(v)$，$e_2 \in \beta(v)$，且$\overline{f}(e_1) = f(e_1) + l(P)$，$\overline{f}(e_2) = f(e_2) + l(P)$。对于$\alpha(v)$和$\beta(v)$中其余的边$e$，则有$\overline{f}(e) = f(e)$。因此

$$\begin{aligned}
&\sum_{e\in\alpha(v)} \overline{f}(e) - \sum_{e\in\beta(v)} \overline{f}(e)\\
=&[\sum_{e\in\alpha(v)-\{e_1\}} \overline{f}(e) + \overline{f}(e_1)] - [\sum_{e\in\beta(v)-\{e_2\}} \overline{f}(e) + \overline{f}(e_2)]\\
=&[\sum_{e\in\alpha(v)-\{e_1\}} f(e) + (f(e_1 + l(P))] - [\sum_{e\in\beta(v)-\{e_2\}} f(e) + (f(e_2 + l(P))]\\
=&[\sum_{e\in\alpha(v)} f(e) + l(P)] - [\sum_{e\in\beta(v)} f(e) + l(P)]\\
=&0.
\end{aligned}$$

七

1. 请将下图随意定向成有向图
2. 给出你所定向的有向图的关联矩阵
3. 用有向图的关联矩阵求出下图的生成图个数



定理 10.15. 设G是无环连通无向图，将G的每条边任意定向，得到一个有向图D。则有G的生成树个数为

$$\tau(G) = det(B_f(D) \times B_f^T(D)).$$

ch10

1. 各种定义，包括关联矩阵，邻接矩阵基本关联矩阵，基本圈矩阵，基本割集矩阵等等
2. Warshall算法
3. 求基本圈矩阵，基本割集矩阵
4. 利用有向图关联矩阵求生成树个数

总结，复习要点

1. 概念定义
2. 性质与判定（比如平面图里面的性质，顶点边染色里面的性质和判定）
3. 课本上出现的在考试范围内的算法（重中之重，基本上一定会考），一定要会算，也要注意算法里的步骤和中间输出，不要直接给结果
4. 课本上较简单或中等难度的定理，引理证明；还有一些课本上的例题
5. 作业题，有些很难的就可以不用看了