HW5

1. 证明: 对边e: $u \xrightarrow{e} v$

我们说: e从u出发, e进入v.

考虑:

$$S=\{(v,e) \mid 边e从v出发\}$$

对S用双计数有

- 从v的角度: $|S|=\sum_v d^+(v)$
- 从边e的角度: $|S|=|E|$

(因为每条边只从一个顶点出发)

因此:

$$\sum_v d^+(v)=|E| \qquad (i)$$

类似地, 考虑

$$T=\{(v,e) \mid 边e进入v\}$$

则同样由双计数有

- 从v的角度: $|T|=\sum_v d^-(v)$
- 从边e的角度: $|T|=|E|$

(因为每条边只进入一个顶点)

因此:

$$\sum_v d^-(v)=|E| \qquad (ii)$$

由(i)及(ii)得:

$$\sum_v d^+(v)=|E|=\sum_v d^-(v)$$

□

2. (1) 图略

(2) 设树有n个顶点, l片树叶, 注意, 树叶顶点的度数是1, 因此, 其余$n-l$个顶点度数都≥ 2. 另外, 我们知道树的边数是$n-1$.

由握手定理:

$$2|E|=\sum_v d(v)$$

得:

$$2(n-1)\geq l+2(n-l)=2n-l$$

从而 $l\geq 2$. □

3. (1). 图 $G=(V,E)$.

对任意顶点 $v\in V$，用 $N(v)$ 表示 v 的邻居顶点.

用 C，W 表示两个顶点集合

用 u 表示某个顶点.

初始化：

```
C = ∅,  W = ∅.
u = V 中的任意一个顶点.
C = {u} ∪ N(u)
W = {u}
while (C != W)
{
    u = any vertex in C - W
    C = C ∪ N(u)
    W = W ∪ {u}
    if (C == V)
        break;
}
if (C == V)
    output "连通"
else
    output "不连通".
```

算法效率：在 while 循环中，u 每次都会被设成一个新的顶点. 因此，while 循环的次数最多是 $|V|$ 次；在 while 循环里面需要计算 $N(u)$，这最多须要 $|V|$ 次，因此算法运行最多要 $|V|^2$ 步.

(2) 根据定义：$G=(V,E)$ 与 $H=(V',E')$ 同构

是指存在双射：

$$f: V \longrightarrow V'$$

满足：$(x,y)\in E \iff (f(x),f(y))\in E'$ (*)

算法：用

$$F=\{f: V\to V' \text{ 的双射}\}$$

表示所有 V 到 V' 的双射的集合。

逐一检查 F 中的每一个双射，看其是否满足条件 (*)。

若有某个双射满足，则输出"同构"

否则，输出"不同构"。

算法效率：设 $|V|=|V'|=n$.

则：

$$|F| = \text{从 } V \text{ 到 } V' \text{ 的双射个数} = n! = O\left(n\cdot\left(\frac{n}{e}\right)^n\right).$$

检查每个 $f\in F$ 是否满足 (*) 需要检查所有的 $(x,y)\in V\times V$ 及 $(f(x),f(y))\in V'\times V'$，因此要 $2|V|^2=2n^2$ 次计算。因此，总共须

$$O\left(n\cdot\left(\frac{n}{e}\right)^n\cdot 2n^2\right)=O\left(n^3\cdot\left(\frac{n}{e}\right)^n\right).$$

计算步骤。 □

4. 略。