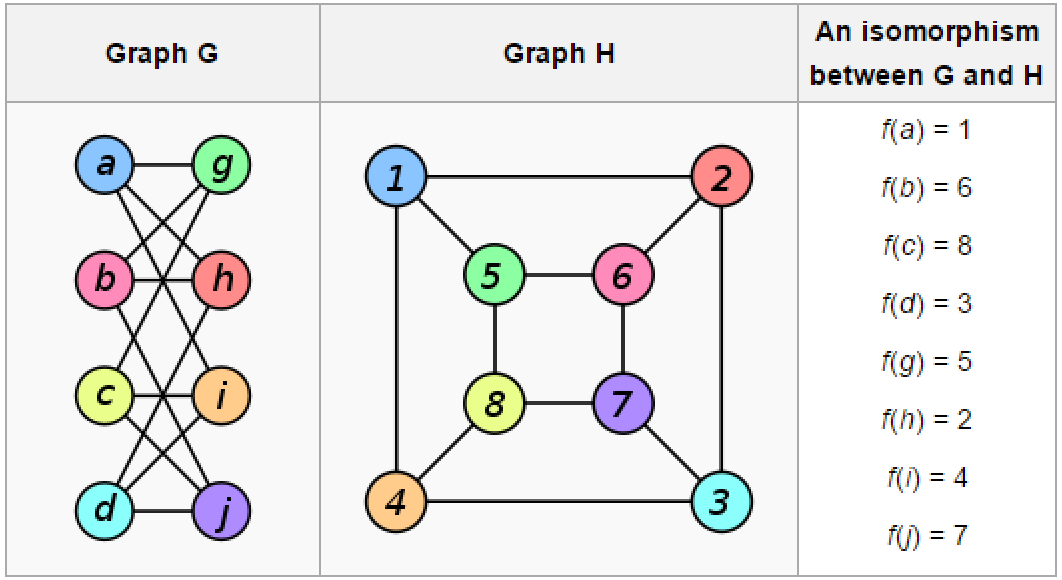
问题求解与实践

——图的同构和树的同构

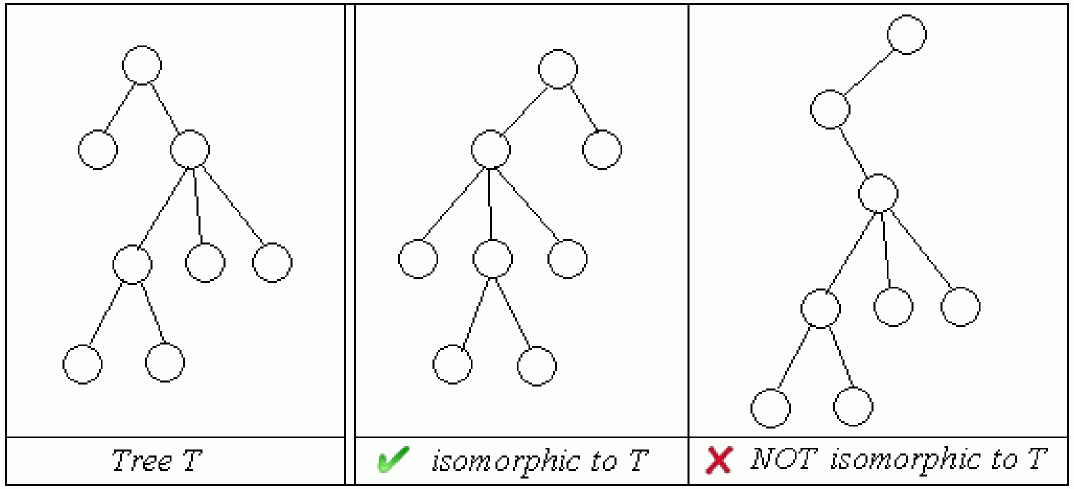
主讲教师： 陈雨亭、沈艳艳

问题描述

- 给出两个有向图，判断是否同构？

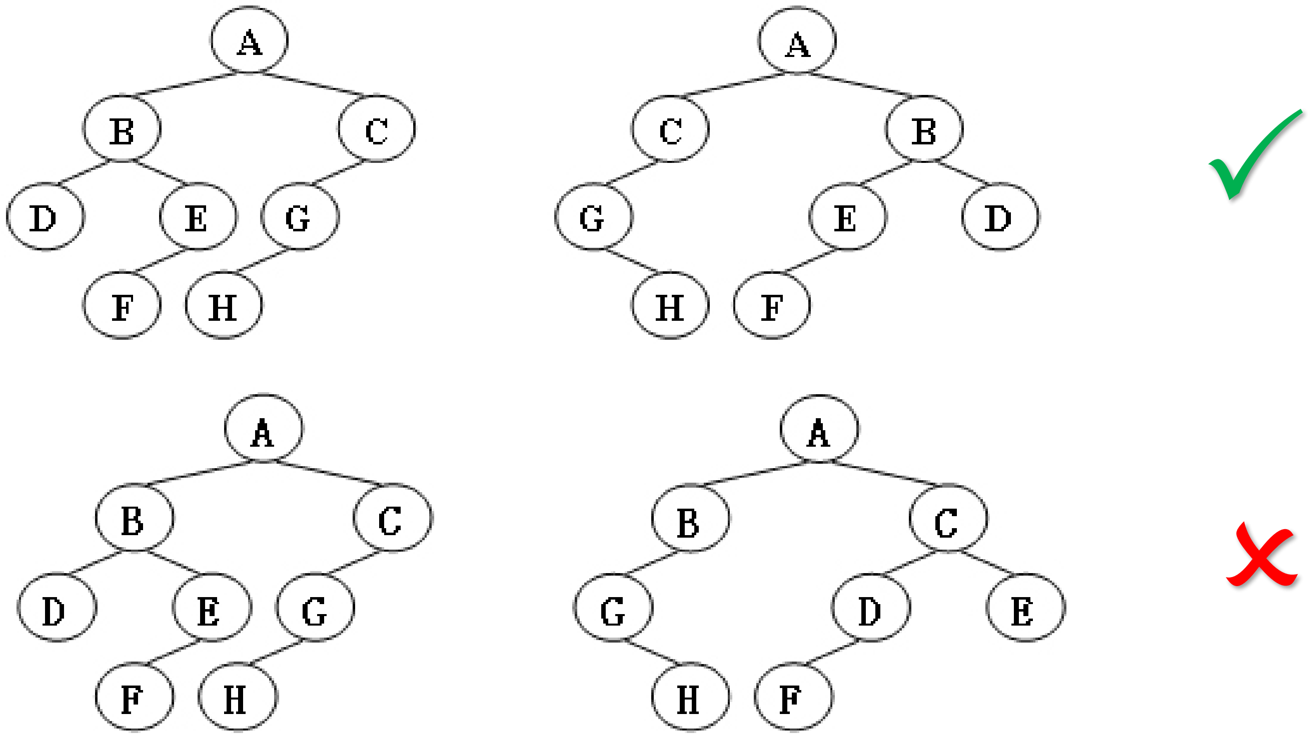


- 给出两棵有根树，判断是否同构？



树的同构（简化版）

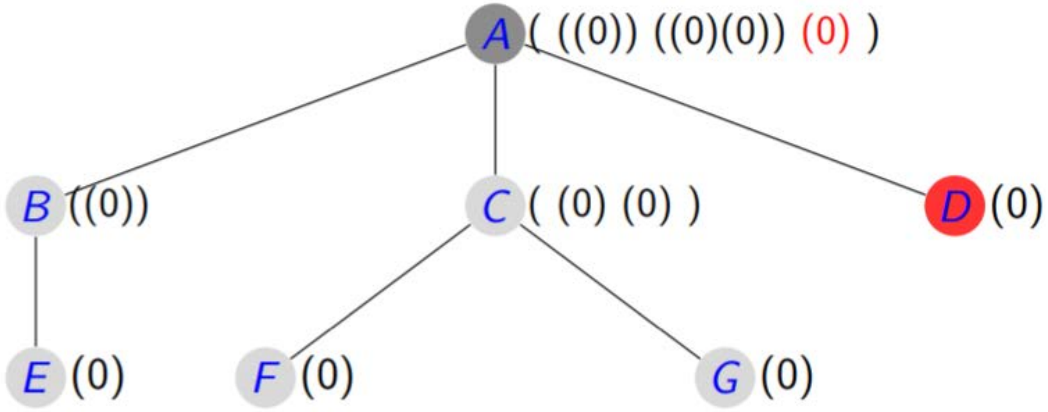
- 给定两棵树T1和T2。如果T1可以通过若干次左右孩子互换就变成T2，则我们称两棵树是“同构”的



思路：设计一个树到序列的映射，保证同构的树映射到相同的序列，不同构的树映射到不同的序列。如果要判断两棵树同构，只需检查它们对应的序列是否相同

树同构的AHU algorithm

- Let’s assign parenthetical tuples to all tree vertices.



ASSIGN-KNUTH-TUPLES(v)

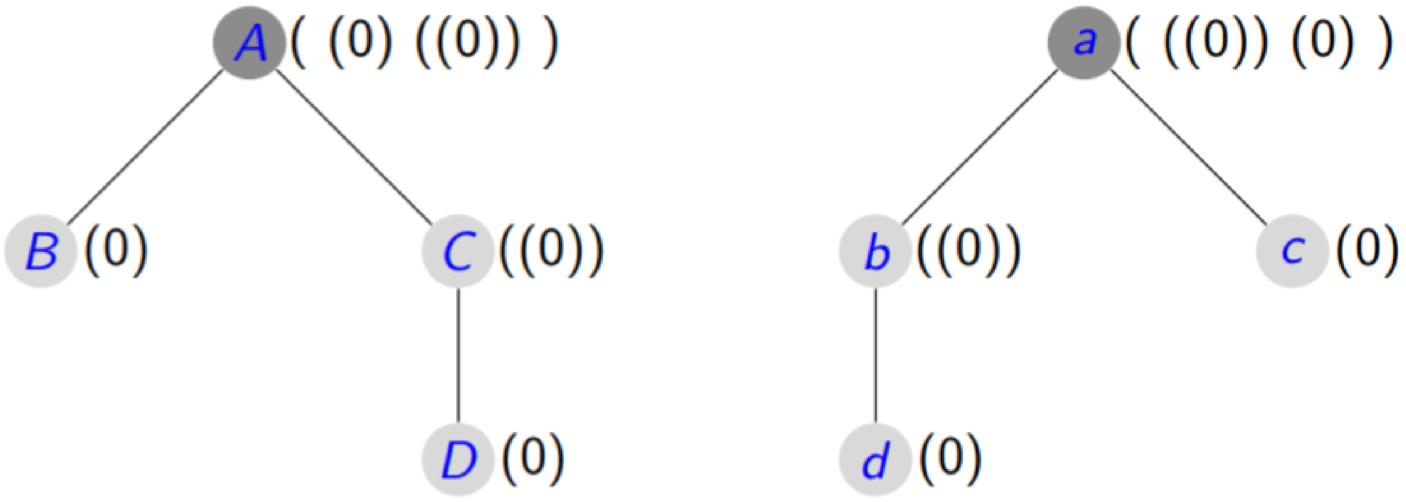
```
if v is a leaf then
    Give v the tuple name (0)
else
    for all child w of v do
        ASSIGN-KNUTH-TUPLES(w)
    end for
end if
Concatenate the names of all children of v to temp
Give v the tuple name temp
```

树同构的AHU algorithm

Observation

There is no order on parenthetical tuples.

Example



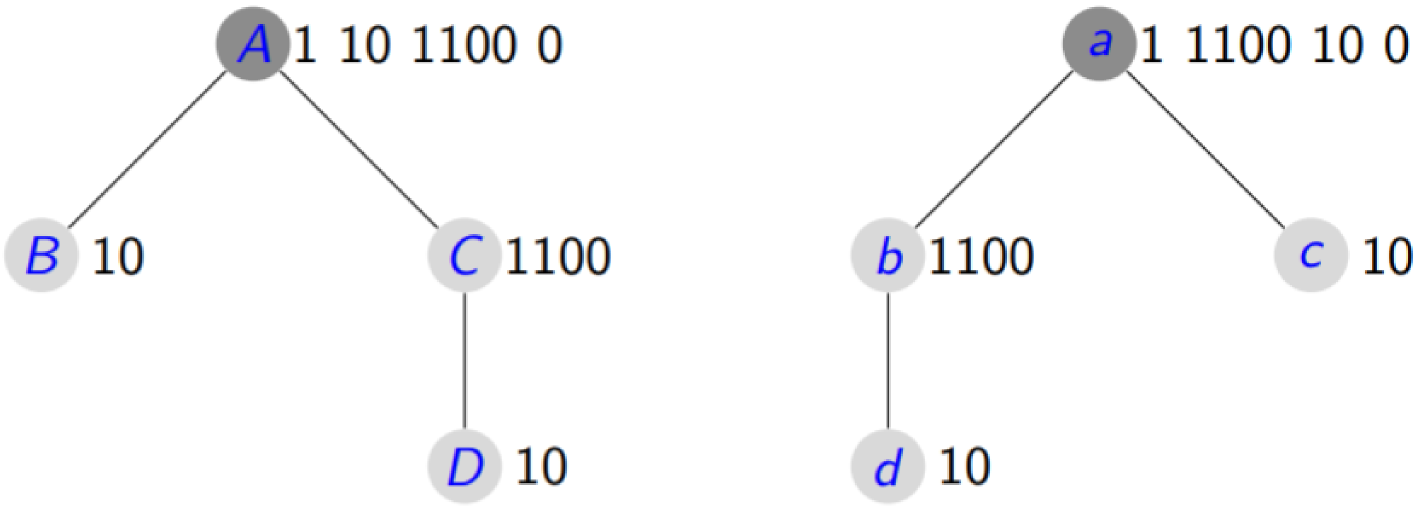
Let's convert parenthetical tuples to *canonical names*. We should drop all "0"-s and replace "(" and ")" with "1" and "0" respectively.

树同构的AHU algorithm

Observation

There is no order on parenthetical tuples.

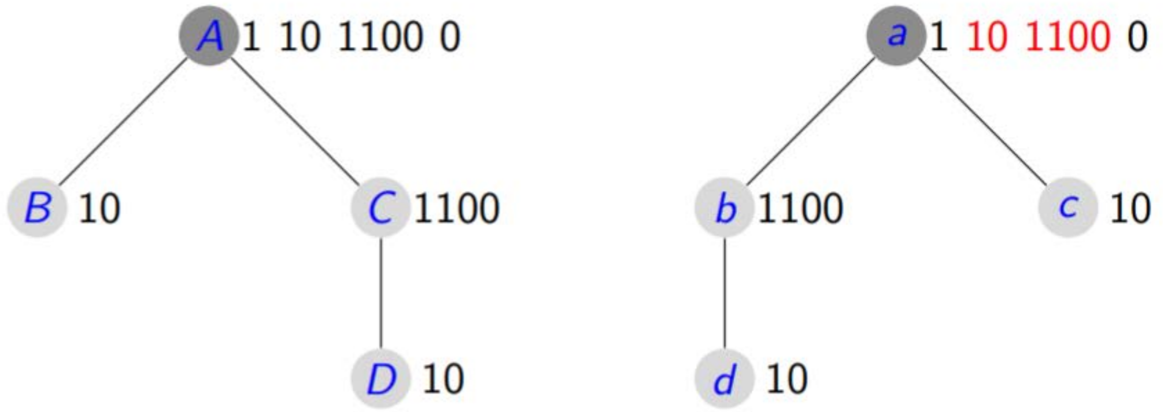
Example



Let's convert parenthetical tuples to *canonical names*. We should drop all "0"-s and replace "(" and ")" with "1" and "0" respectively.

树同构的AHU algorithm

Example



ASSIGN-CANONICAL-NAMES(v)

```
if v is a leaf then
    Give v the tuple name "10"
else
    for all child w of v do
        ASSIGN-CANONICAL-NAMES(v)
    end for
end if
Sort the names of the children of v
Concatenate the names of all children of v to temp
Give v the name 1temp0
```

更详细的讨论，请参考：https://logic.pdmi.ras.ru/~smal/files/smal_jass08_slides.pdf

Quiz：给定一个数字串，如何证明其对应的树均同构？

图的同构

Definition

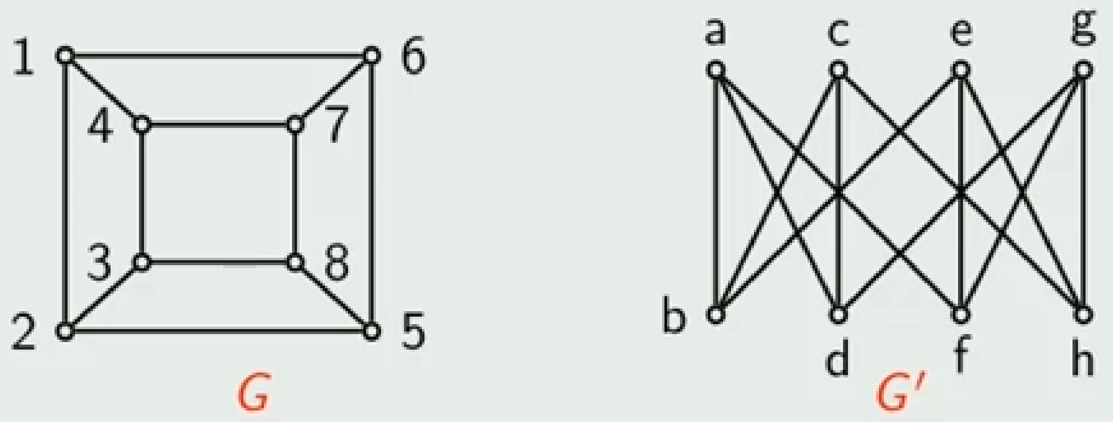
设两个图 $G = < V, E >$ 和 $G' = < V', E' >$，如果存在双射函数$g : V \to V'$，使得对于任意的 $e = (v_i, v_j)$(或者 $< v_i, v_j >$) $\in E$ 当且仅当 $e' = (g(v_i), g(v_j))$(或者 $< g(v_i), g(v_j) >$) $\in E'$，并且 e 与 e' 的重数相同，则称 G 与 G' 同构(isomorphism)，记为 $G \cong G'$。

对于同构，形象地说，若图的结点可以任意挪动位置，而边是完全弹性的，只要在不拉断的条件下，一个图可以变形为另一个图，那么这两个图是同构的。

图的同构

Example

证明下图中 $G \cong G'$。

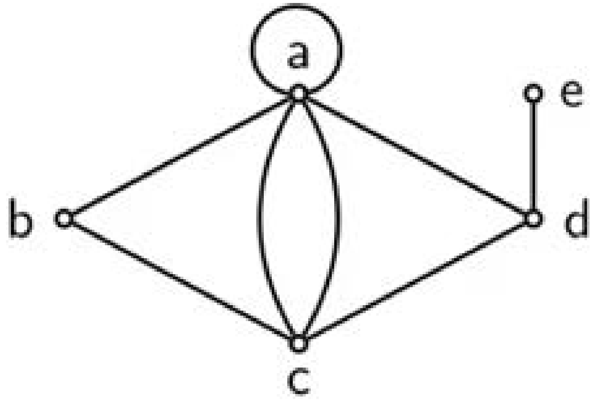


证明：构造结点之间的双射函数 f：

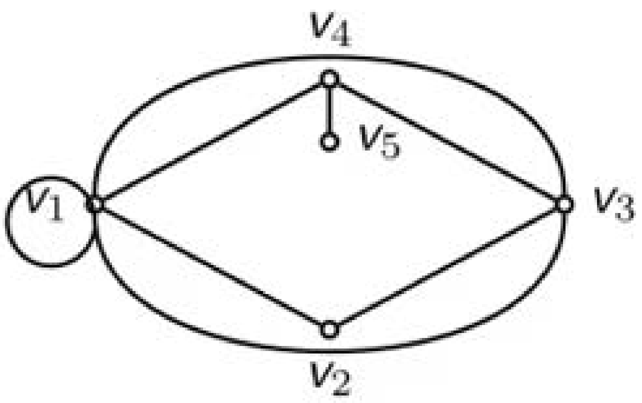
$f(1) = a, f(2) = b, f(3) = c, f(4) = d, f(5) = e, f(6) = f, f(7) = g, f(8) = h$. 容易验证，$f$ 满足图的同构定义，所以 $G \cong G'$。

图的同构

判定同构的方法关键就是找到结点间的对应关系，而在两个带有 n 个结点的图之间有 n! 种可能的一一对应关系。尤其是当 n 很大时，判断任意两个图是否同构常常是一件困难的事情。



G_1



G_2

图的同构

☞ 同构的必要条件

- 结点数目相同
- 边数相同
- 度数相同的结点数相同

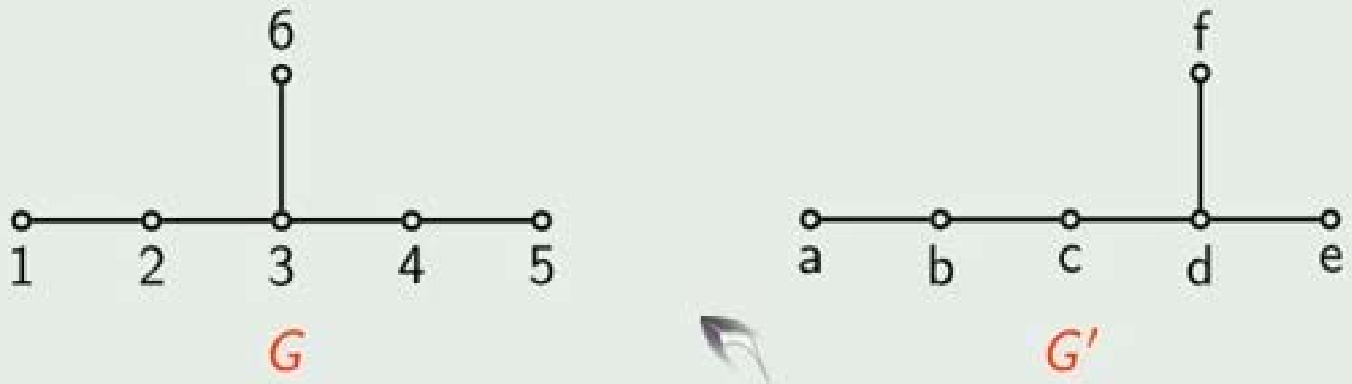
必要条件的应用场景

我们可以通过同构的必要条件说明两个图不同构。

图的同构

Example

下图中 G 和 G' 不同构。



图同构的三个必要条件一定不能作为充分条件来使用。

图的同构

- 图同构是NP问题，但是既没有人找到多项式算法(证明是P问题)，也没有人能证明是NP-complete问题。

- 我们可以用Hash的方法以一定的概率确定两图是否同构
  - 对于图中每一个点i，结合与周边的点关系，迭代K次后求出一个hash值

$$F_t(i) = \left(F_{t-1}(i) \times \mathrm{A} + \sum_{i \to j} F_{t-1}(\mathrm{j}) \times \mathrm{B} + \sum_{j \to i} F_{t-1}(\mathrm{j}) \times C + D \times (i == a)\right) \ mod \ P$$

  - 如果两个图所有点的hash值相等，则图形（更大概率地）同构

- 图以二维数组的方式输入
- a[q][2]指的第q条边的两端顶点
- n：图的点数
- m: 图的边数
- co: 每个点的hash值

$$F_t(i) = \left(F_{t-1}(i) \times \mathrm{A} + \sum_{i \to j} F_{t-1}(\mathrm{j}) \times \mathrm{B} + \sum_{j \to i} F_{t-1}(\mathrm{j}) \times C + D \times (i == a)\right) \ mod \ P$$

```
void graph_hash(){
    int q,w,e;
    for (int i=0;i<n;i++){
        for (q=0;q<n;q++) f[q]=1;
        for (int z=0;z<K;z++){
            memcpy(tf,f,sizeof(f));
            for (q=0;q<n;q++) f[q]*=A;
            for (q=0;q<m;q++){
                f[a[q][0]]+=tf[a[q][1]]*B;
                f[a[q][1]]+=tf[a[q][0]]*C;
            }
            f[i]+=D;
            for (q=0;q<n;q++) f[q]%=P;
        }
        co[i]=f[i];
    }
    sort(co,co+n);
}
```

问题

- 问题1: 给任意两个图，请设计一个方法来计算二者是否相似？
- 问题2: 已知两个相似的图，请设计一个（可编程的）方法建立两个图结点之间的映射关系。
  - 太难了，还是改成两棵树