Graph Isomorphism Problem

with programming project focused on Tree Isomorphism Problem

Kamil Szymon Jadeszko

HS Mittweida
Network Algorithms Course

January 11th, 2016

Definition (Graph Isomorphism)

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be graphs.
A mapping $\phi : V_1 \to V_2$ is called graph isomorphism iff

$$\forall u, v \in V_1 \quad (u, v) \in E_1 \iff (\phi(u), \phi(v)) \in E_2.$$

Remark

If this mapping exists we say that G_1 and G_2 are isomorphic.

Examples

[on the board]

- GIP is NP problem.

- GIP is NP problem.
- There is still open question if GIP is NP-complete problem.

- GIP is NP problem.
- There is still open question if GIP is NP-complete problem.
- Brute force algorithm has complexity $O(n!)$

- GIP is NP problem.
- There is still open question if GIP is NP-complete problem.
- Brute force algorithm has complexity $O(n!)$
- Best known algorithm runs in $2^{O(\log^c(n))}$ time - very new boundary, work of László Babai, published last year. Previous $2^{O(\sqrt{n\log(n)})}$

- GIP is NP problem.
- There is still open question if GIP is NP-complete problem.
- Brute force algorithm has complexity $O(n!)$
- Best known algorithm runs in $2^{O(\log^c(n))}$ time - very new boundary, work of László Babai, published last year. Previous $2^{O(\sqrt{n\log(n)})}$
- GIP has practical applications - biology, cryptography,

It's not so easy to live with computer scientists

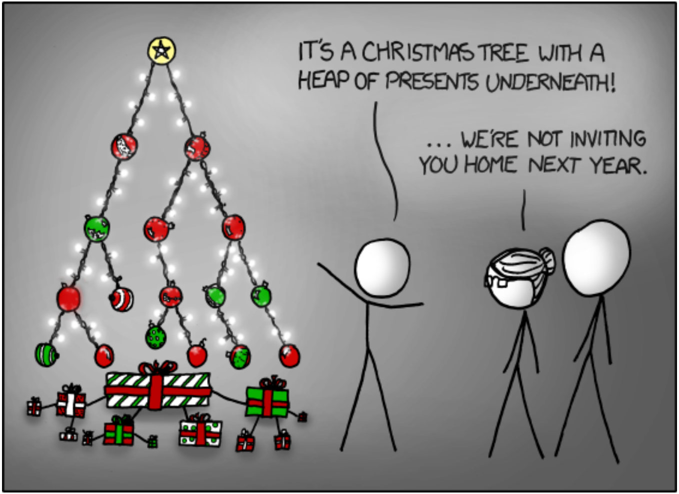

Figure 1: source: XKCD.com

Definition (Tree)

Tree is simple, connected, acyclic graph.

Definition (Rooted tree)

Rooted tree $T(V, E, r)$ is a tree with one selected vertex $r \in V$ called root.

Did I told you, that computer scientists are weird?

Figure 2: source: weather.gov

Basic terms:

1. Root
2. Leaf
3. Child

Python project

1. Rooted Ordered Trees
2. Rooted Trees
3. Ordinary Trees

Specific case - ordered (planted) trees

Definition (first() and next() operator)

[on the board]

Definition (Rooted Ordered Tree Isomorphism)

Let $T_1 = (V_1, E_1, r_1)$ and $T_2 = (V_2, E_2, r_2)$ be two ordered trees. Mapping $\phi : V_1 \to V_2$ is called ordered tree isomorphism iff:

1. $\phi(r_1) = r_2$
2. $\phi(first(v)) = first(\phi(v))$ for all $v \in V_1$ where v isn't a leaf
3. $\phi(next(v)) = next(\phi(v))$ for all $v \in V_1$ where v is nonlast child

Algorithm (ROTI)

1. Compare trees sizes - if are different trees aren't isomorphic.
2. Assign to every vertex of T_1 his index in pre-order traversal.
3. Assign to every vertex of T_2 his index in pre-order traversal.
4. Make bijection $\phi : V_1 \to V_2$ s.t

$$\phi(v_1) = v_2 \iff index(v_1) = index(v_2)$$

5. Check previous mentioned conditions of r.o.t. isomorphism for ϕ

Algorithm

Complexity

This algorithm runs in $O(n)$ time, where n is size of tree.

Rooted trees

Definition (Rooted Tree Isomorphism)

Let $T_1 = (V_1, E_1, r_1)$ and $T_2 = (V_2, E_2, r_2)$ be rooted trees
A mapping $\phi : V_1 \to V_2$ is called rooted tree isomorphism iff

$$\phi \text{ is graph isomorphism} \quad \text{and} \quad \phi(r_1) = r_2$$

Examples

[on the board]

Algorithm (Rooted tree labeling)

[explanation using interactive python shell]

Algorithm (RTI)

1. Compare trees sizes - if they are different trees aren't isomorphic.
2. Compare labels of trees - if they are different trees aren't isomorphic, otherwise - they are.

Algorithm

Complexity

This algorithm runs in $O(n^2 \cdot \log(n))$ time, where n is size of tree and can be easy optimized to $O(n^2)$

Remark

There exist rooted tree isomorphism algorithms which run in $O(n)$ and can be found in positions 1. and 2. from references.

Ordinary trees

1. Naive way.
2. Better way.

Ordinary trees

Lemma

Existence of $O(n)$-complexity algorithm for rooted tree implies existence of $O(n)$-complexity algorithm for ordinary trees.

Proof: [on the board]

Definition (Tree center)

A center of tree is a vertex v such that the longest path from v to a leaf is minimal over all vertices.

Thank you for your attention.

References

A. Aho, J. Hopcrot, J. Ullman
The Design and Analysis of Computer Algorithms

G. Valiente
Algorithms on Trees and Graphs

M. Bonamy
A Small Report on Graph and Tree Isomorphism

A. Smal
Tree Isomorphism Talk