An O(n^2) Algorithm for Graph Isomorphism: A Novel Approach: S*

By Sherry Sarkar

Consider the following algorithm:

We have a Graph, G = (V, E) and we wish to search this graph for a certain vertex, V. The traditional approach is, indeed, Dijktra's. However, I propose this:

What if we introduce a ***heuristic***?

What if our algorithm prioritized paths that **appear** to lead to a solution quicker?

Therefore, I propose an extension of Dijktra's algorithm, where we both consider weights and AND this heuristic function. I call this novel, completely original algorithm

S*

S for Sherry.

Theorem: Two graphs G = (V, E) and G' = (V', E') are isomorphic if they share two common vertices, S and E, and when you run S*, you get the same path.

In graph theory, an **isomorphism of graphs** *G* and *H* is a bijection between the vertex sets of *G* and *H*

$$f\colon V(G) \to V(H)$$

such that any two vertices *u* and *v* of *G* are adjacent in *G* if and only if $f(u)$ and $f(v)$ are adjacent in *H*. T

An isomorphism is a structure preserving bijection. Consider the following map G -> G': given the S* path from S to E, take a vertex V and map it to the vertex in the path that sorta looks the same, and then you map that vertex to the V'.

The run time of ~~A*~~ S* is O(n^2) where n is the number of edges. So we are done.

The Far Superior O(1) approach to GI

By Daniel Hathcock

Idea: **Reductions**

We want to find a reduction from graph isomorphism to a different problem which we know how to solve quickly.

Given a graph G, consider the group of isomorphisms from G to itself (the automorphism group).

Theorem: A graph is identified by its automorphism group.

Proof: trivial

So, if we can find the automorphism groups for graphs G and G', we can determine if they are isomorphic by testing whether their automorphism groups are isomorphic!

Good news: We can quickly and easily find the automorphism group for a graph!

How? Ideas?

Just look for it! (e.g. search using ~~A*~~S*)

Great! Now we can solve graph isomorphism quickly.

Can we solve the group isomorphism problem quickly? Well of course! Just reduce it to GI, and use our new fast solution for GI to solve!

Brilliant! But no better than Sherry's O(N^2) (yet)...
But it gives us an idea!

Instead of reducing GI to group isomorphism, let's reduce GI to GI!

Now, we can solve GI using the fast algorithm we just came up with for GI, and the reduction is in constant time! $O(1)$!

Outstanding!

Can we do better?...

Of course!

As n increases, it becomes common knowledge whether two graphs are isomorphic.

We can get a $o(1)$ (little o) algorithm by just not running an algorithm, but instead just looking at the graph and figuring it out