CS101

ISOMORPHISM ALGORITHM

Utkarsh Patel-2022CSB1140

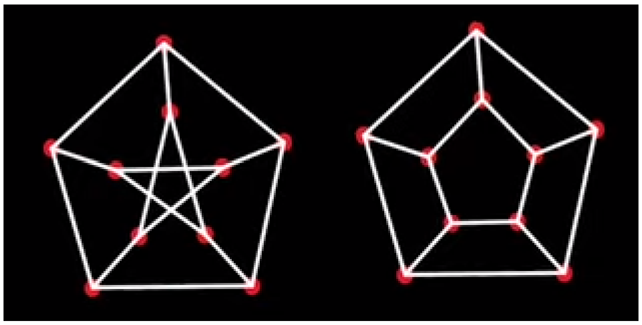
What is isomorphism?

A graph G1(v1,e1) and another graph G2(v2,e2) are said to be isomorphic if and only if there exists a bijection function f:v1->v2 such that they preserve the edges / adjacencies.

Algorithm 1-First algorithm involves labeling the first graph in an arbitrary order and then trying to find the corresponding permutations of the vertices in the other graph . Obviously this is a very inefficient method to find isomorphism in n! Steps .

We can implement it by making adjacency matrix for the two given graphs A and B. Now our goal is to find a permutation matrix P(B) such that multiplication of P(B),B,P(B) transpose will give the adjacency matrix A.

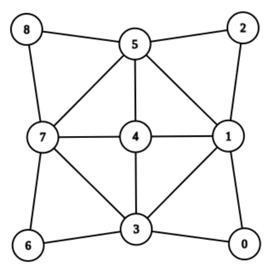
Algorithm 2- Sometimes proving two graphs are non-isomorphic can be very easy if we manage to find a structure that is non-existent in the other graph.This exercise can be done in 2^n steps.For example , the rectangle shape in graph 2 can never be found in graph 1.



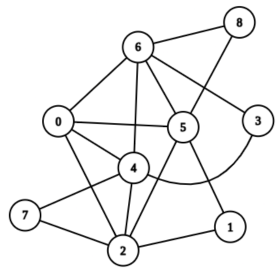
Graph 1 **Graph 2**

It's implementation can be done by picking a structure from a graph A then traversing the graph B to find the same structure ,if found , repeat the above process until all the possible structures are exhausted or an anomaly is found.

Algorithm 3- This algorithm is a part of a famous NAUTY algorithm .The real algorithm is a little complex ,but it's subpart aims to find standard labeling after giving arbitrary numbers to each vertex .The process has major steps as follows-
1-giving arbitrary numbers to each vertex
2-arranging and dividing numbers in order of their degrees
3-if symmetry is found,break in all possible parts
4-refine the division until all vertices are divided
5-take the largest sequence of both the graphs,it will be a bijection,check the isomorphism



0268|4|1357
>0|268|4|1357-0|268|4|13|57-0|26|8|4|13|57->0|2|6|8|4|13|57-0|2|6|8|4|1|3|5|7
->0|6|2|8|4|13|57-0|6|2|8|4|3|1|7|5
>2|068|4|1357-2|068|4|15|37-2|08|6|4|15|37->2|0|8|6|4|15|37-2|0|8|6|4|1|5|3|7
->2|8|0|6|4|15|37-2|8|0|6|4|5|1|7|3
>6|028|4|1357-6|028|4|37|15-6|08|2|4|37|15->6|0|8|2|4|37|15-6|0|8|2|4|3|7|1|5
->6|8|0|2|4|37|15-6|8|0|2|4|7|3|5|1
>8|026|4|1357-8|026|4|57|13-8|26|0|4|57|13->8|2|6|0|4|57|13-8|2|6|0|4|5|7|1|3
->8|6|2|0|4|57|13-8|6|2|0|4|7|5|3|1



1378|0|2456

>1|378|0|2456-1|378|0|25|46-1|78|3|0|25|46->1|7|8|3|0|25|46-1|7|8|3|0|2|5|4|6

->1|8|7|3|0|25|46-1|8|7|3|0|5|2|6|4

>3|178|0|2456-3|178|0|46|25-3|78|1|0|46|25->3|7|8|1|0|46|25-3|7|8|1|0|4|6|2|5

->3|8|7|1|0|46|25-3|8|7|1|0|4|6|5|2

>7|138|0|2456-7|138|0|24|56-7|13|8|0|24|56->7|1|3|8|0|24|56-7|1|3|8|0|2|4|5|6

->7|3|1|8|0|24|56-7|3|1|8|0|4|2|6|5

>8|137|0|2456-8|137|0|56|24-8|13|7|0|56|24->8|1|3|7|0|56|24-8|1|3|7|0|5|6|2|4

->8|3|1|7|0|56|24-8|3|1|7|0|6|5|4|2

At last we need to take only the lexicographically largest order of vertices that would define the bijection from A to B (if exists) , then we need to recheck if the bijection provided actually defines isomorphism of the graphs.

Here we can say that=8<->8,6<-> 3,2<-> 1,0<-> 7,4<-> 0,7<-> 6,5<-> 5,3<-> 4,1<->2

CONCLUSION

Hence we can conclude that the problem of checking the isomorphism is still not solved completely,even the most efficient algorithms fail to solve such problems where the number of vertices exceeds the number of 100.