**K L HYDERABAD**

**FRESHMAN ENGINEERING DEPARTMENT**

A Project-Based Lab Report

On

Isomorphism in biological networks

**SUBMITTED BY:**

I.D NUMBER NAME

* + - * 2010030046 E.Pravallika
      * 2010030168 Tahseen Begum
      * 2010030344 N. Sowgna
      * 2010030445 Keerthana Pulugam

**UNDER THE ESTEEMED GUIDANCE OF**

**Dr. P. Sree Lakshmi**

**<DESIGNATION>** 

**KONERU LAKSHMAIAH EDUCATION FOUNDATION**

(Deemed to be University)

Moinabad Road, Aziz Nagar, Hyderabad - 500075

**DEPARTMENT OF BASIC ENGINEERING SCIENCES**

****

**CERTIFICATE**

This is to certify that the project-based laboratory report entitled “Isomorphism in biological networks” submitted by Mr./Ms**. Tahseen Begum,** **E.Pravallika,** **N.Sowgna,** **Keerthana Pulugam** bearing Regd. No. 2010030168, 2010030046, 2010030344, 2010030445 to the **Department of Basic Engineering Sciences, KL University** in partial fulfillment of the requirements for the completion of a project in the “DAA Project”course in II B Tech IV Semester, is a Bonafede record of the work carried out by him/her under my supervision during the academic year 2021-22.

## **Signature of the Supervisor**

Name and Designation

## **Signature of the HOD Signature of the External Examiner**

**ACKNOWLEDGEMENTS**

It is great pleasure for me to express my gratitude to our honorable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project-based laboratory report.

I express my sincere gratitude to our Principal **Dr. L. Koteswara Rao** for his administration of our academic growth.

I express sincere gratitude to our Coordinator for her leadership and constant motivation provided in the successful completion of our academic semester. I record it as my privilege to deeply thank you for providing us with the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project **supervisor Dr. P. Sree Lakshmi** for her novel association of ideas, encouragement, appreciation, and intellectual zeal which motivated us to venture into this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to making this project report success.

Name: Tahseen Begum

Regd. No: 2010030168

Name: E.Pravallika

Regd. No: 2010030046

Name: N.Sowgna

Regd. No: 2010030334

Name: Keerthana Pulugam

Regd. No: 2010030445

**ABSTRACT**

* Number of real world problems is represented by graph.
* Graph isomorphism is the area of pattern matching and widely used in various applications such as image processing, protein structure, computer and information system, chemical bond structure, Social Networks.
* This project surveys both various applications of graph isomorphism and their importance in the society.

**INDEX**

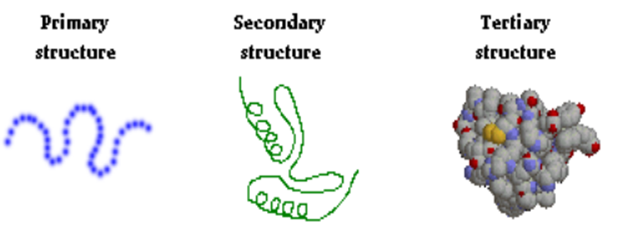
|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE NO** |
| 1 | Problem Statement | 05 |
| 2 | Solution strategy- Algorithm used | 06-07 |
| 3 | Numeric example of the algorithm (Sample input - expected output) | 08 |
| 4 | Flow chart of the proposed solution | 09 |
| 5 | Hardware and Software used | 09 |
| 6 | Implementation | 10-14 |
| 7 | Results | 15-17 |
| 8 | Future scope of improvement | 18 |
|  |  |  |

9 References 19

**Problem Statement**

We Use Protein Structure ,Nodes Represents Protein And Edges Represents Their Interactions Between Nodes.

* We Have Three Levels Of Protein Structure I.E. , Primary Structure ,Secondary Structure , Tertiary Structure.

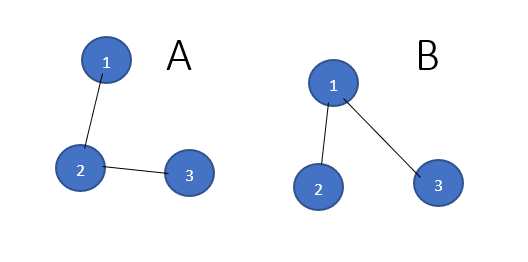
****

* For Example Collection Of Food Is Available And Every Food Is Having Their Own Structure That Is Graph Structure. For Providing The Food And That Food Contains Some Proteins, Then Find Protein Graph Structure First And Check Where It Is Available In The Food Structure Or Not.

**Solution strategy- Algorithm used**

We have Graph A and B , now we have to check the two graph are isomorphic are not using VF2 Algorithm

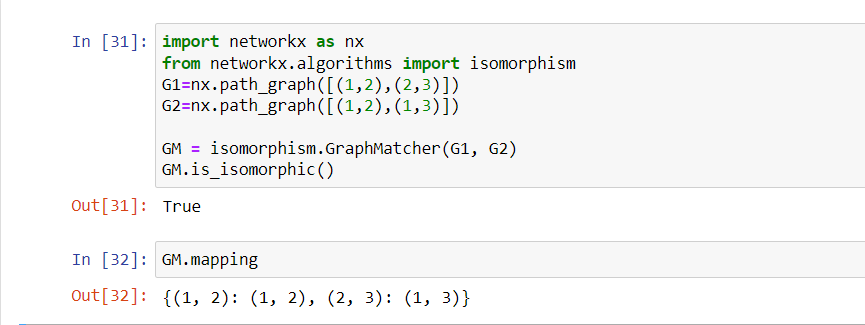
Our vertex are 1,2,3 in graph A and B



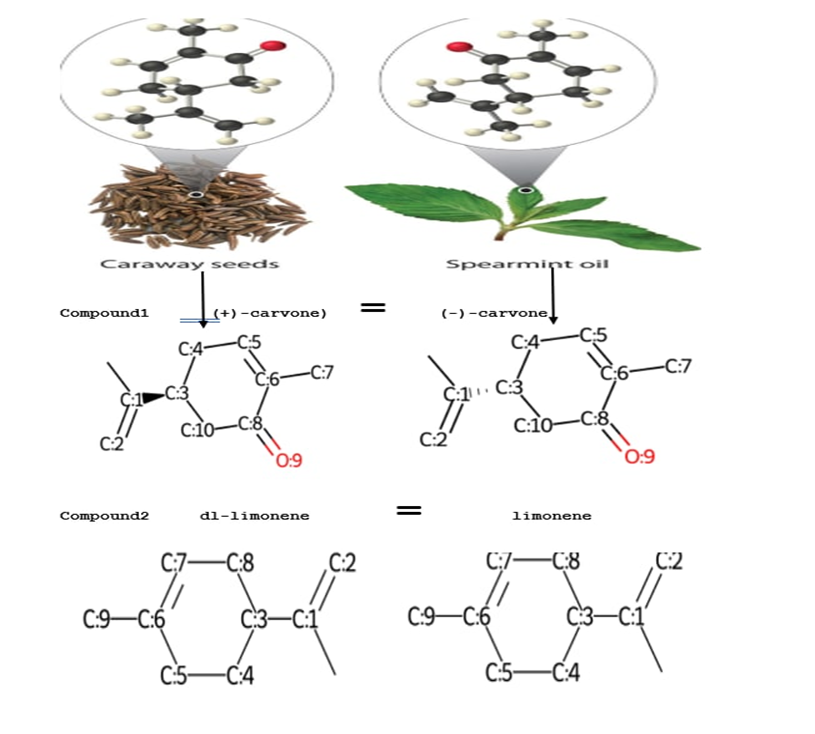
* Step 1 :
* I match empty A with empty B it always works.
* We can match 1A with 1B,2B,3B
* Now we take 1A with 1B it always works
* Step 2 :
* I can match 2A with 2B or 3B
* I match 2A with 2B always works because 1A,2A and 1B,2B are connected.
* Step 3 :
* I can match 3A with any node in Graph B we cannot connect because there is no edge between 2B and 3B in graph B , so we will go back to again step2.
* Step 4:
* I can match with 2A with 3B we cannot connect because there is no edge between 2B and 3B in graph B , so we will go back to again step2,But in step2 we didn’t have solution , so we will go to step1.
* Step 5:
* I match 1A with 2B and 2A with 1B and 3A with 3B
* The graphs are isomorphic

**Numeric example of the algorithm (Sample input - expected output)**

An implementation of VF2 algorithm for graph isomorphism testing.The simplest interface to is to call networkx.is\_isomorphic().



**Flow chart**

****

**Hardware and Software used**

1)Windows 11 for x64-based system

2) brackets

3) jupyter notebook

**IMPLEMENTATION**

def mol\_with\_atom\_index(mol):

for atom in mol.GetAtoms():

atom.SetAtomMapNum(atom.GetIdx())

return mol

from urllib.request import urlopen

from urllib.parse import quote

def CIRconvert(ids):

try:

url = 'https://cactus.nci.nih.gov/chemical/structure/' + quote(ids) + '/smiles'

ans = urlopen(url).read().decode('utf8')

return ans

except:

return 'Did not work'

from rdkit import Chem

p\_\_c1\_name1 = input("Enter primary protien compoud1 of caraway seeds : ")

s\_c1\_formula1 = CIRconvert(p\_\_c1\_name1)

print(s\_c1\_formula1)

mol1 = Chem.MolFromSmiles(s\_c1\_formula1)

mol\_with\_atom\_index(mol1)

from rdkit import Chem

p\_\_c1\_name2 = input("Enter primary protien compoud1 of spearmint oil : ")

s\_c1\_formula2 = CIRconvert(p\_\_c1\_name2)

print(s\_c1\_formula2)

mol2 = Chem.MolFromSmiles(s\_c1\_formula2)

mol\_with\_atom\_index(mol2)

from rdkit.Chem.EnumerateStereoisomers import EnumerateStereoisomers,StereoEnumerationOptions

isomers = tuple(EnumerateStereoisomers(mol1))

len(isomers)

from rdkit import Chem

for smi in sorted(Chem.MolToSmiles(x,isomericSmiles=True)for x in isomers):

print(smi)

isomer=Chem.MolFromSmiles(smi)

mol\_with\_atom\_index(isomer)

from pysmiles import read\_smiles

import networkx as nx

smiles='CC(=C)[C@H]1CC=C(C)C(=O)C1'

mol\_1=read\_smiles(s\_c1\_formula1)

mol\_2=read\_smiles(s\_c1\_formula2)

nodes\_c1 =mol\_1.nodes(data='element')

nodes\_c2=mol\_2.nodes(data='element')

adj\_matrix\_c1=nx.to\_numpy\_matrix(mol\_1)

adj\_matrix\_c2=nx.to\_numpy\_matrix(mol\_2)

print(nodes\_c1)

print(adj\_matrix\_c1)

print(nodes\_c2)

print(adj\_matrix\_c2)

from networkx.algorithms import isomorphism

G1=nx.from\_numpy\_matrix(adj\_matrix\_c1)

G2=nx.from\_numpy\_matrix(adj\_matrix\_c2)

GM=isomorphism.GraphMatcher(G1,G2)

GM.is\_isomorphic()

from networkx.algorithms import isomorphism

G1=nx.from\_numpy\_matrix(adj\_matrix\_c1)

G2=nx.from\_numpy\_matrix(adj\_matrix\_c2)

GM=isomorphism.GraphMatcher(G1,G2)

GM.is\_isomorphic()

from rdkit import Chem

p\_\_c2\_name2 = input("Enter primary protien compoud2 of spearmint oil : ")

s\_c2\_formula2 = CIRconvert(p\_\_c2\_name2)

print(s\_c2\_formula2)

mol2 = Chem.MolFromSmiles(s\_c2\_formula2)

mol\_with\_atom\_index(mol2)

from rdkit.Chem.EnumerateStereoisomers import EnumerateStereoisomers,StereoEnumerationOptions

isomers = tuple(EnumerateStereoisomers(mol1))

len(isomers)

from rdkit import Chem

for smi in sorted(Chem.MolToSmiles(x,isomericSmiles=True)for x in isomers):

print(smi)

isomer=Chem.MolFromSmiles(smi)

mol\_with\_atom\_index(isomer)

from pysmiles import read\_smiles

import networkx as nx

smiles='CC(=C)C1CCC(=CC1)C'

mol\_1=read\_smiles(s\_c2\_formula1)

mol\_2=read\_smiles(s\_c2\_formula2)

nodes\_c1 =mol\_1.nodes(data='element')

nodes\_c2=mol\_2.nodes(data='element')

adj\_matrix\_c1=nx.to\_numpy\_matrix(mol\_1)

adj\_matrix\_c2=nx.to\_numpy\_matrix(mol\_2)

print(nodes\_c1)

print(adj\_matrix\_c1)

print(nodes\_c2)

print(adj\_matrix\_c2)

from networkx.algorithms import isomorphism

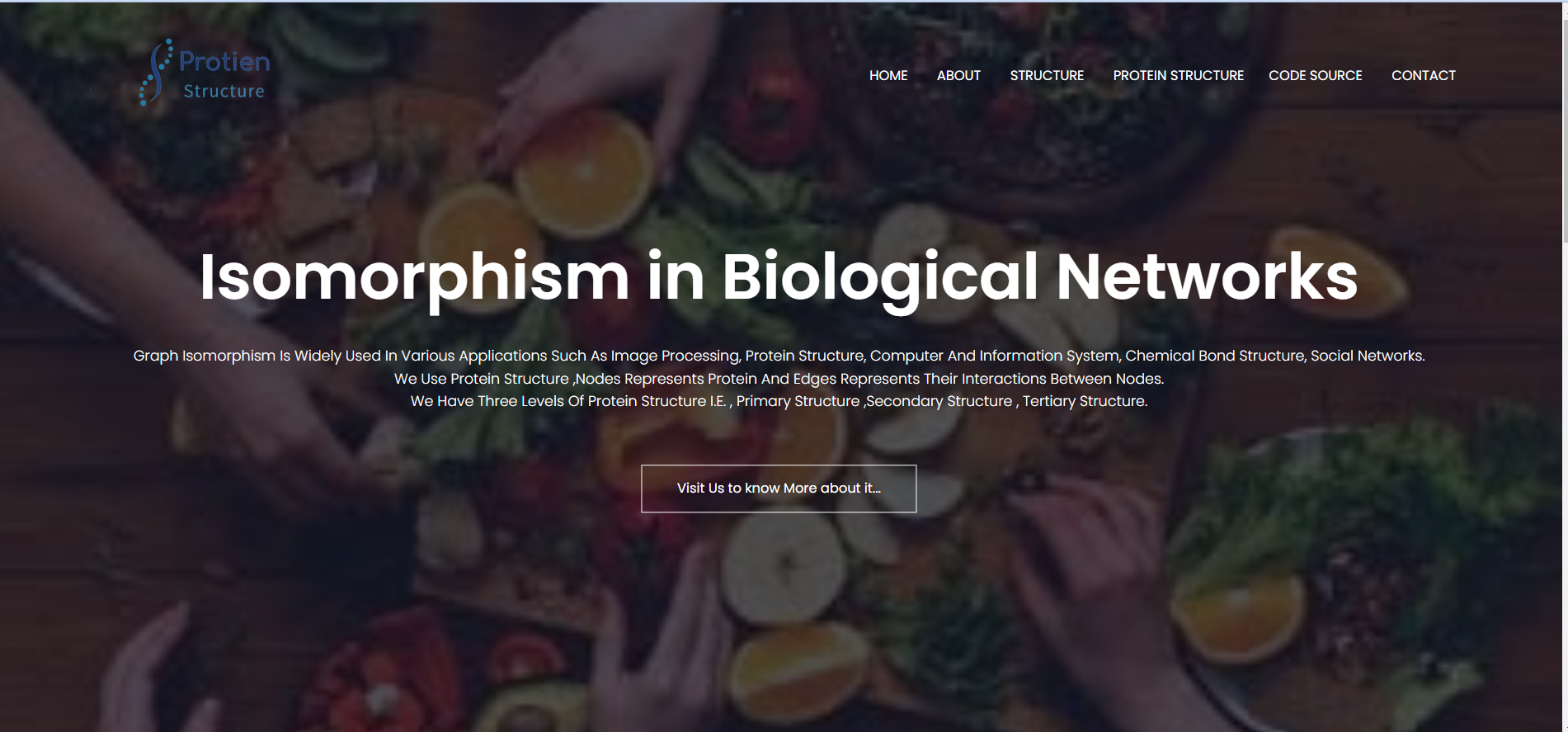
G1=nx.from\_numpy\_matrix(adj\_matrix\_c1)

G2=nx.from\_numpy\_matrix(adj\_matrix\_c2)

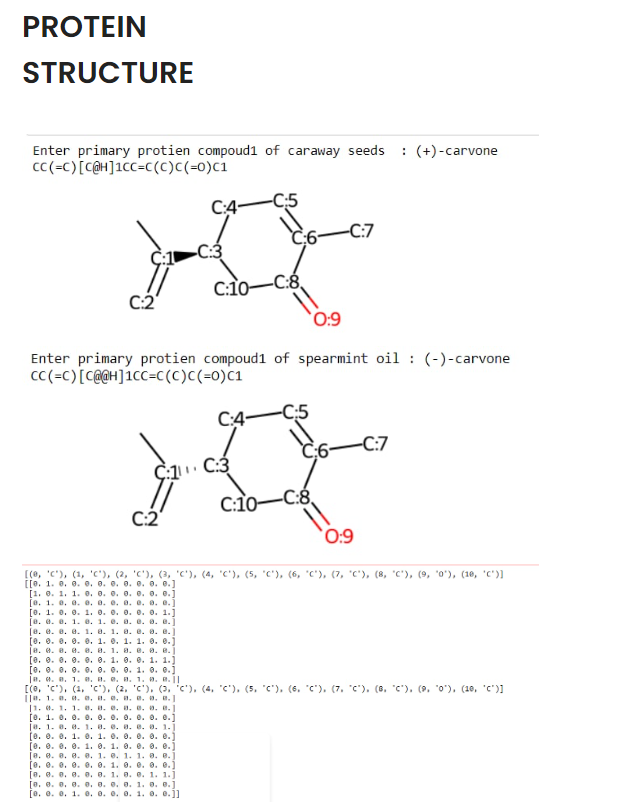
GM=isomorphism.GraphMatcher(G1,G2)

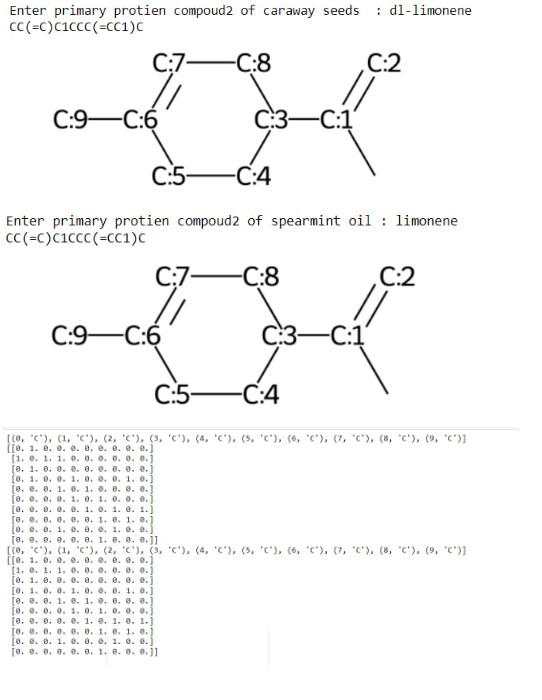
GM.is\_isomorphic()

**RESULTS**



****

****

****

**Future scope of improvement**

This project can be further enhanced to provide a 3D shape generator so that we can show the diagram in a 3D shape. Now it shows a normal diagram in the future we make that diagram into a 3D shape by doing the changes.

**REFERENCES**

https://github.com/K-L-U-H/Isomorphism-in-biological-networks

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3152790/>

https://cactus.nci.nih.gov/chemical/structure/