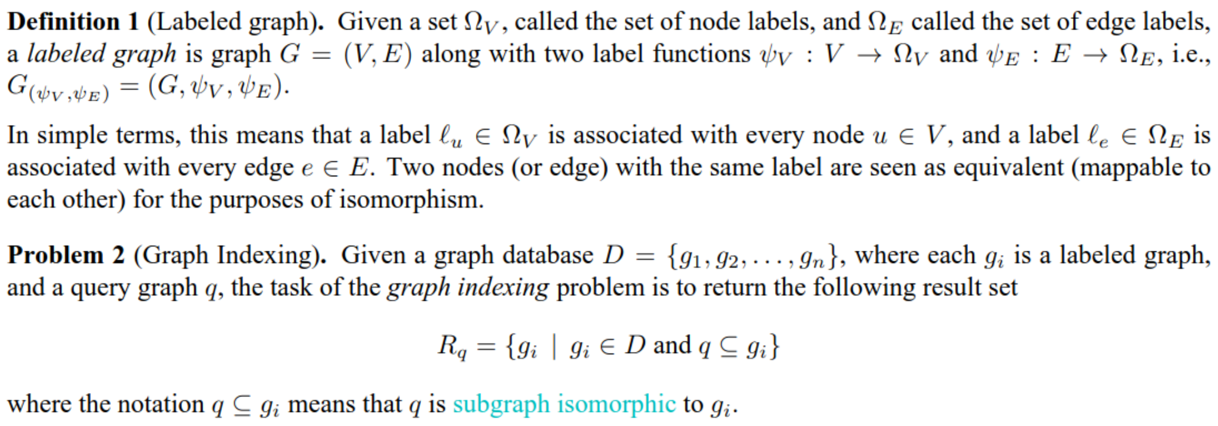
COL7361 Data MIning

Q3 Report - Graph Indexing

Team Eleventh Hour



Definition 1 (Labeled graph). Given a set Ω_V, called the set of node labels, and Ω_E called the set of edge labels, a *labeled graph* is graph $G = (V, E)$ along with two label functions $\psi_V : V \rightarrow \Omega_V$ and $\psi_E : E \rightarrow \Omega_E$, i.e., $G_{(\psi_V, \psi_E)} = (G, \psi_V, \psi_E)$.

In simple terms, this means that a label $\ell_u \in \Omega_V$ is associated with every node $u \in V$, and a label $\ell_e \in \Omega_E$ is associated with every edge $e \in E$. Two nodes (or edge) with the same label are seen as equivalent (mappable to each other) for the purposes of isomorphism.

Problem 2 (Graph Indexing). Given a graph database $D = \{g_1, g_2, \ldots, g_n\}$, where each g_i is a labeled graph, and a query graph q, the task of the *graph indexing* problem is to return the following result set

$$R_q = \{g_i \mid g_i \in D \text{ and } q \subseteq g_i\}$$

where the notation $q \subseteq g_i$ means that q is subgraph isomorphic to g_i.

Introduction

The task in this assignment is to provide a solution to the above-mentioned graph indexing problem. As suggested by Gemini, “Graph indexing enables high-performance, real-time traversal and query execution in highly connected datasets. Key use cases include identifying complex fraud patterns, delivering personalized real-time recommendations, mapping social network relationships, optimizing logistical supply chains, and managing intricate IT network infrastructures.” The following report talks about the dataset we worked on, and the methodology employed.

Dataset Description

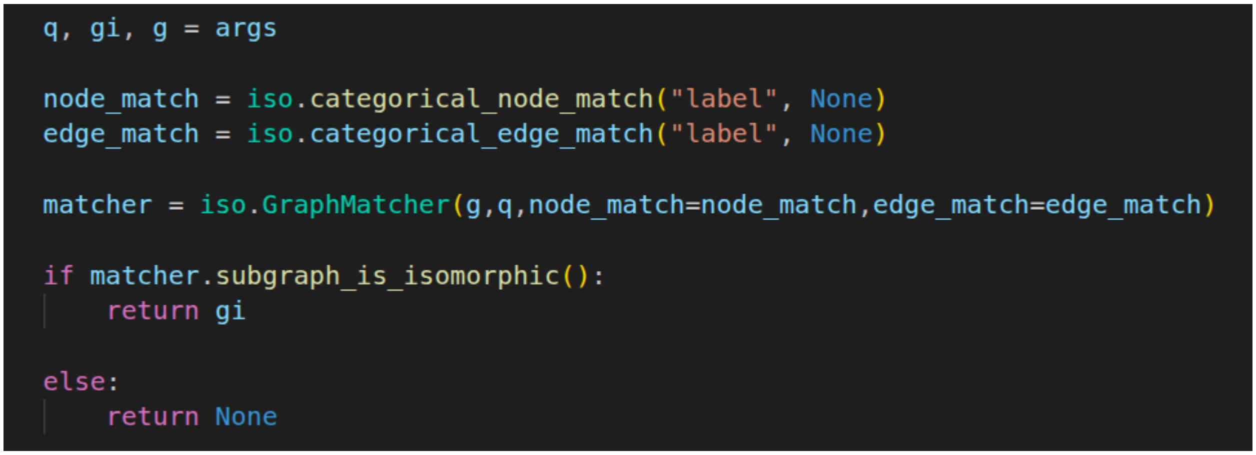
Graph indexing task was to be done on two molecular datasets namely **Mutagenicity** and **NCI-H23**. According to an article published in the *Journal of Medicinal Chemistry*, “Mutagenicity is one of the numerous adverse properties of a compound that hampers its potential to become a marketable drug”. This dataset was created for the purpose of increasing the reliability and accuracy of predicting mutagenicity of molecules (which form the toxic substructure of a compound). The dataset has 4337 molecular structures represented as graphs with corresponding labels. Hence algorithmic substructure mining can help in identifying these mutagenic structures.

The second dataset is generated with the aim of evaluating descriptors that identify connected components in chemical compounds to classify them.

Methodology

Our solution to the graph-indexing problem follows the brute force approach of performing subgraph isomorphism tests between every query and candidate graph pairs.

For this process, we use the NetworkX library in python, which has functions for performing subgraph isomorphism checks.



```
q, gi, g = args

node_match = iso.categorical_node_match("label", None)
edge_match = iso.categorical_edge_match("label", None)

matcher = iso.GraphMatcher(g,q,node_match=node_match,edge_match=edge_match)

if matcher.subgraph_is_isomorphic():
    return gi

else:
    return None
```

This code snippet executes the subgraph isomorphism check. To speed up the process, our program was modified to parallelize the subgraph isomorphism checks by utilizing the multi-core CPU architecture so that each core runs a process of SI test. This was possible with the help of multiprocessing module, native to python.

NetworkX uses the VF2 algorithm to perform the SI tests, which incrementally builds mapping between the nodes of the two graphs in a Depth-First manner.

The parallelization was helpful in speeding up the task. For instance, the program without parallelization ran for 7 mins 30 seconds on Mutagenicity dataset whereas it was reduced to 1 min 40 seconds after applying parallelization.

References

As mentioned in the file “direct_si.py”, Chatgpt was used to modify the program (that was initially executing SI tests sequentially) to parallelize the SI tests. Documentation of NetworkX was referred for using the in-built SI test command.