Molecule Database

By Group 9: Timmy, Danny, Andreas, and Edwin

Background

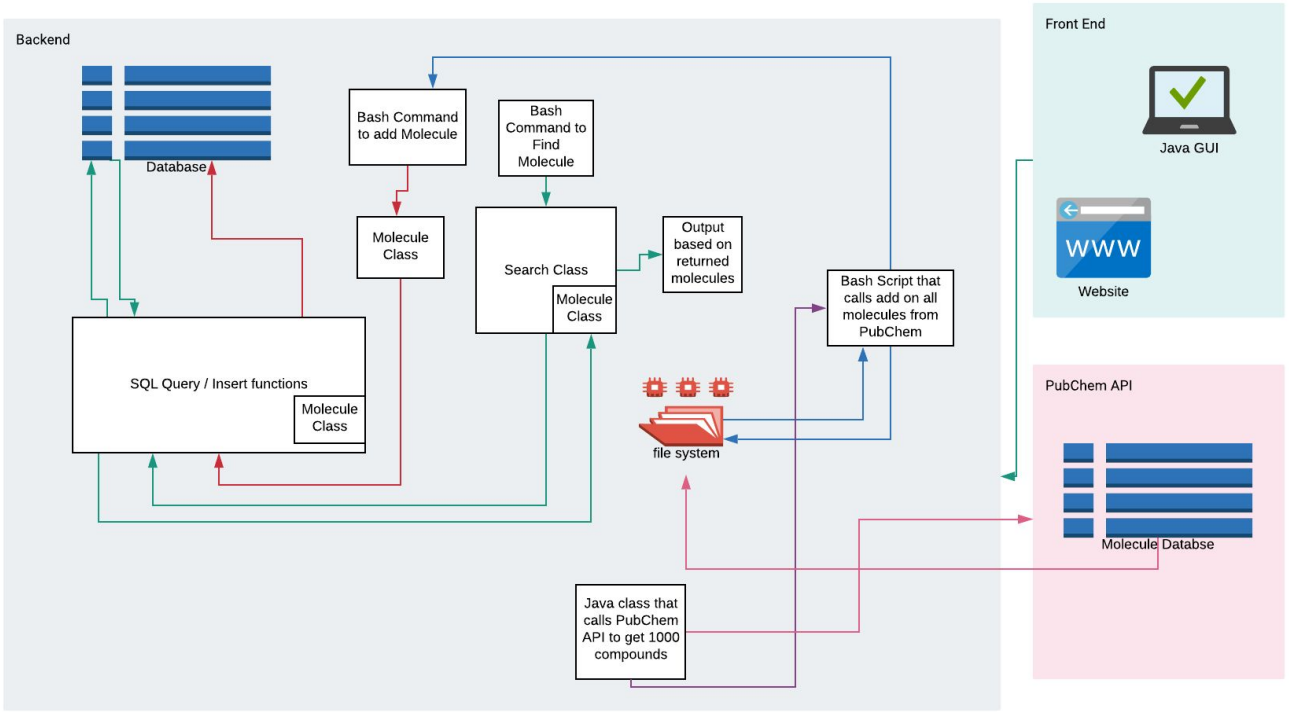
- Goal of the Project: Implement a molecular database that supports adding and searching for isomorphic molecules.
- What Project Consists:
  - CLI that supports adding and searching molecules
  - Web page and corresponding operations class for accessing the database.
  - Java GUI that provides molecular entry and search capabilities. it also displays molecules in graphical format and database statistics.
  - Ability to search for the most similar molecule to a given molecule
  - Ability to download 1,000 known compounds from an existing database: Pubchem

Comparisons . . .

We will be comparing with the other projects in terms of

1) Algorithm
2) Database
3) Features

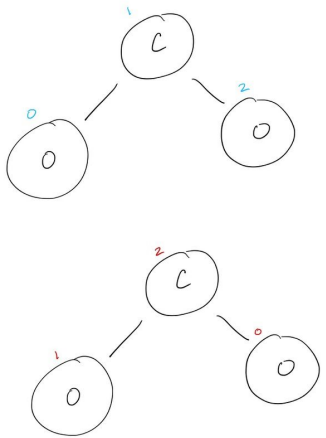
Project Architecture



1) Algorithms other groups

- Group 2:
  - Find Isomorphism: used pruning through hash maps, implement recursive backtracking and capable of early stopping.
  - Add Molecule: uses a hash map implementation to hash molecules on to the dataset.
- Group 6:
  - Find Isomorphism: recursive backtracking.
  - Add molecule: parse text file into DB and check for duplicates before adding.

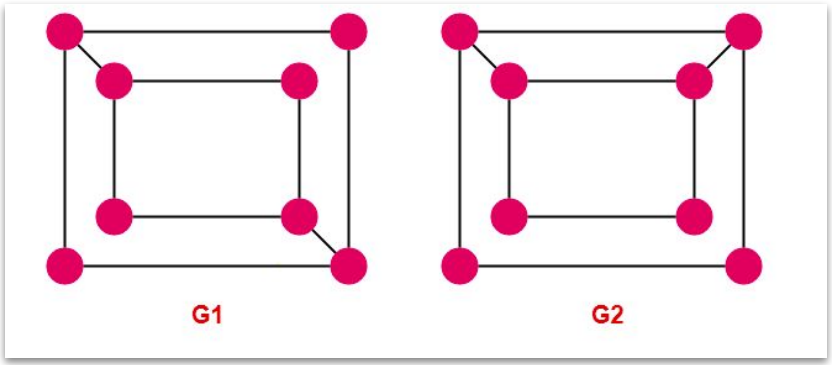
Graph Isomorphism

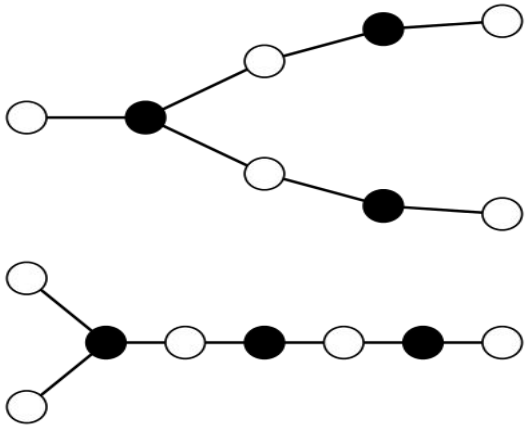


- Ideas:
  - Problem set up is not like the general problem
  - Heuristics advantage
  - Disjoint sets
  - Filtering (Weak and Strong Isomorphism)
  - Multinomial Runtime
- Algorithm
  - Most Important Ideas:
    - A function that creates a correspondence between two molecules
    - A way to get the ambiguous atoms in a molecule
    - A way to partition the set of ambiguous molecules based on their atoms
    - A way to create the permutations

Graph Isomorphism cont'd

- Algorithm
  - Implementation
    - Test out the weak isomorphism by checking if for each vertex in molecule 1 and its edges (in molecular terms a atom and the other atoms it connects) there exists another vertex in molecule 2 that has the same structure in molecule 1 while preserving uniqueness.
    - Where does weak Isomorphism fail:





Graph Isomorphism cont'd

- Algorithm
  - Strong Isomorphism:
    i. Get all the ambiguous atoms.
       - Hashset stores a pair that is comprised of a string and an ArrayList of strings.
       - Get the ambiguous atoms by seeing if there already exists a pair that is exactly the same as in the Hash Set.
    ii. Make an Initial Correspondence of the molecules by using the same function as weak Isomorphism.
    iii. Create a stack that stores partial solutions.
    iv. Permute set of disjoint atoms.
    v. Perform DFS on the solutions.
    vi. Return isomorphism if any of the solutions eliminates all ambiguity in the molecule.

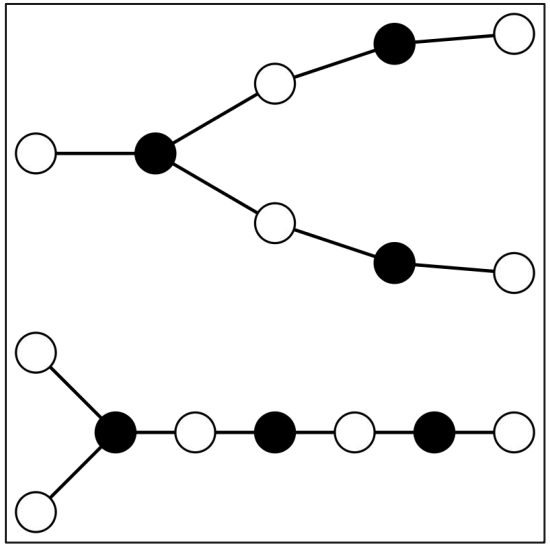
2) Database

1. **Group 9**:
    a. H2 Database Engine
2. Group 2:
    a. CSV
3. Group 6:
    a. SQLite

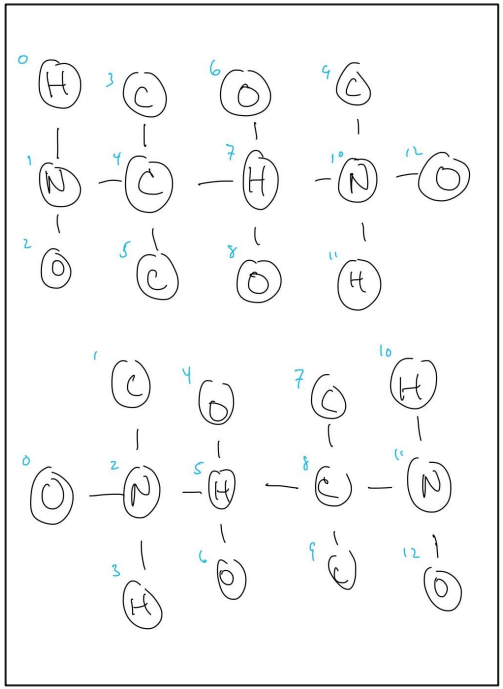
3) Features that we have in common -

- Group 6:
  - Find Most Similar
    - Selects a molecule that has the same number of atoms
- Group 9:
  - Find Most Similar
    - Calculates the distance to the molecule by looking at how many atoms should be changed to match the labels in the other molecule and how many edges should be added to get the same weakly isomorphic graph. The distance is given by:
      - Distance = 1*(#atoms changed) + 4*(#of edges changed)

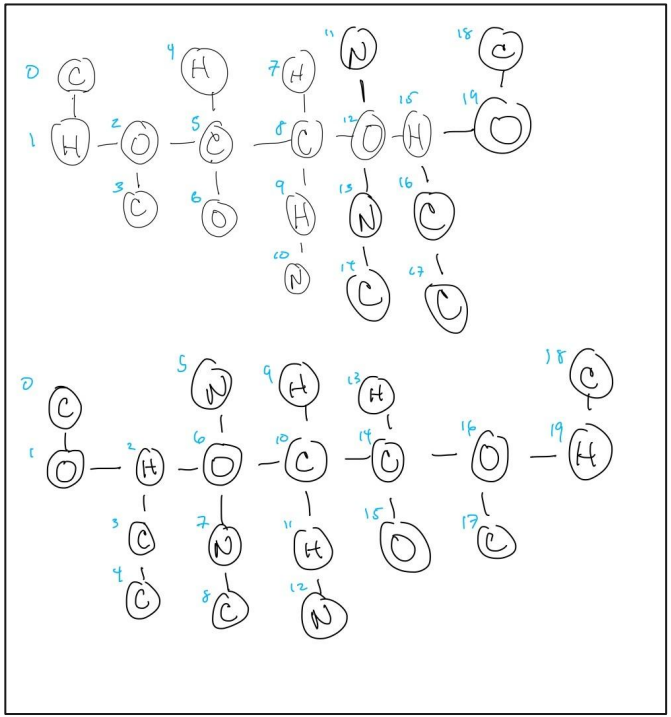
Benchmarks



8 atoms
Not Isomorphic

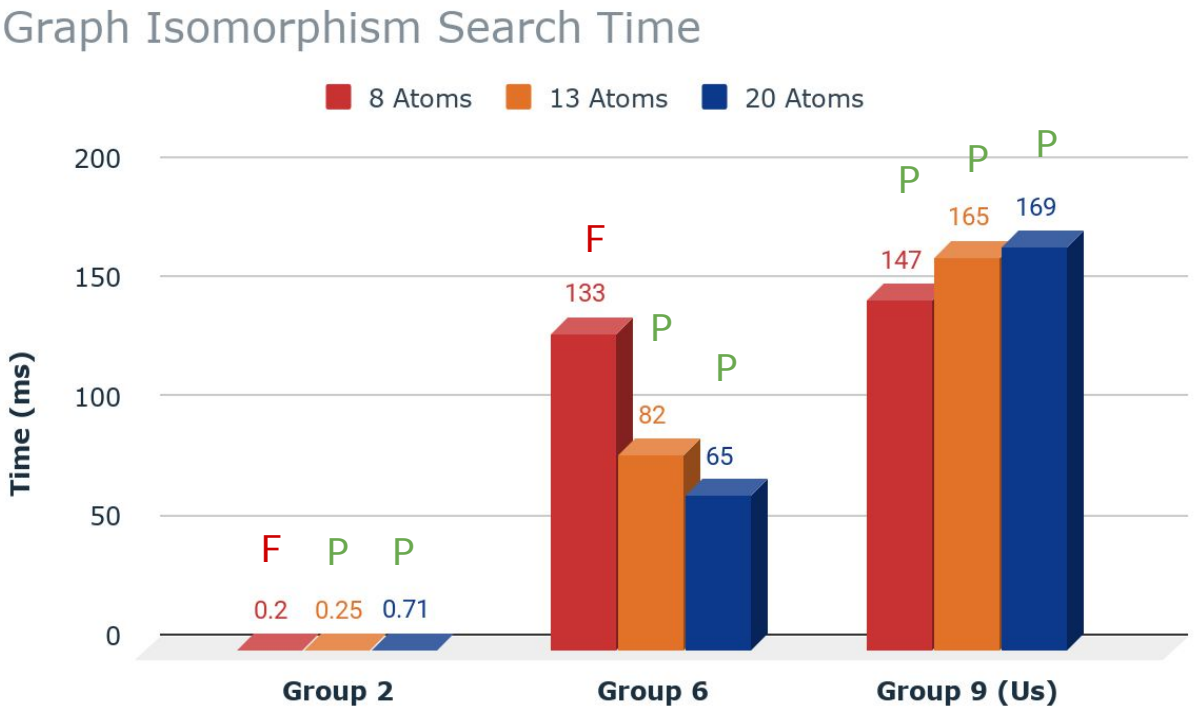


13 atom
Isomorphic



20 atoms
Isomorphic

Benchmarks



P - PASSED
F - FAILED

Distinction

Weaknesses

- No implementation of find Subgraph.(Group 6 and Group 2)
- Storing duplicates inside the database.(Group 6 doesn’t have this)
- No stopping condition for large molecules

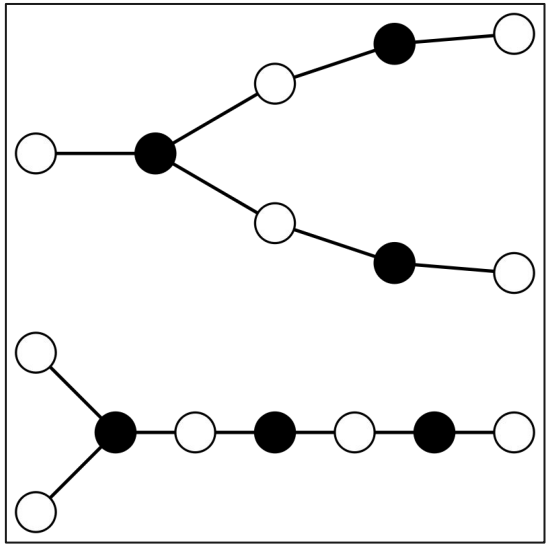
Strengths

- Intuitive Java GUI and Web GUI.
- Ensures graphs are generally isomorphic and not only weakly isomorphic.
- More robust similarity distance metric.

Live Demonstration

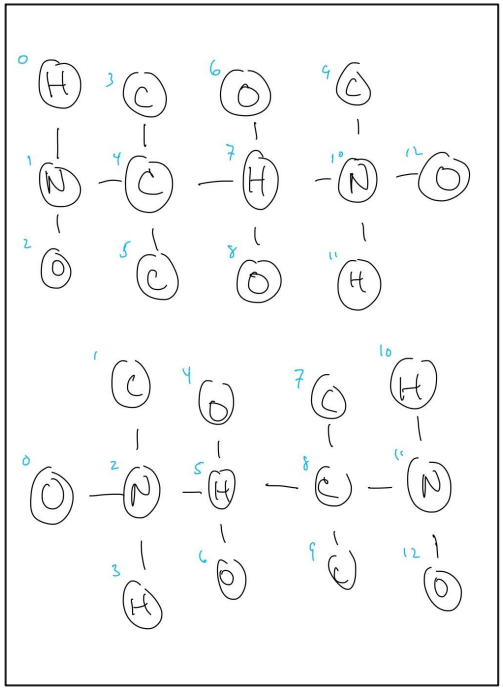
Questions?

Appendix

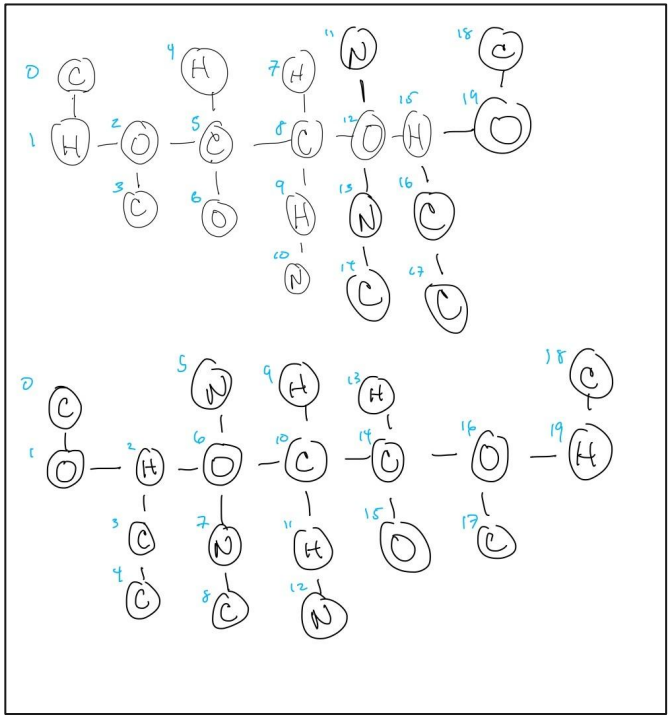


8 nodes
Not Isomorphic

https://en.wikipedia.org/wiki/Degree_(graph_theory)



13 nodes
Isomorphic



20 nodes
Isomorphic

References

1) Datta, S., Limaye, N., & Nimbhorkar, P. (2008). 3-Connected Planar Graph Isomorphism Is in Log-Space. Leibniz International Proceedings in Informatics, LIPIcs, 2, 155–162. https://doi.org/10.1109/ccc.2009.16

2) Pöial, Jaanus. (2003). Implementation of directed multigraphs in Java.. 163. 10.1145/957289.957337.

3) Abulaish, Muhammad, and Zubair Ali Ansari. "SubISO: A Scalable and Novel Approach for Subgraph Isomorphism Search in Large Graph." 11th IEEE International Conference on Communication Systems & Networks (COMSNETS), 2019.

4) Cordella, L. P., Foggia, P., Sansone, C., & Vento, M. (2004). A (sub)graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(10), 1367–1372. https://doi.org/10.1109/TPAMI.2004.75

5) Carletti, V., Foggia, P., Saggese, A., & Vento, M. (2018). Challenging the Time Complexity of Exact Subgraph Isomorphism for Huge and Dense Graphs with VF3. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 804–818. https://doi.org/10.1109/TPAMI.2017.2696940