**Dataset**

Source：https://www.kaggle.com/datasets/bfbarry/dictionary-graph

It contains a graph representation of the English dictionary, where each word is a node, and its edges are defined when a word appears in the definition of another word. The intriguing aspect of this dataset lies in its construction of a rich network of relationships through mutual references among words, reflecting the semantic connections between vocabulary items. In this graph, the edges between words encompass not only spelling similarities but also their co-occurrence in definitions.

By analyzing this graph, we can delve into the semantic connections between words, revealing more intricate and subtle associations among them that go far beyond traditional spelling and part-of-speech relationships.

**Problem**

Embarking on our journey to explore the profound connections between words in the lexicon, we set out to uncover the potential intimacy between two isolated vertices. Our aim is not merely to find the distance between them but to discover how similar their "friend sets" are. It's akin to searching for a hidden passage between two islands in the vast ocean of text, a passage carrying their connections and similarities.

By retracing the definitions of these words, we form their "friend sets," representing companions that co-occur in their definitions. Subsequently, we employ algorithms to measure the similarity between these two friend sets, somewhat analogous to comparing the fauna and flora in two forests during an adventure.

In the end, we identify the two most similar points in this ocean of vocabulary, revealing a connection beyond our expectations, much like discovering a hidden bridge between two distant islands, linking their definitions and semantics. This finding not only adds a new dimension to our exploration but also unveils the subtle and mysterious ties within this lexical city.

**Step**

1. Find a suitable dataset.
2. Explore valuable questions related to the identified dataset.
3. Write corresponding code based on the questions.
4. Test each module and functionality.

**Test**

1. The FoF similarity measure is a way to quantify the extent to which two vertices share common neighbors, providing insights into the similarity of their local neighborhood structures in the graph. We tested with vertices 50 and 1, and the result indicates that these two vertices have no common neighbors and are not connected.
2. At the same time, the distance between vertex 0 and vertex 1 was calculated. Since these two vertices are connected, the return value is 1.
3. We use a function implements a graph clustering algorithm. It randomly selects k initial representative nodes and iteratively updates them. During each iteration, it assigns each node to the nearest representative node based on the average distance. After assigning nodes to clusters, it updates the representative nodes to the average value for each group. This process is repeated for a fixed number of iterations (in this case, 10), and the final representative nodes are printed.
4. We use a function uses a greedy approach to find the densest subgraph by iteratively adding neighboring nodes to the current subgraph until no more nodes can be added. The density of each subgraph is calculated and compared, and the information about the densest subgraph is printed at the end.

