19CSE212
Data Structures and Algorithms
CASE STUDY

Team
APOORVA S B : CB.EN.U4CSE21205

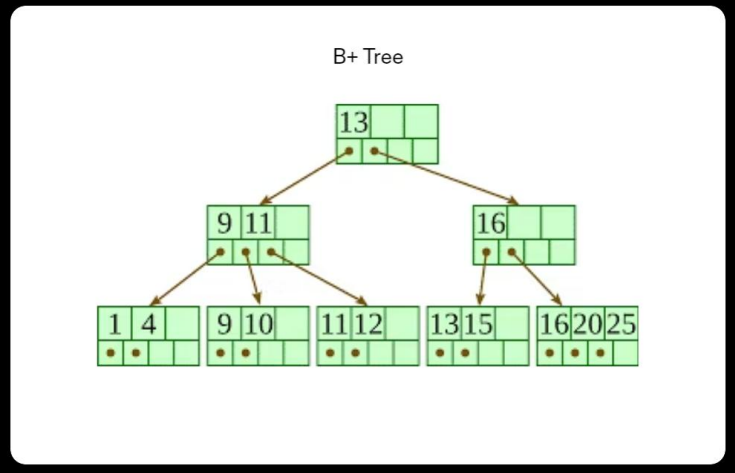
M SHIVA: CB.EN.U4CSE21235

Automatic Text Summarization Using Hybrid Data Structures

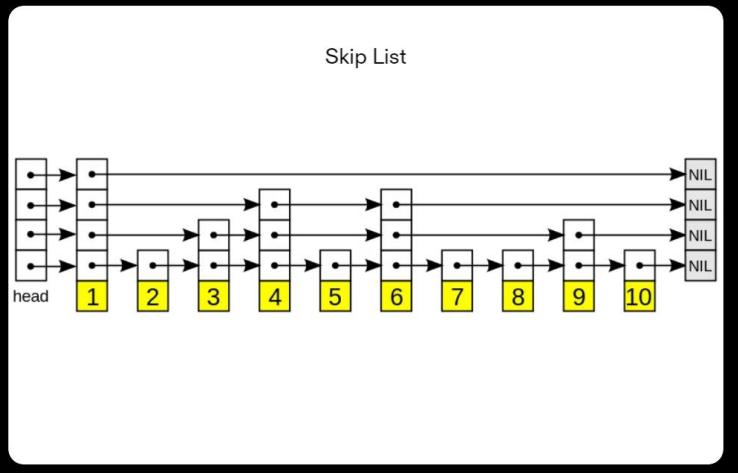
Hybrid Data Structures

- Data structures are used to organize and access data efficiently. Combining different data structures into a hybrid structure allows for greater flexibility and efficiency when performing coding tasks. Hybrid data structures are used in real-life applications, such as the use of B+ trees for indexing and retrieval of a large amount of data. Another example of a hybrid data structure is the skip list, which combines linked lists with binary search trees and allows for insertion, deletion, and searching with an average time complexity of O(log n). Different hybrid data structures have different appearances and are used for different purposes.

B+ Tree



Skip List



Text Summarization

- Our automatic text summarization project employs a hybrid data structure that combines a hash map and a heap. The hash map stores and retrieves word frequencies efficiently, while the heap tracks top-ranked sentences. This combination creates a fast and efficient system for automatic text summarization using the TF-IDF method.
- **TF - IDF**
- TF-IDF is a metric that uses term frequency and inverse document frequency to determine the importance of sentences in a text corpus. Term frequency ranks words based on their frequency of occurrence in the text file. Hans Peter Luhn's definition of the term frequency states that:
- *“The weight of a term that occurs in a document is simply proportional to the term frequency.”*
- Inverse document frequency reduces the weight of stop words and emphasizes words crucial to the context in the TF-IDF method. Combining term frequency and inverse document frequency enables the ranking of sentence importance, resulting in effective summaries. The hybrid data structure and TF-IDF method are valuable tools for automatic text summarization.

PageRank Algorithm

- The second implementation was using the PageRank algorithm, which uses graphs and heaps to summarize sentences based on the similarity parameter. Each sentence is compared and an edge is added based on the cosine similarity factor if the similarity>=0.5. The number of incoming and outgoing edges is used to determine the importance of a sentence. A node with multiple incoming and outgoing edges is generally not very important as several other sentences with similar meanings are present in the text. Hence, combining the graph data structure with a min heap, ordered based on the number of incoming and outgoing nodes, helps in picking the important sentences.

TF-IDF

- Chosen Hybrid Data Structure:

- This hybrid data structure merges a hash table and a heap.
- It efficiently stores and retrieves TF-IDF scores.
- The hash table is implemented using a dictionary of dictionaries.
- Sentences are stored in a heap.
- The data structure can quickly collect items of a required frequency or parameter.
- It functions as a priority queue with updateable priorities.

PageRank

- Chosen Hybrid Data Structure:

- The hybrid data structure is a graph with a heap.
- The hybrid data structure maintains the connection between different sentences, based on their similarity, while the heap ranks them based on their incoming and outgoing edges.
- The use of this data structure allows for efficient maintenance of the similarities and the ordering and extraction of sentences.

Design Choices and Tradeoffs:

- The tradeoff is that TF-IDF or PageRank is not the most accurate algorithm for extractive text summarization and it might not give the most concise text summary always. The implementation allows the user to enter text in one field and details like several sentences are received output almost instantly. The entire code is then presented with a GUI made using the Tkinter package. The code is built on the widely available open-source natural language toolkit NLTK.

Design Choices and Tradeoffs:

- The data sets constitute of internet articles as well as the NLTK corpus, especially the state_union data set which comprises recorded presidential speeches. The TF-IDF summarizer was effective in summarizing the given texts and had drawbacks at very few points, wherein there was an anomaly in the text, such as a heading or a quote mentioned as it is. Hence, to evaluate this model specifically, we have considered data sets with only information and no quotes or dates, which cause wrong outputs due to incorrect processing. Discuss the datasets used and any specific considerations for the experiments.

Conclusion and Discussion:

- The data structures together can efficiently rank and summarize a given input text using the term frequencies and inverse document frequencies as a numerical standard for the text. The results obtained from both algorithms appear to be efficiently summarized, and the overall result with testing thus far is that both algorithms are almost equally effective in summarizing a given text. The result of the project is a GUI-based application that runs on a hybrid data structure background, comprised of a hash map and a heap. The performance of the implementation varies based on the input size and data type, however, when provided with plain data conveyed by text, it displays great accuracy in selecting top sentences. However, to overcome these few drawbacks and to address comparison metrics, we have implemented another algorithm called the Page Rank algorithm, popularized by Google as a hybrid data structure model of a graph and a heap.