ADB Project Design Phase

Name	Section	B.N
Khaled Hesham Sayed	1	22
Abdelaziz Salah Mohammed	2	3
Kirollos Samy Hakim	2	13
Karim Mahmoud Kamal	2	12

Supervisor
Eng. Abdelrahman kaseb

Approximate Nearest Neighbor Search using Hierarchical Navigable small-world graphs (HNSW)

Introduction:

Hierarchical Navigable Small World (HNSW) is an indexing method used in computer science and information retrieval to efficiently search and retrieve nearest neighbors in high-dimensional spaces. It was introduced as an improvement over traditional methods like KD trees and ball trees, especially in scenarios with large datasets and high-dimensional feature spaces as in our case as we have 70 dimensions and over 20M records.

Here's a brief description of the key features of the HNSW indexing method:

1. Hierarchical Structure: HNSW organizes data in a hierarchical manner, creating a **multi-level graph structure**. Each level represents a different granularity of the data, with finer details at lower levels and coarser details at higher levels.
2. Navigable Small World: HNSW builds a **"small-world"** network, meaning that even though the graph is hierarchical, each node has connections to nodes that are not strictly at the same level. This small-world property allows for efficient navigation between different levels of the hierarchy during the search process.
3. Efficient Nearest Neighbor Search: One of the primary objectives of HNSW is to provide a fast and effective mechanism for finding nearest neighbors in high-dimensional

spaces. The hierarchical and small-world structure enables efficient pruning of the search space, reducing the number of distance computations needed during the search process.

4. Scalability: HNSW is designed to be scalable, performing well on large datasets with high-dimensional features. It manages to achieve this scalability by maintaining the hierarchical structure and small-world connections.
5. Trade-off between Accuracy and Speed: HNSW offers a trade-off between search accuracy and speed. It provides a good balance, making it suitable for applications where real-time or near-real-time search is essential, such as in recommendation systems, image retrieval, and other similar domains.

Overall, HNSW is a powerful indexing method that addresses some of the challenges associated with high-dimensional data retrieval, making it a valuable tool in various machine learning and information retrieval applications.

Why to work with HNSW:

The choice of indexing method depends on the specific requirements and characteristics of the data and the application. Here are some reasons why you might consider using HNSW over other indexing methods, along with examples of other indexing methods:

Advantages of HNSW:

1. High-Dimensional Data: HNSW is particularly well-suited for high-dimensional data, where traditional methods like KD-trees and ball trees may become less effective due to the curse of dimensionality.

2. Efficiency in Nearest Neighbor Search: HNSW excels in providing fast and efficient nearest neighbor search, making it suitable for applications such as similarity search in large datasets.
3. Scalability: HNSW maintains good performance as the dataset size increases, making it scalable for applications with growing data.
4. Small-World Property: The small-world property of HNSW allows for effective navigation between different levels of the hierarchy, enabling quicker searches.

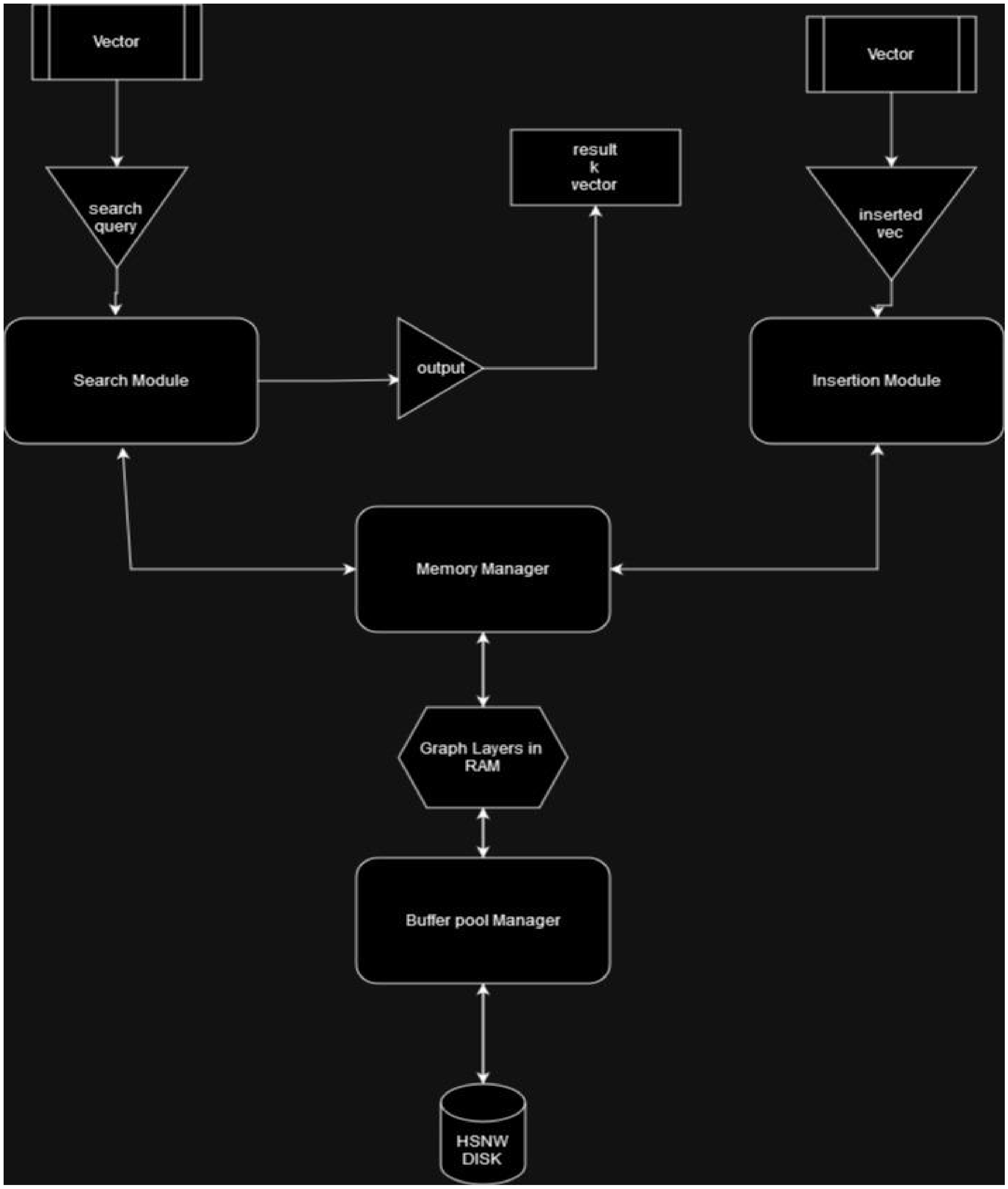
Examples of Other Indexing Methods:

1. KD-Trees (K-Dimensional Trees): A space-partitioning data structure that recursively divides the data space into regions, often used for multidimensional search keys.
2. Locality-Sensitive Hashing (LSH): A technique that uses hash functions to map similar data points to the same buckets with high probability, facilitating efficient approximate nearest neighbor search.
3. Inverted Index: Commonly used in text retrieval, an inverted index maps terms to documents that contain those terms, enabling fast keyword-based searches.

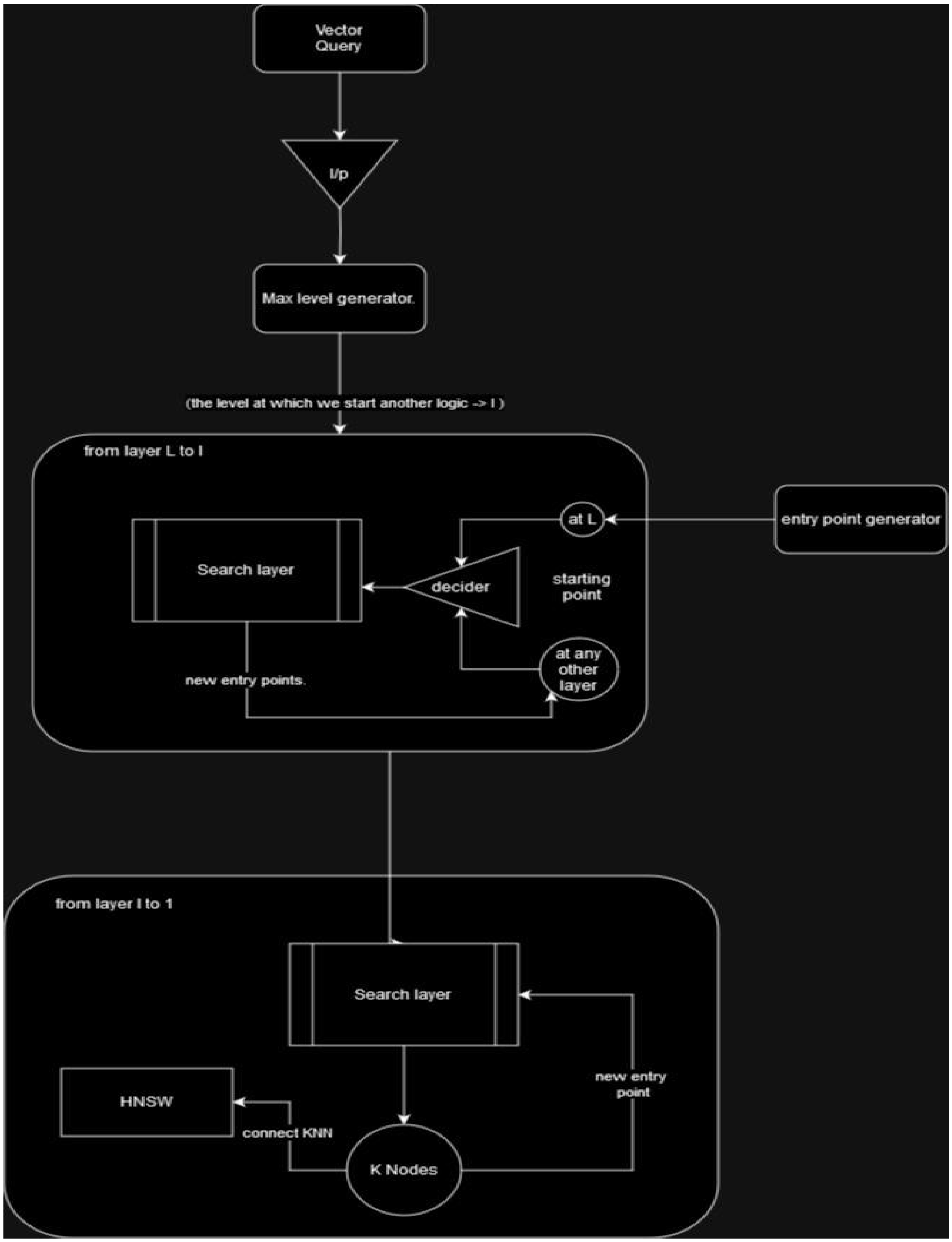
The choice between these methods depends on factors such as the dimensionality of the data, the size of the dataset, the nature of the queries, and the desired trade-offs between search accuracy and speed. HNSW is particularly attractive when dealing with high-dimensional data and the need for efficient nearest-neighbor searches (**which is exactly our case**).

<u>Design of high level modules:</u>

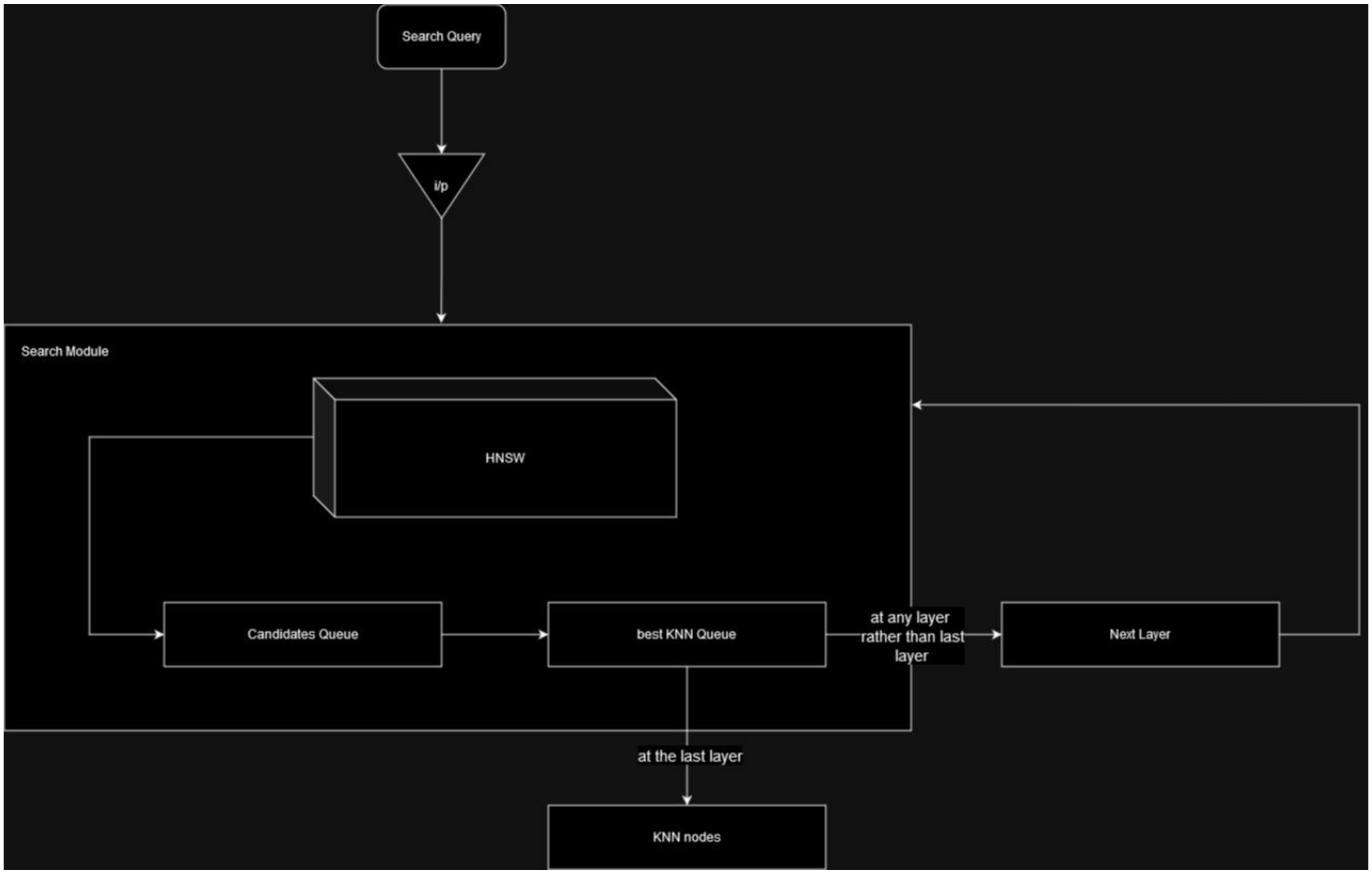
Top Module



Insertion Module



Search Module



Reference:
https://arxiv.org/ftp/arxiv/papers/1603/1603.09320.pdf
https://www.pinecone.io/learn/series/faiss/hnsw/
https://arxiv.org/abs/1603.09320
https://zilliz.com/learn/hierarchical-navigable-small-worlds-HNSW
https://www.crunchydata.com/blog/hnsw-indexes-with-postgres-and-pgvector