Context

Nearest neighbor search is a fundamental algorithmic primitive that is employed extensively across several fields including computer vision, information retrieval (recommender systems or web search), and machine learning (retrieval-augmented generation for LLMs). Given a large pointset \( P \subset \mathbb{R}^d \), we want to quickly retrieve the point \( p \in P \) with minimal distance to a given query point \( q \in \mathbb{R}^d \). In low-dimensional space \((d < 4)\), the problem is well-understood and solved by standard data structures such as \( k\)-d trees. With high dimensionality however \((d > 50)\) exact search effectively requires a linear scan in theory and practice. Therefore, one is primarily interested in approximate search. The current state-of-the-art techniques use so-called navigable search graphs on the pointset \( P \). Popular examples are DiskANN and HNSW. The vertices are the points \( P \) and edges connect points such that a simple greedy search quickly converges to the solution. In each step, the greedy search expands to those neighbors of the current vertex which are closest to the query-point by distance.

Topic

The goal of this work is to develop improved search procedures as well as optimize the search graph in a post-processing step. Some initial ideas are as follows.

- Construct a hierarchical search graph with ranks assigned to vertices. Develop a greedy search that preferably explores downward edges in the hierarchy and only backtracks or explores upward edges when it gets stuck.
- Optimize the search graph navigability by simulating queries at construction time, and inserting edges that would further speed up the search.

Requirements

- Interest in algorithms and data structures
- Firm programming skills in modern C++

Literature

- Full Delaunay Hierarchies [Link](https://epubs.siam.org/doi/abs/10.1137/19781611972900.5)
- DiskANN [Link](https://suhasjs.github.io/files/diskann_neurips19.pdf)
- HNSW [Link](https://arxiv.org/abs/1603.09320)