STATUS REPORT

Group:
Deepak Kukkapalli [vk23p]
Karthik reddy Vemireddy [kv23b]
Muqueet Mohsin Shaik [ms23ch]

Finalised problem statement:
To find the ***smallest k-core subgraph*** containing query vertices. This is very helpful for applications like targeted recommendations (advertisements, social media suggestions), biological analysis etc. The issues with the current solutions, such as the Greedy algorithms, are that they are NP-hard problems. So they lack quality guarantees and often give outputs with oversized subgraphs, increasing costs and reducing their relevance.

Basic goal and motivation:
To develop a ***Progressive Search Algorithm (PSA***) that computes near-optimal k-core subgraphs with provable theoretical bounds on subgraph size. By the end of the semester, we aim to:

- Implementing PSA with integrated lower and upper-bound computations.
- Validating its performance on real-world datasets, ensuring it achieves a user-defined approximation ratio while minimising runtime and subgraph size.

Assumptions and Methods:
- **Progressive Refinement:** Tightening the bounds iteratively, which allows us to terminate early if we reach a desired approximation ratio.
- **Difference:** Unlike Greedy techniques, PSA provides theoretical quality bounds and avoids oversized outputs through bound-guided refinement.

Software tools and data sets:
- ***Languages/Frameworks:*** C++ (STL: vectors, unordered maps, priority queues).
- ***Algorithms:*** DFS, adjacency list traversal, heap-based prioritisation.
- ***Datasets:*** Real-world graphs from SNAP (Email, Yelp, YouTube) and KONECT.

Experimental Plan:

- **Baseline Comparison:** To compare PSA's subgraph size, runtime, and approximation ratio against S-Greedy results from the paper on Email and Yelp data sets to begin with.
- **Scalability Testing:** To measure runtime and memory usage for increasing graph sizes and to try it on a larger dataset like YouTube.

Current status and partial results:

- Successfully implemented lower bound search algorithms, and were able to run and test out the email dataset, wherein we were getting the partial vector nodes of the lower bound k core subgraph.

Plan for the remaining time:

- Moving forward we will start implementing the upper bound heuristic-based algorithm. By having lower and upper bound subgraphs on hand, we will try to reduce the gap between both the lower and upper bound search to a point where it is less than or equals an approximation ratio (c=1.8), such that the optimal k core subgraph is guaranteed.