Graph Algorithms

Overview

The task of this assignment is to investigate different algorithms for a few selected graph analysis problems, to evaluate them and finally to choose one for each problem and implement it in a system of your choice. Note that for every step in this task you should explain your choices and decisions.
Also your solution should contain the code (and data is possible) that you used for testing.

Problems

1. **Strongly Connected Components:** Find all strongly connected components in a graph
2. **Path Finding:** For a given start vertex, find paths to other vertices in a graph
3. **Ranking:** Compute a ranking for the vertices in a graph

Objective 1

For each of the problems listed above, investigate different algorithms that solve the problem. Evaluate different properties of the algorithms that might be important for an implementation. Finally select one algorithm that you would implement to solve the problem. Explain your choice. Some aspects that you should consider might be

- time complexity
- memory consumption
- potential for parallelization
- restrictions
- application; e.g. for Ranking, on what type of graphs does the algorithm make sense, how can the result be interpreted
- feel free to find more aspects

Objective 2

Implement the three algorithms you chose in the first task in a system and/or language of your choice. Explain your design choices. Note that we will evaluate your solution for performance (or potential bottlenecks) and readability of the code.

Some systems you might want to consider are PGX, SPARK, SNAP or GraphLab but you are free to implement your own. Note that this task requires you to actually implement the algorithm – using built-in functions that do the job is not an option.

In case you chose to implement your own system: Of course you do not have to write an entire graph analytics system. The algorithms including the data-structures you want to use are sufficient – but explain your design choices.