CSCI 102 assignment 7 – Efficient graph implementations

April 14, 2024

In this assignment you will implement two `Graph` datastructures.

- Implement `Graph` with an adjacency map, `AdjacencyMapGraph`. In this implementation, `InnerVertex<V>` will have two maps keeping track of its incoming and outgoing edges. Avoid keeping track of a list of edges. Which methods need to be modified? Which are sped up or slow down? You may use the unsorted map implementation we saw in class, but keep in mind a hash map is a better option.
- Implement `Graph` with an adjacency matrix, `AdjacencyMatrixGraph`. In this implementation, in addition to lists of vertices and entries, keep a list of lists which contains the edge connecting vertex i and j in position (i, j). Remake the entire matrix when adding or removing a vertex. Which methods need to be modified? Which are sped up or slow down?
- In both implementations, write a main method that adds nodes “A”, “B”, “C”, “D”, and 12 edges between the nodes that store the concatenation of the letters, i.e. the edge from “A” to “B” stores “AB” and that from “B” to “A” stores “BA”. Print all the vertices and edges. Then remove “B” and “C” and print all the vertices and edges.

Please submit your code and answers to the questions in a zipped folder on Brightspace by April 22.