

Algorithms and Data Structures for Data Science

Lab_adjList

CS 277

Brad Solomon

April 21, 2023



UNIVERSITY OF
ILLINOIS
URBANA - CHAMPAIGN

Department of Computer Science

Learning Objectives

Implement core graph ADT functions using adjacency lists

Introduce modifications to graphs for storing weights

Graph ADT

Find

`getVertices()` — return the list of vertices in a graph

`getEdges(v)` — return the list of edges that touch the vertex v

`areAdjacent(u, v)` — returns a bool based on if an edge from u to v exists

Insert

`insertVertex(v)` — adds a vertex to the graph

`insertEdge(u, v)` — adds an edge to the graph

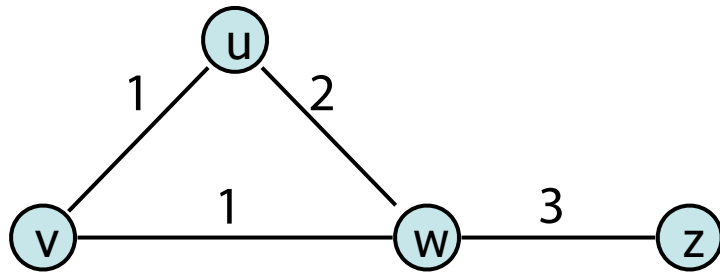
Remove

`removeVertex(v)` — removes a vertex from the graph

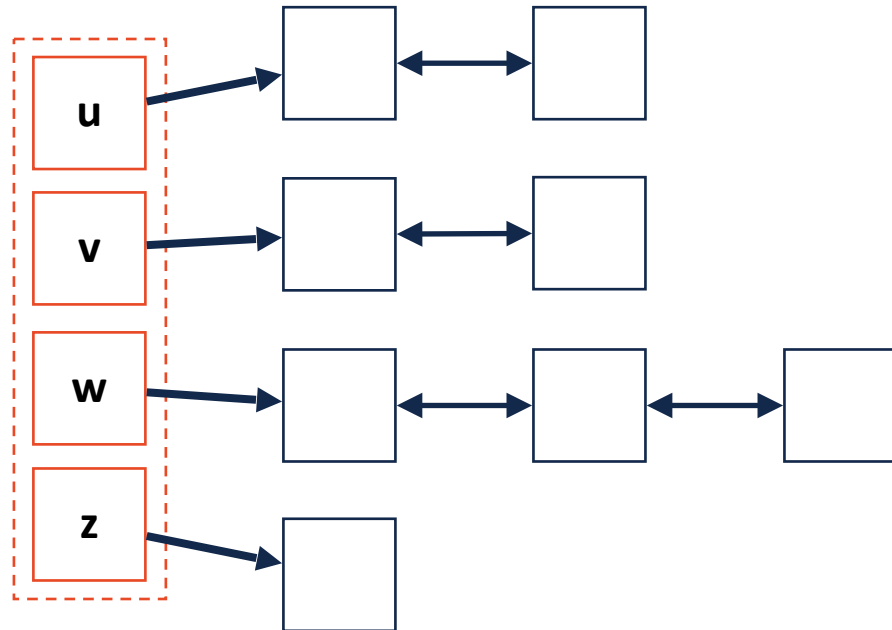
`removeEdge(u, v)` — removes an edge from the graph

Weighted Adjacency List

Vertex Storage:

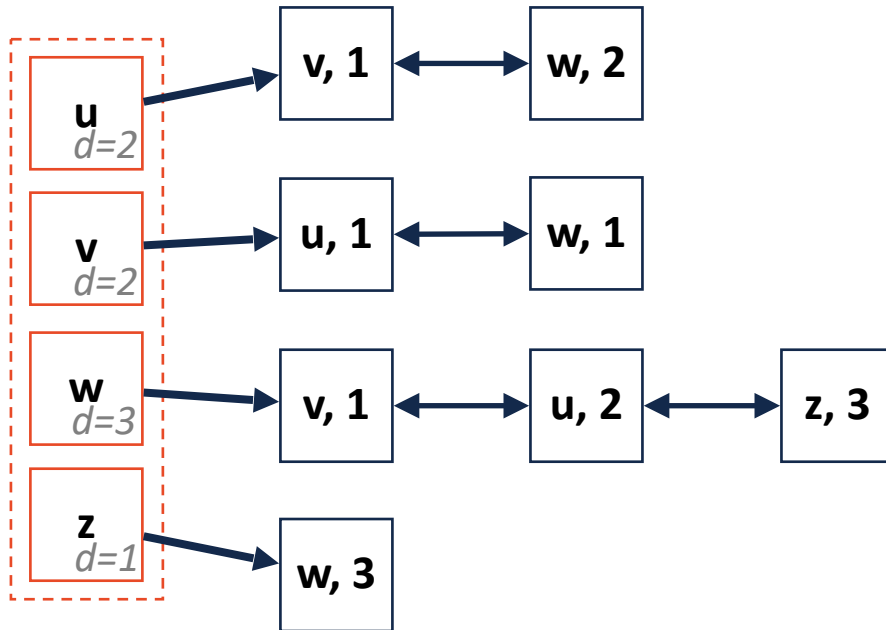
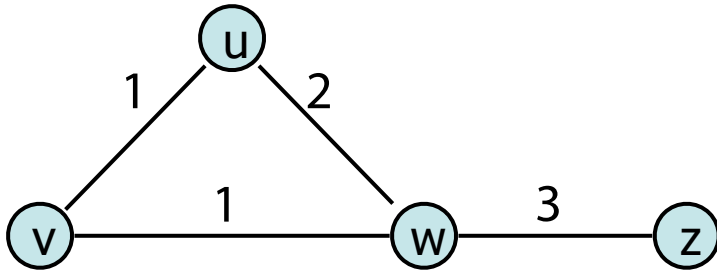


Edge Storage:



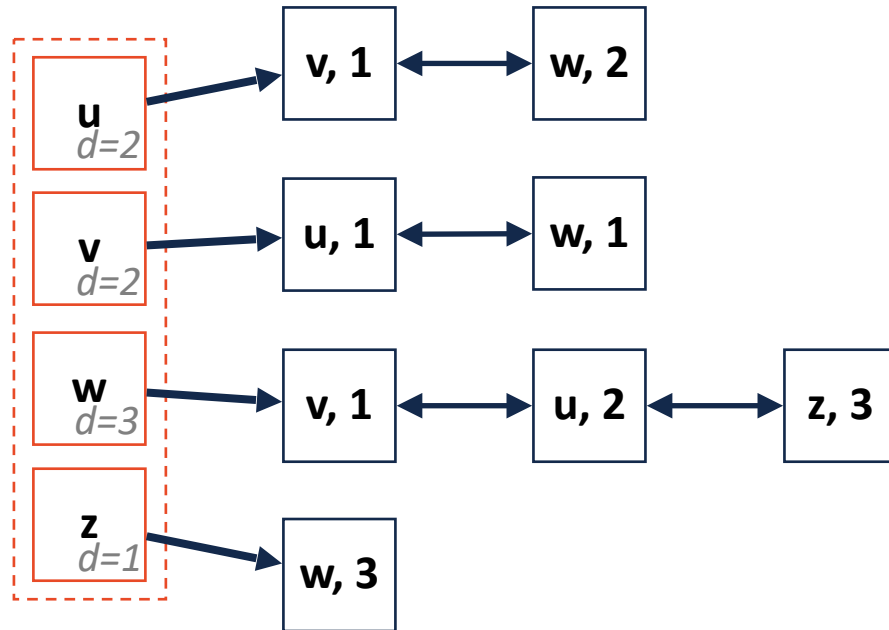
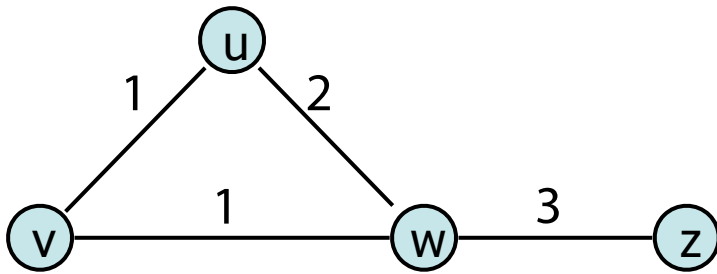
Graph Implementation: Adjacency List

insertVertex(v):



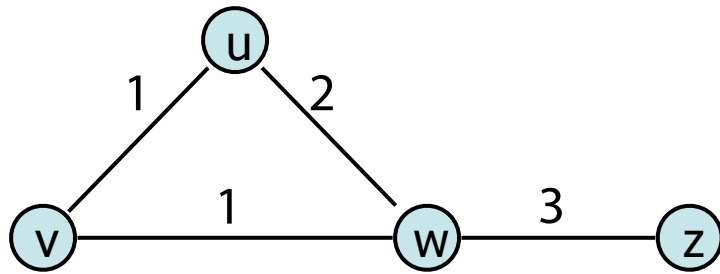
Graph Implementation: Adjacency List

insertEdge(u, v, w):

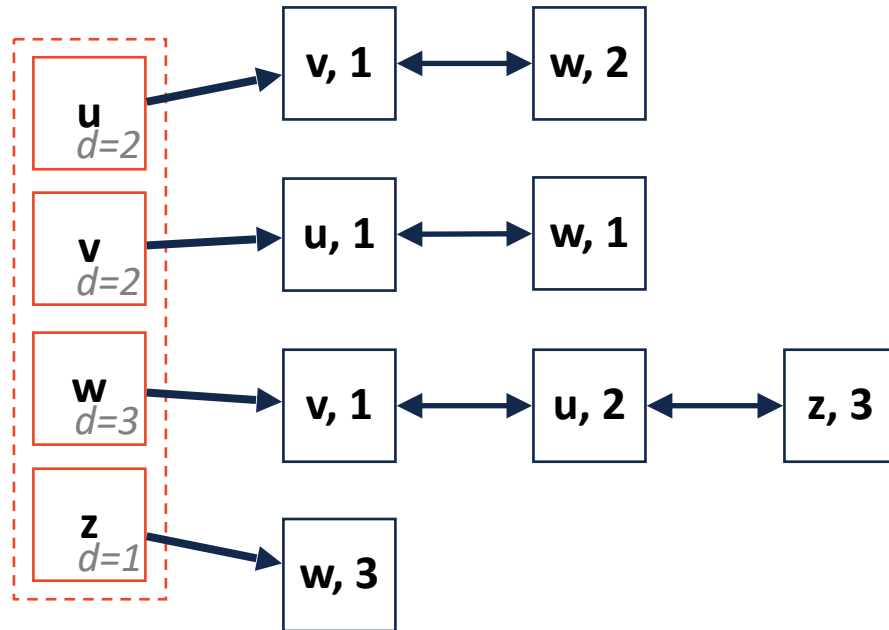


Graph Implementation: Adjacency List

removeVertex(v):



removeEdge(u, v):



Graph Implementation: Adjacency List

getAllEdges(w):

