

Algorithms and Data Structures for Data Science

lab_graph

CS 277

Brad Solomon

March 29, 2024



UNIVERSITY OF
ILLINOIS
URBANA - CHAMPAIGN

Department of Computer Science

Learning Objectives

Practice coding graph implementations using lists and matrices

Review a conceptual understanding of graph implementations

Compare and contrast the efficiency of edge list and adj matrix

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

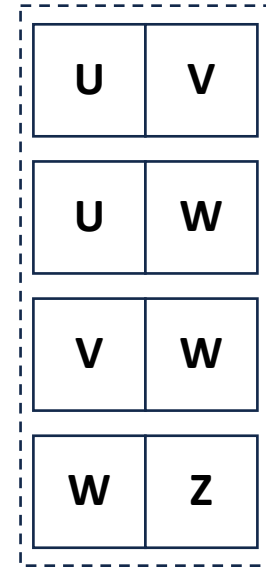
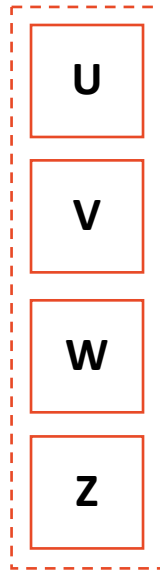
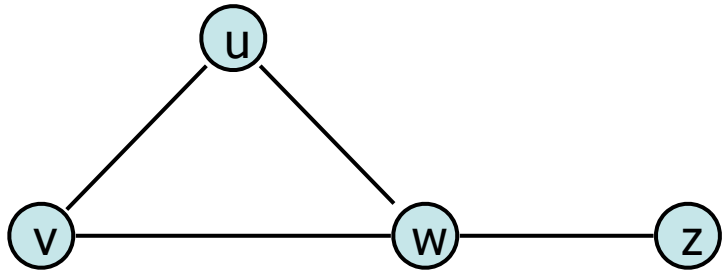
`insertEdge(u, v)` — adds an edge to the graph

Remove

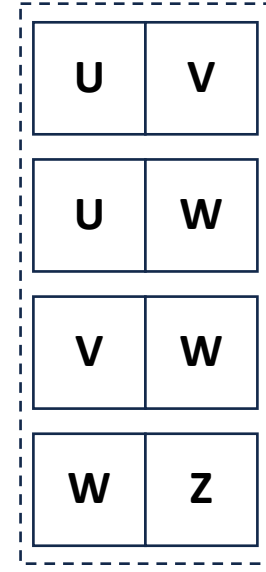
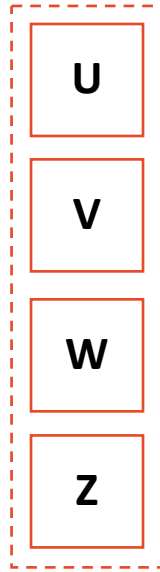
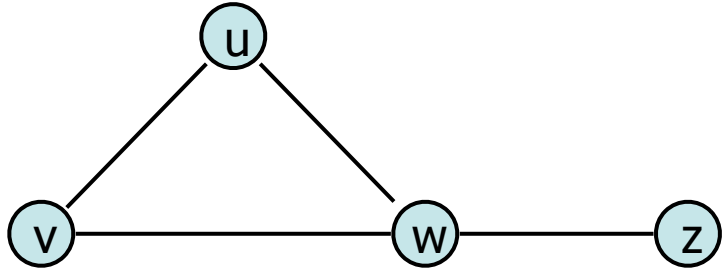
`removeVertex(v)` — removes a vertex from the graph

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

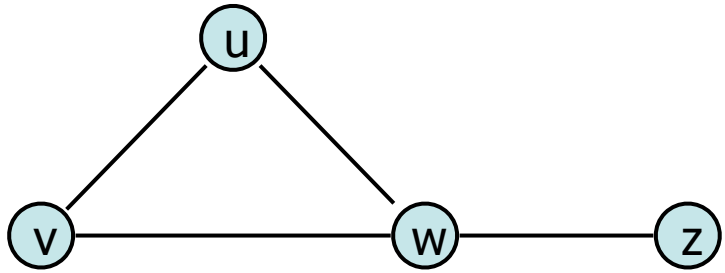
insertVertex('A')



`insertEdge('A', 'u')`



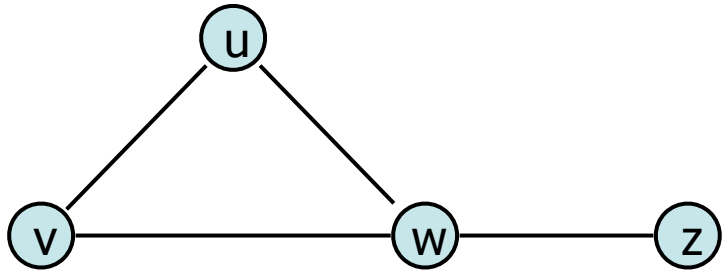
insertVertex('A')



u	0
v	1
w	2
z	3

	u	v	w	z
u	0	1	1	0
v	1	0	1	0
w	1	1	0	1
z	0	0	1	0

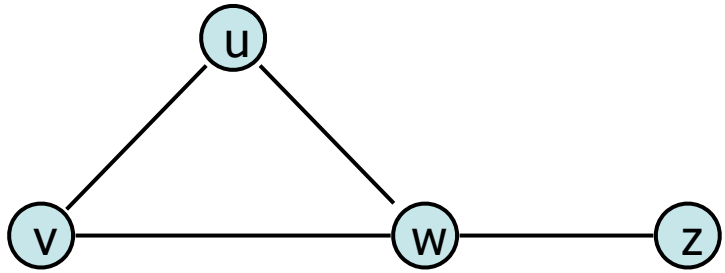
removeVertex('v')



u	0
v	1
w	2
z	3

	u	v	w	z
u	0	1	1	0
v	1	0	1	0
w	1	1	0	1
z	0	0	1	0

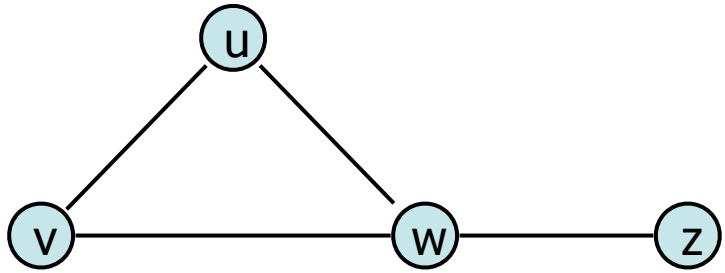
`insertEdge('A', 'u')`



u	0
v	1
w	2
z	3

	u	v	w	z
u	0	1	1	0
v	1	0	1	0
w	1	1	0	1
z	0	0	1	0

`removeEdge('v', 'w')`



u	0
v	1
w	2
z	3

	u	v	w	z
u	0	1	1	0
v	1	0	1	0
w	1	1	0	1
z	0	0	1	0