

# Algorithms and Data Structures for Data Science

## Lab\_adjMatrix

CS 277

Brad Solomon

April 14, 2023



UNIVERSITY OF  
**ILLINOIS**  
URBANA - CHAMPAIGN

Department of Computer Science

# Learning Objectives

Implement core graph ADT functions using adjacency matrices

# 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

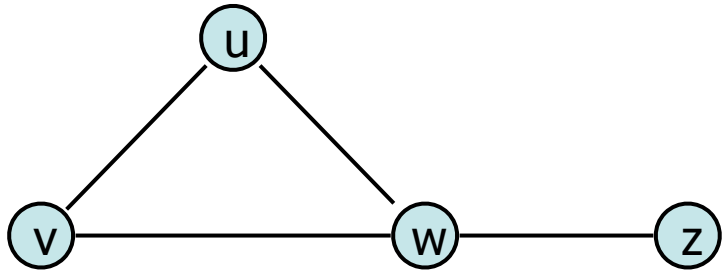
`insertEdge(u, v)` — adds an edge to the graph

## Remove

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

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

# Graph Implementation: Adjacency Matrix

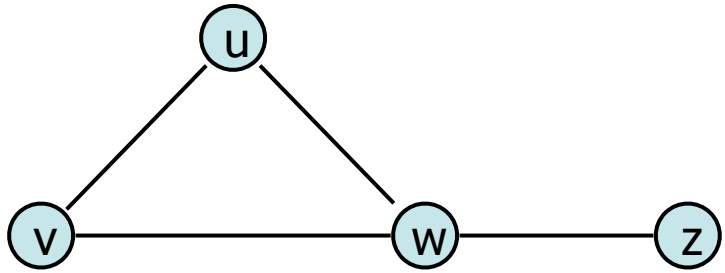


**insertVertex(v):**

|   |   |
|---|---|
| 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 |

# Graph Implementation: Adjacency Matrix

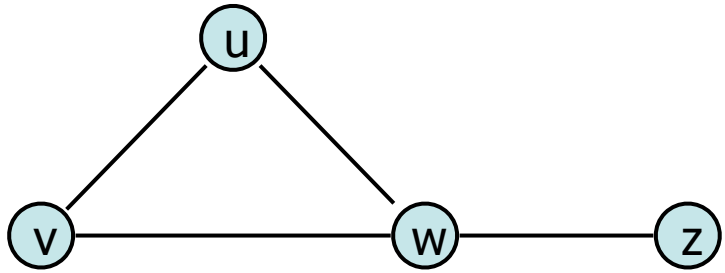


**insertEdge(u, v):**

|   |   |
|---|---|
| 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 |

# Graph Implementation: Adjacency Matrix



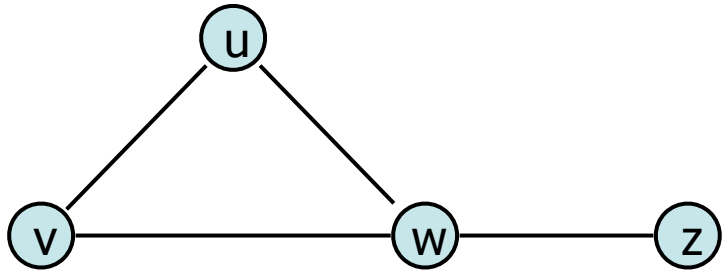
**removeVertex(v):**

|   |   |
|---|---|
| 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 |

**removeEdge(u, v):**

# Graph Implementation: Adjacency Matrix



**getDegree(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 |