Algorithms and Data Structures for Data Science

lab_graph

CS 277

March 29, 2024

Brad Solomon

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

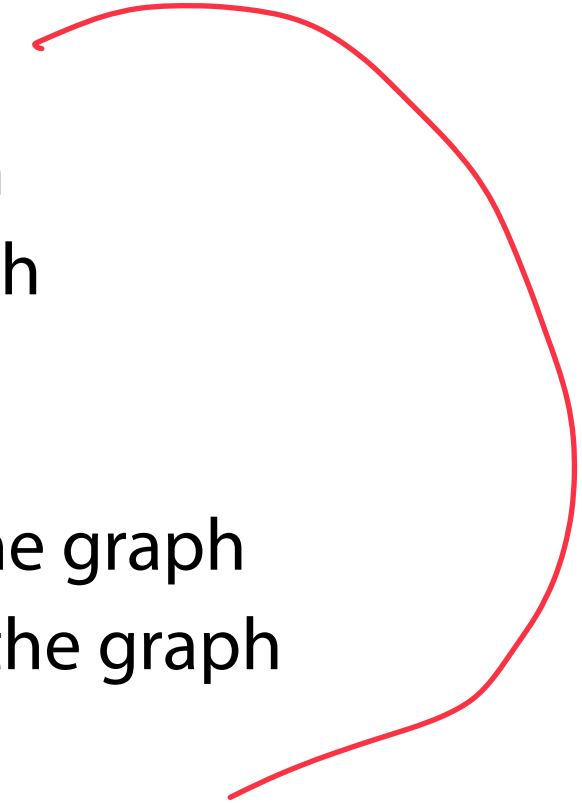
insertVertex(v) — adds a vertex to the graph

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

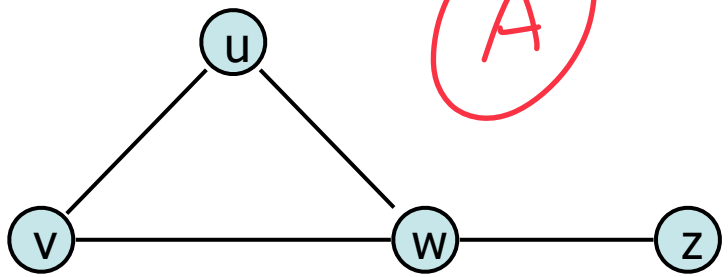
Remove

removeVertex(v) — removes a vertex from the graph

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



insertVertex('A')



A

Picture of graph

List of vertices

U
V
W
Z

Implementation

List of edges

U	V
U	W
V	W
W	Z

stores edges that exist

utils.py

↳ class EdgeList

self.vertices = []

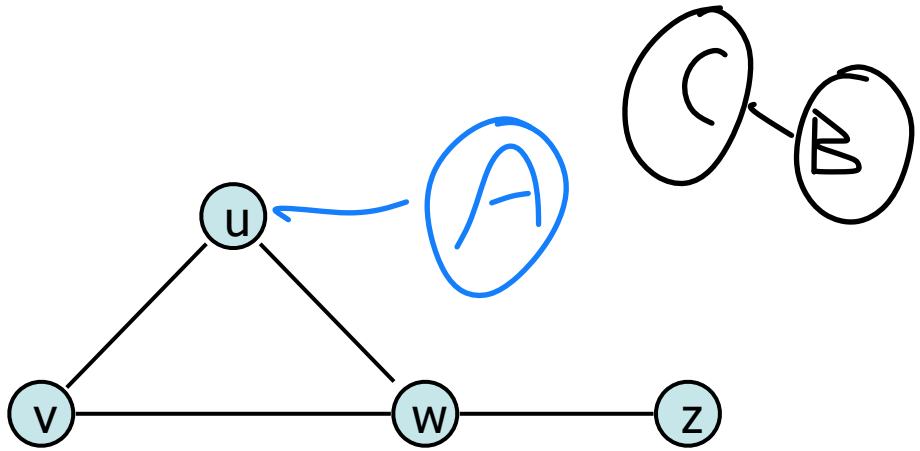
self.edges = []

Add A to list of vertices

e1.vertices

input object

insertEdge('A', 'u')



U
V
W
Z

A

(C, B)

U	V
U	W
V	W
W	Z

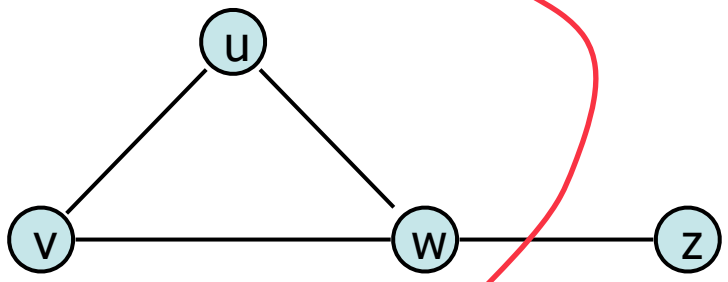
A	u

1) Add 'A' to vertex list (Because it didn't exist before)

2) Add ('A', 'u') to edge list

insertVertex('A')

am.vertices



dictionary d

↳ look up d['A']

↳ change/add d['A'] = #

d[key] = val

change 2

U	0
V	1
W	2
Z	3
A	

??

All edges possible

↓ 1 = exist, 0 = No

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

change J

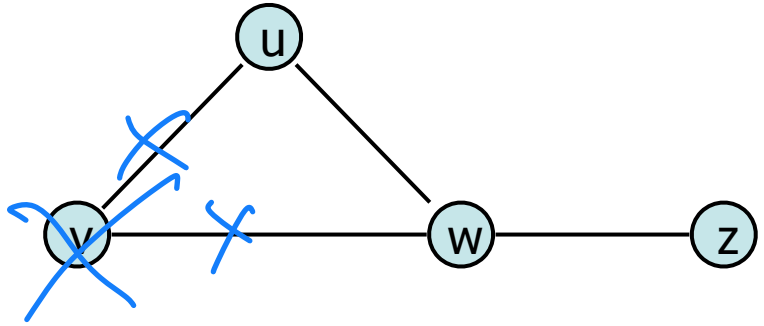
d.keys() → list (u, v, w, z)

← Does 'A' exist in dictionary?

removeVertex('v')

list.pop(0)

dictionary.pop('v')



U	0
V	1
W	2
Z	3

All keys after my remove change

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

0 u
1 w
2 z

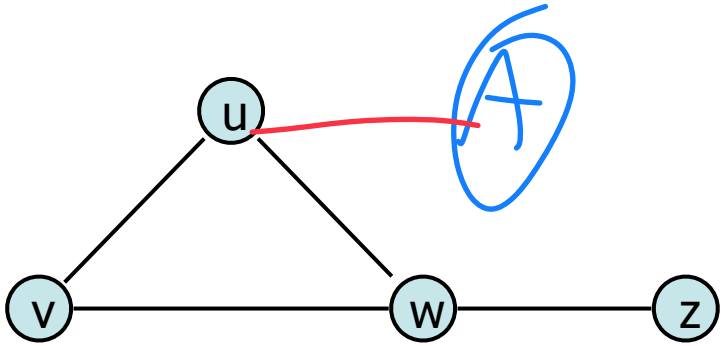
0 u	0	1	0
1 w	1	0	1
2 z	0	1	0

1) Remove vertex from vertex dictionary
   ↳ w/ pop()

2) Remove edges → list removal

3) Collect our vertex dictionary values

insertEdge('A', 'u')



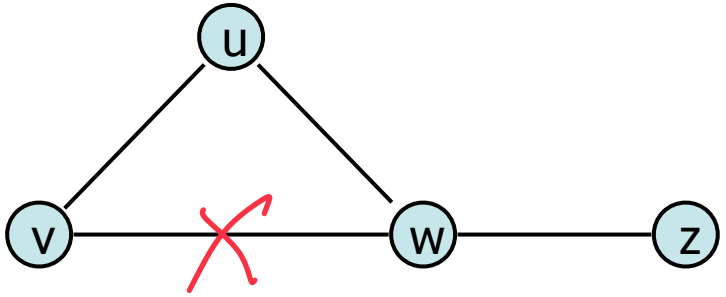
U	0
V	1
W	2
Z	3

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

1) If vertex (or vertices) are new, add vertex

2) Set value at ['A']['u'] = 1 but also ['u']['A'] = 1

`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