CS 225

Data Structures

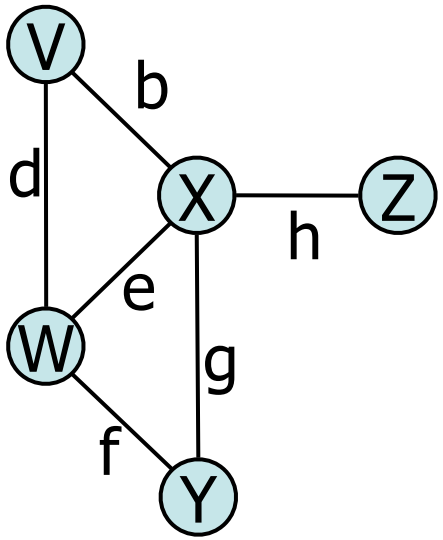
Nov. 27 – Graph Traversal

Wade Fagen-Ulmschneider

Graph ADT

Data:

- Vertices
- Edges
- Some data structure maintaining the structure between vertices and edges.



Functions:

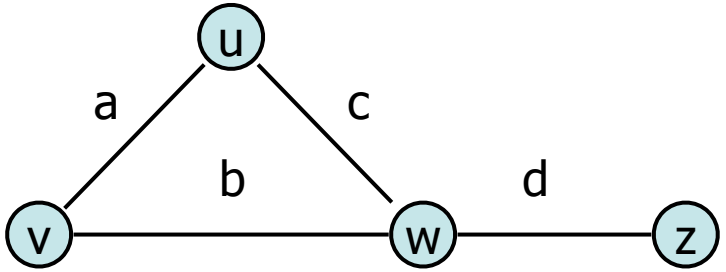
- insertVertex(K key);
- insertEdge(Vertex v1, Vertex v2, K key);

- removeVertex(Vertex v);
- removeEdge(Vertex v1, Vertex v2);

- incidentEdges(Vertex v);
- areAdjacent(Vertex v1, Vertex v2);

- origin(Edge e);
- destination(Edge e);

Edge List



Vertex List

u
v
w
z

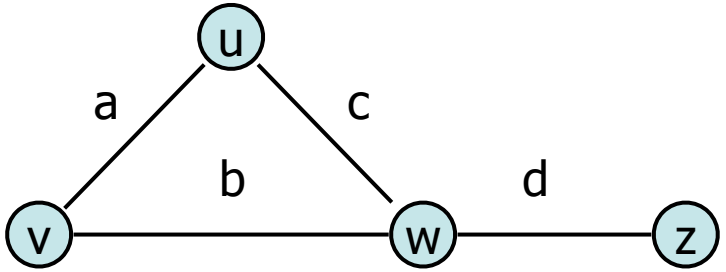
Edge List

u	v	a
v	w	b
u	w	c
w	z	d

Key Ideas:

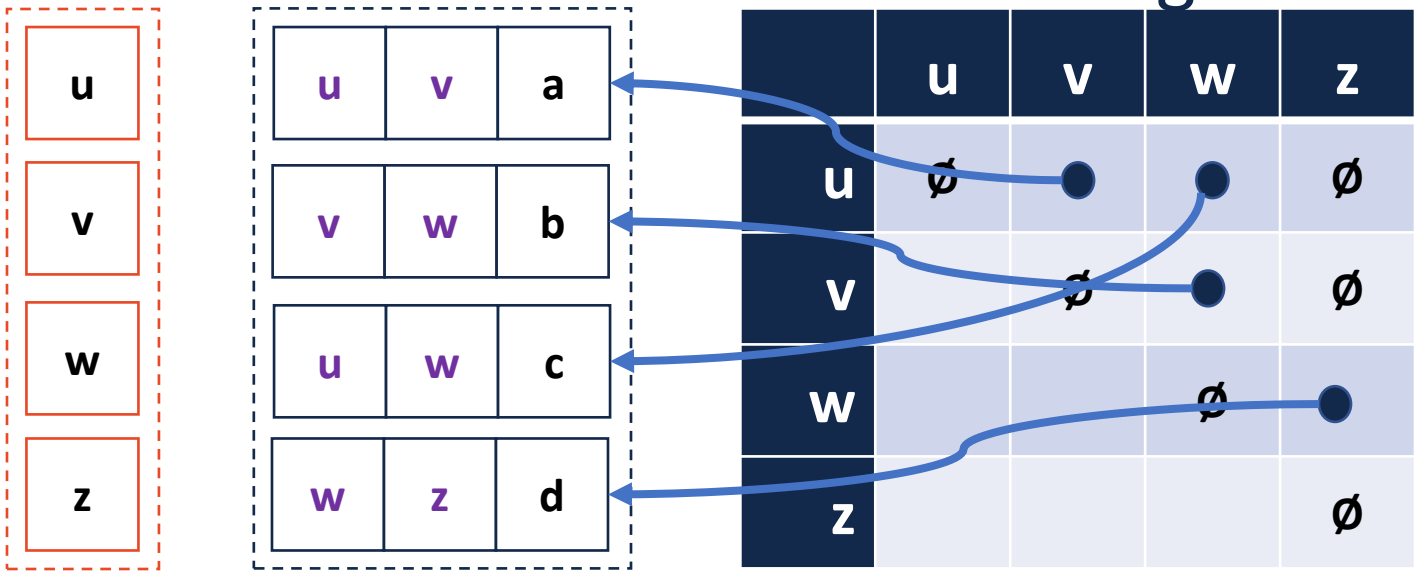
- Given a vertex, O(1) lookup in vertex list
  - Implement w/ a hash table, etc
- All basic ADT operations runs in O(m) time

Adjacency Matrix

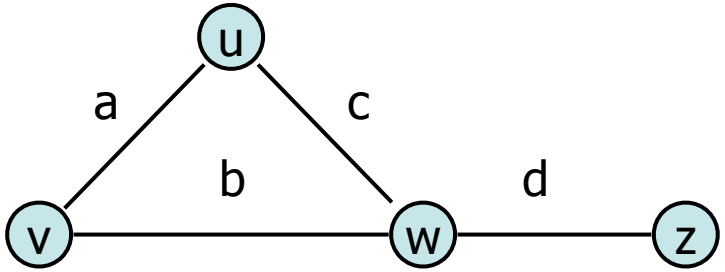


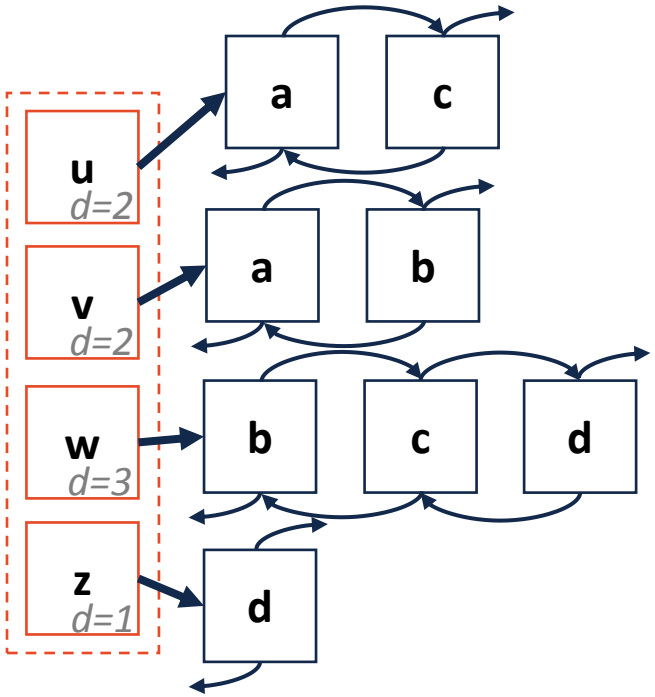
Key Ideas:

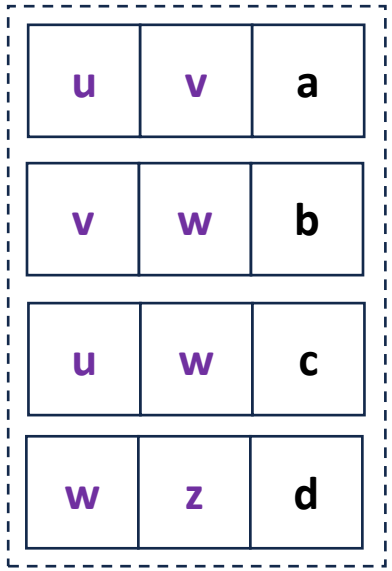
- Given a vertex, O(1) lookup in vertex list
- Given a pair of vertices (an edge), O(1) lookup in the matrix
- Undirected graphs can use an upper triangular matrix



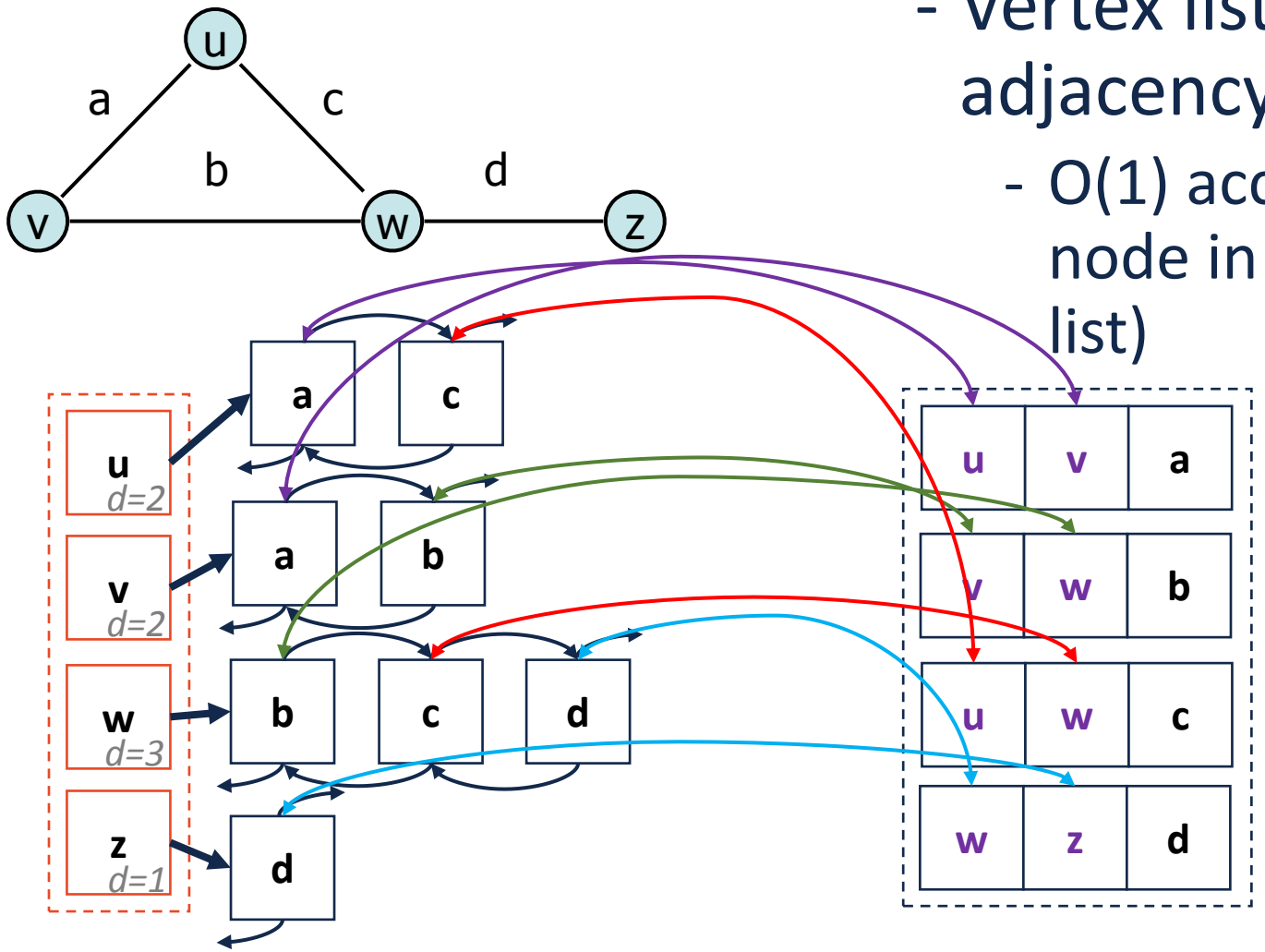
Adjacency List







Adjacency List



Key Ideas:

- O(1) lookup in vertex list
- Vertex list contains a doubly-linked adjacency list
  - O(1) access to the adjacent vertex's node in adjacency list (via the edge list)
  - Vertex list maintains a count of incident edges, or **deg(v)**
  - Many operations run in O(deg(v)), and deg(v) ≤ n-1, O(n).

Expressed as big-O	Edge List	Adjacency Matrix	Adjacency List
Space	n+m	n+m	n^2
insertVertex(v)	1	n	1
removeVertex(v)	m	n	deg(v)
insertEdge(v, w, k)	1	1	1
removeEdge(v, w)	1	1	1
incidentEdges(v)	m	n	deg(v)
areAdjacent(v, w)	m	1	min(deg(v), deg(w))

Mattox Monday

Now: Exam 11 – Theory Exam

- Exam is live right now!
- Topics: Hash Tables, Disjoint Sets, Heaps

Next Week: Exam 12 – Programming Exam

- Disjoint Set Implementations
- Graph Implementations
  - Vertex Object
  - Edge Object
  - Edge List
  - Adjacency Matrix
  - Adjacency List

Mattox Monday

POTD

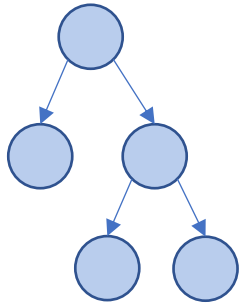
- Final weeks of POTDs are more “puzzle”/”interview”-based
- Two-day POTDs, released on M/W/F

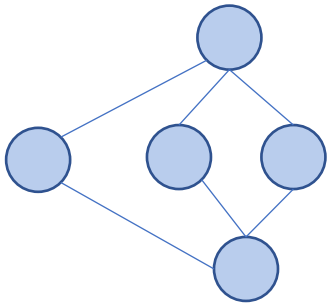
Traversal:

Objective: Visit every vertex and every edge in the graph.

Purpose: Search for interesting sub-structures in the graph.

We've seen traversal before ….but it's different:

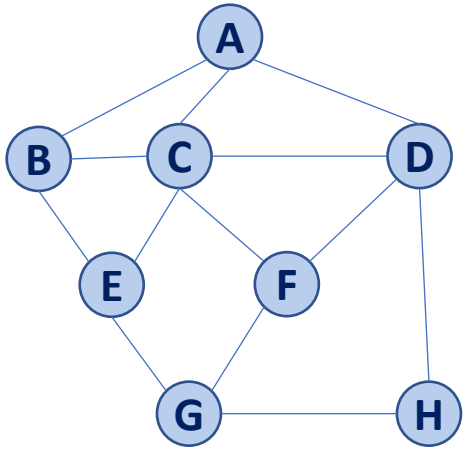




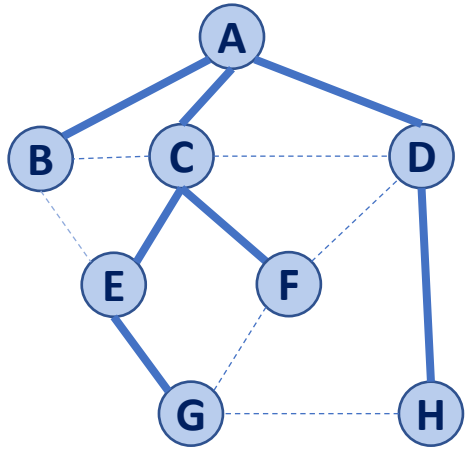
- Ordered
- Obvious Start
-

-
-
-

Traversal: BFS



Traversal: BFS



d	p		Adjacent Edges
0	A	**A**	**C B D**
1	A	**B**	**A C E**
1	A	**C**	**B A D E F**
1	A	**D**	**A C F H**
2	C	**E**	**B C G**
2	C	**F**	**C D G**
3	E	**G**	**E F H**
2	D	**H**	**D G**

```
 1  BFS(G):
 2    Input: Graph, G
 3    Output: A labeling of the edges on
 4        G as discovery and cross edges
 5
 6    foreach (Vertex v : G.vertices()):
 7      setLabel(v, UNEXPLORED)
 8    foreach (Edge e : G.edges()):
 9      setLabel(e, UNEXPLORED)
10    foreach (Vertex v : G.vertices()):
11      if getLabel(v) == UNEXPLORED:
12        BFS(G, v)
```

```
14  BFS(G, v):
15    Queue q
16    setLabel(v, VISITED)
17    q.enqueue(v)
18
19    while !q.empty():
20      v = q.dequeue()
21      foreach (Vertex w : G.adjacent(v)):
22        if getLabel(w) == UNEXPLORED:
23          setLabel(v, w, DISCOVERY)
24          setLabel(w, VISITED)
25          q.enqueue(w)
26        elseif getLabel(v, w) == UNEXPLORED:
27          setLabel(v, w, CROSS)
```

CS 225 – Things To Be Doing

Exam 11 (theory) starts Monday after break

More Info: https://courses.engr.illinois.edu/cs225/fa2017/exams/

MP6: A one week reflection MP!

Due: Friday, Nov. 17 at 11:59pm

MP7: The final MP!

Extra Credit (+14): Monday, Dec. 4 at 11:59pm
Due: Monday, Dec. 11 at 11:59pm

Lab: lab_dict this week

Due: Wednesday, Nov. 29 @ 7pm (Before the first lab after break!)

No POTDs over break

Worth +1 Extra Credit /problem (up to +40 total)