How can we represent a graph?

What are basic things we might want to do?

iterate over all such edges in a multigraph

Is there an edge from v_i to v_j?

Iterate over all edges from v_i?

Less often, iterate over all edges to v_i?

V = set of vertices in graph

<u>For Lab 4</u>

$\underbrace{\text{STL}}_{\text{3-letter acronym for airport}} \rightarrow$ airport object

Set of airports

Could use a Set versus a list.

Object "STL"

city name St. Louis

time zone

location

Expected constant time search by dest.

provide an comparator to constructor

List of edges (Flights) that originate in St. Louis

list of outgoing edges

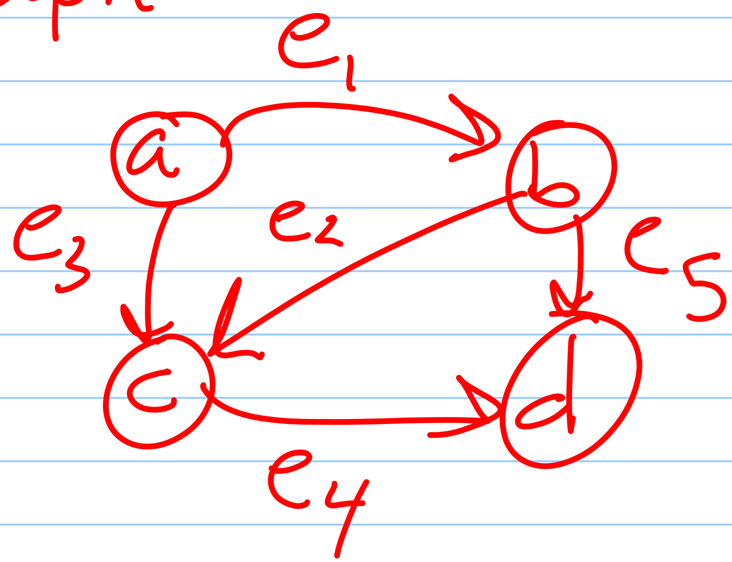
Could also keep a list of edges that terminate in St. Louis

Adjacency List

Representation of a graph where for each vertex you store a list of all outgoing edges

adjacent (in a directed sense)

Graph



Typical way to draw

$a : \overset{e_1}{\cancel{b}}, \overset{e_3}{\cancel{c}}$

$b : \overset{e_5}{\cancel{d}}, \overset{e_2}{\cancel{c}}$

$c : \overset{e_4}{\cancel{d}}$

$d :$

implicit representation of an edge when no multi-edges

incoming edges

also use edges here

a :

b : a

c : a, b

d : b, c

bucket Mapping for multigraph

If we keep a set of outgoing edges for each vertex (with comparison defined by dest)

answer is edge from V_i to V_j in expected constant time.

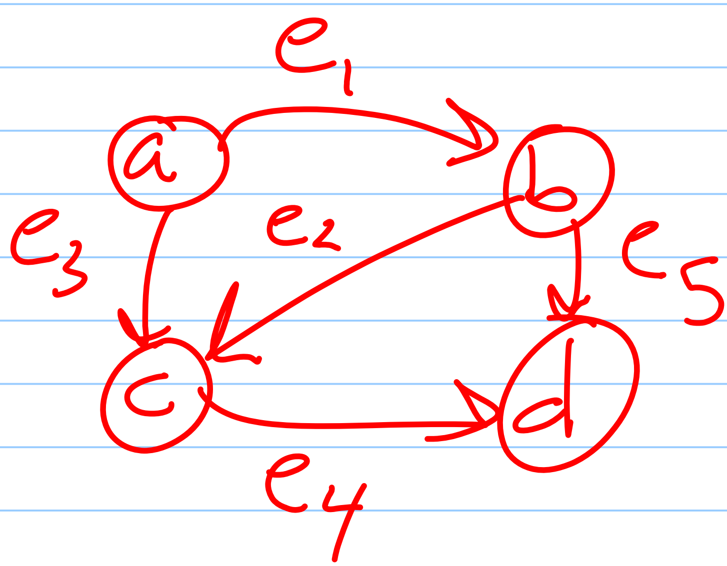
Adjacency Set

Augmented Adjacency Set also keep a set of incoming edges for each vertex

Adjacency Matrix

x null

mapping $\boxed{a \to 0,\ b \to 1,\ c \to 2,\ d \to 3}$



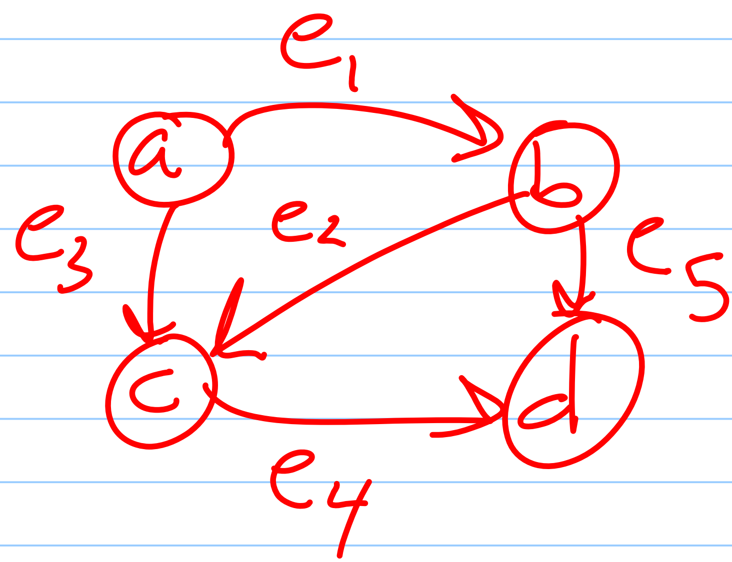
$$
\begin{array}{c|cccc}
 & a & b & c & d \\
\hline
a & \times & e_1 & e_3 & \times \\
b & \times & \times & e_2 & e_5 \\
c & \times & \times & \times & e_4 \\
d & \times & \times & \times & \times
\end{array}
$$

$O(n^2)$ time to iterate over all edges (if not multigraph)

n vertices

m edges

$n \times n$ matrix



adj list

$a \xrightarrow{1} \boxed{e_1} \xrightarrow{2} \boxed{e_3 \ \text{null}}$ 3

$b \xrightarrow{4} \boxed{e_2} \xrightarrow{5} \boxed{e_5 \ \text{null}}$ 6

$c \xrightarrow{7} \boxed{e_4 \ \text{null}}$ 8

d : null 9

$\underbrace{\qquad\qquad\qquad}$

objects among all lists is m

$O(m+n)$ time
to iterate over all edges

Breadth-First Search

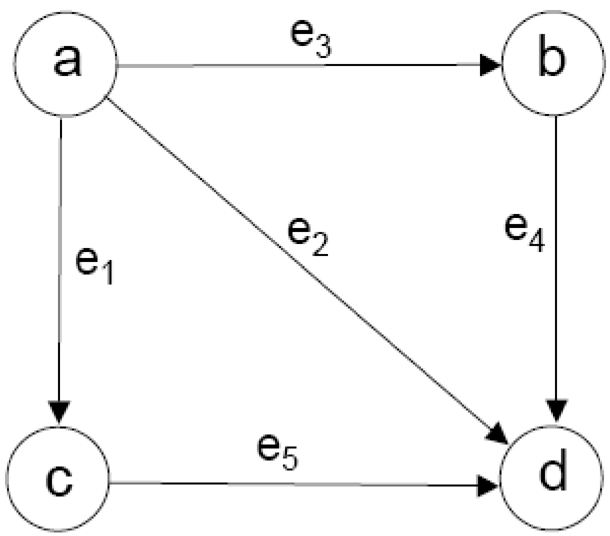
Note Title 11/15/2007

First we'll overview the graph representations.

Then we'll look at problem of finding a shortest path in a directed unweighted graph

Adjacency List

vertices $\{a, b, c, d\}$



outedges

$a \to \{e_3, e_1, e_2\}$

$b \to \{e_4\}$

$c \to \{e_5\}$

$d \to \{\ \}$

inedges

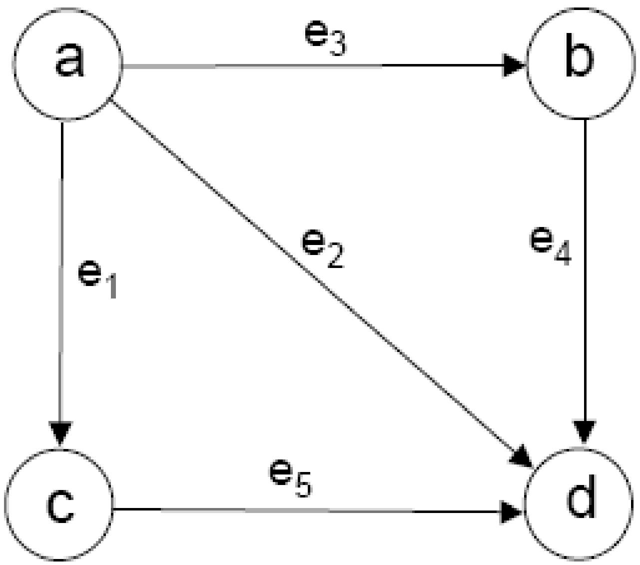
$a \to \{\}$ $b \to \{e_3\}$

$c \to \{e_1\}$ $d \to \{e_2, e_4, e_5\}$

we've called Augmented Adj List

Data Structure	*storeIncomingEdges*	TaggedBucketCollection type
AdjacencyList	*false*	$V \rightarrow$ List<E>
AugmentedAdjacencyList	*true*	$V \rightarrow$ List<E>
Adjacency Set (no multi-edges)	*false*	$V \rightarrow$ Set<E>
Augmented Adjacency Set (no multi-edges)	*true*	$V \rightarrow$ Set<E>
Adjacency Set (with multi-edges)	*false*	$V \rightarrow$ BucketMapping <V, List <E>>
Augmented Adjacency Set (with multi-edges)	*true*	$V \rightarrow$ BucketMapping <V, List <E>>

Adjacency matrix



<u>ids</u>

$a \rightarrow 0$

$c \rightarrow 2$

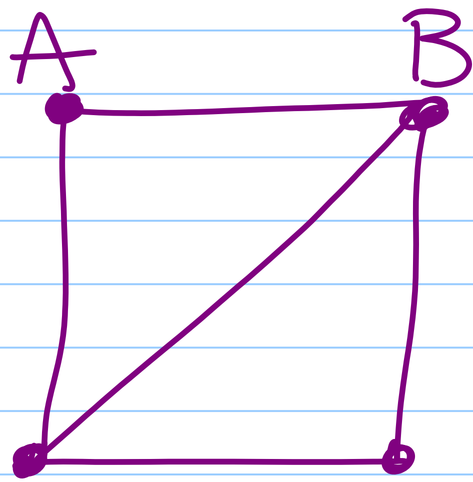
$d \rightarrow 3$

$b \rightarrow 1$

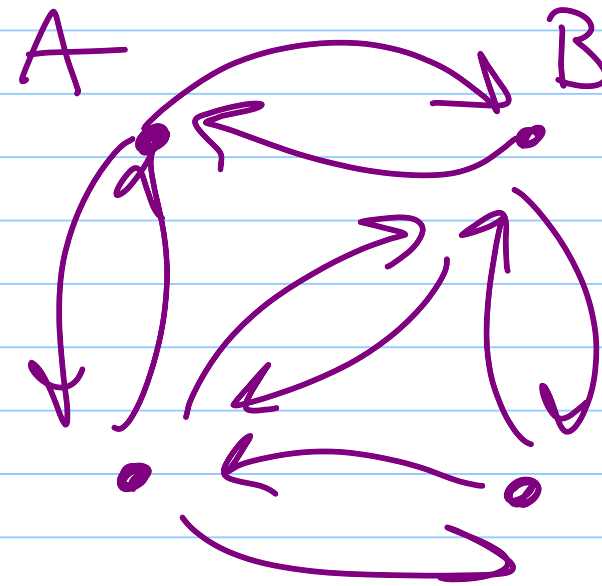
<u>edges</u>

$$
\begin{array}{c|cccc|}
 & 0 & 1 & 2 & 3 \\
\hline
0 & \emptyset & \emptyset & e_1 & e_2 \\
1 & \emptyset & \emptyset & \emptyset & \emptyset \\
2 & \emptyset & \emptyset & \emptyset & e_5 \\
3 & \emptyset & \emptyset & \emptyset & \emptyset \\
\hline
\end{array}
$$

Representing Undirected Graph



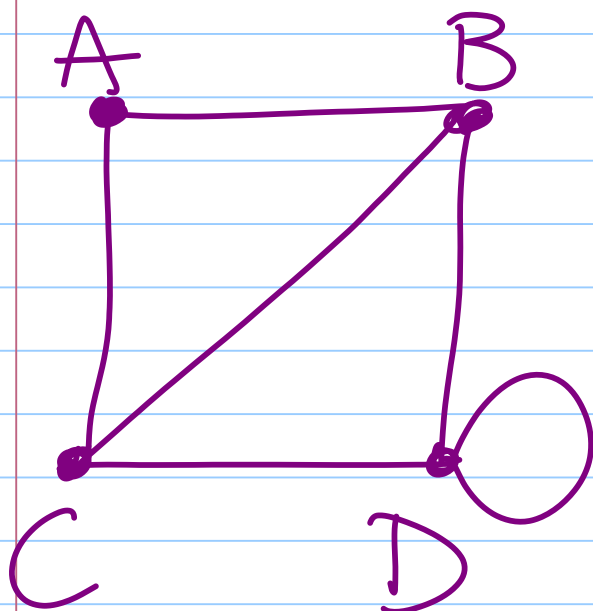
represent as



m edges

2m edges

1 edge   0 no edge



$$
\begin{array}{c|cccc}
 & A & B & C & D \\
\hline
A & 0 & 1 & 1 & 0 \\
B & 1 & 0 & 1 & 1 \\
C & 1 & 1 & 0 & 1 \\
D & 0 & 1 & 1 & 1
\end{array}
$$

Undirected graph, only need to store this part

Analysis (directed graph, no multi-edges)

n vertices
m edges

	Adj List	Adj Set	Adj Matrix
Contains edge v_i, v_j	$O\left(\begin{matrix}\#\text{ edges}\\ \text{out of } v_i\end{matrix}\right)$	expected $O(1)$	$O(1)$
iterate over out edges v	$O\left(\begin{matrix}\#\text{ edges}\\ \text{out of } v\end{matrix}\right)$	$O\left(\begin{matrix}\#\text{ edges}\\ \text{out of } v\end{matrix}\right)$	$O(n)$
iterate over all edges	$O(n+m)$	$O(n+m)$	$O(n^2)$
space	$O(n+m)$	$O(n+m)$	$O(n^2)$