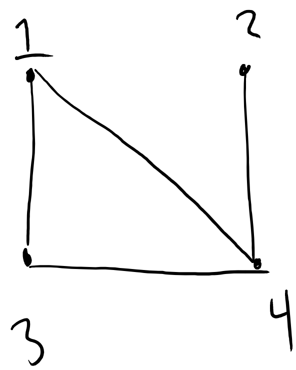
S. KIMMEL

Ways to Represent Graphs in Computer

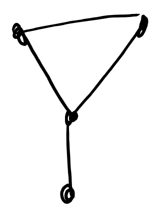
Adjacency Matrix

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 0 & 1 & 1 \\ 2 & 0 & 0 & 0 & 1 \\ 3 & 1 & 0 & 0 & 1 \\ 4 & 1 & 1 & 1 & 0 \end{array}$$



Store as array A in memory. Can learn A[i,j] in $O(1)$ time.

Which adjacency matrix represents this graph?



$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

A

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

B

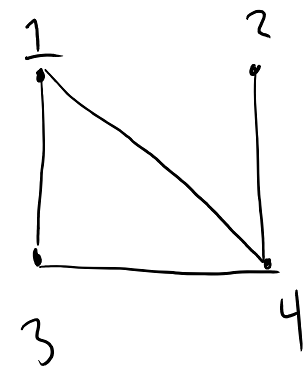
$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

C

S. KIMMEL

Adjacency List

Vertex	Adjacent Vertices
1	3, 4
2	4
3	1, 4
4	1, 2, 3



Store as an array of lists

Can access A[3], (the list) in $O(1)$ time, but then to go through list takes time $O(L)$ where L is length of list. Can learn A[3].length in $O(1)$ time.

Edge List

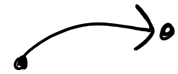
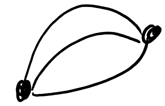
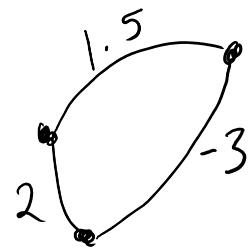
Can represent graph as list of edges, but worstcase time complexity bad for most applications

def: The degree of a vertex is the number of adjacent edges.

def: A vertex v_1 is adjacent to vertex v_2 if connected by an edge

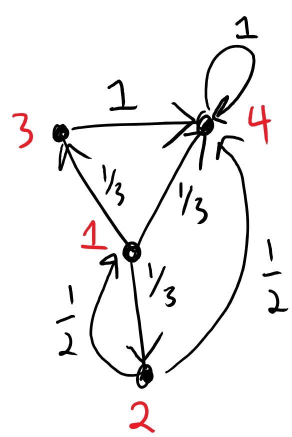
S. KIMMEL

How would you represent a

- directed graph? 
- graph with self-loops? 
- graph with multi edges? 
- graph with weighted edges? 

Using Adjacency Matrix / Adjacency List?

Give representations of this graph using both approaches:



	1	2	3	4
1	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
2	$\frac{1}{2}$	0	0	$\frac{1}{2}$
3	0	0	0	1
4	0	0	0	1

V	List
1	(2, 1/3), (3, 1/3), (4, 1/3)
2	(1, 1/2), (4, 1/2)
3	(4, 1)
4	(4, 1)