ساختمان داده و الگوریتم ها (CE203)

جلسه یازدهم: معرفی گراف و نمایش گراف

سجاد شیرعلی شهرضا

پاییز 1401

دوشنبه، 16 آبان 1401

Based on Stanford CS161 Slides from Summer 2021 by Karey Shi

اطلاع رسانی

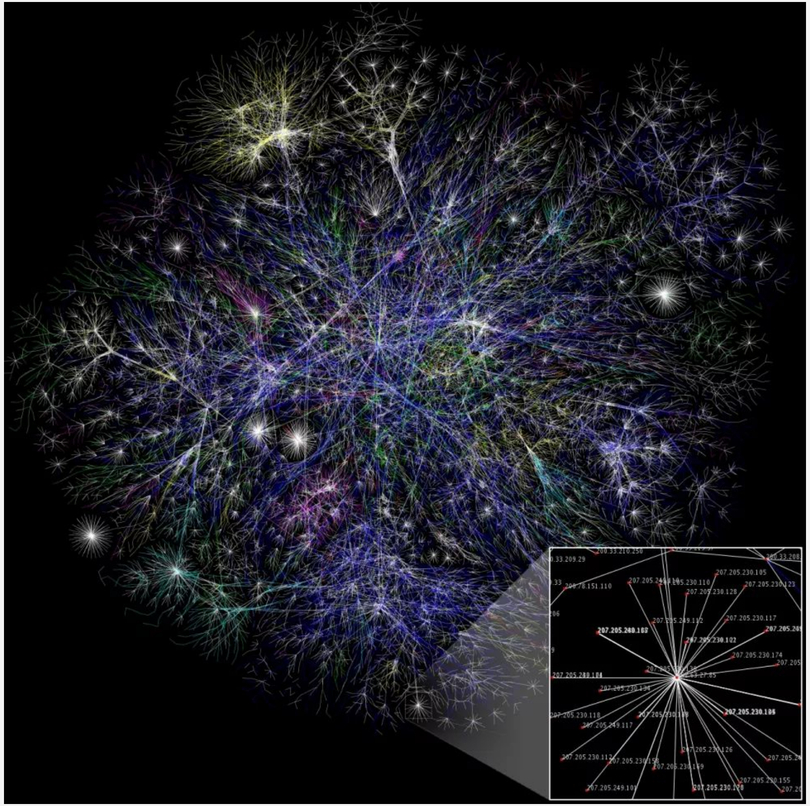
- بخش مرتبط کتاب برای این جلسه: 22

گراف

تعریف و نمونه

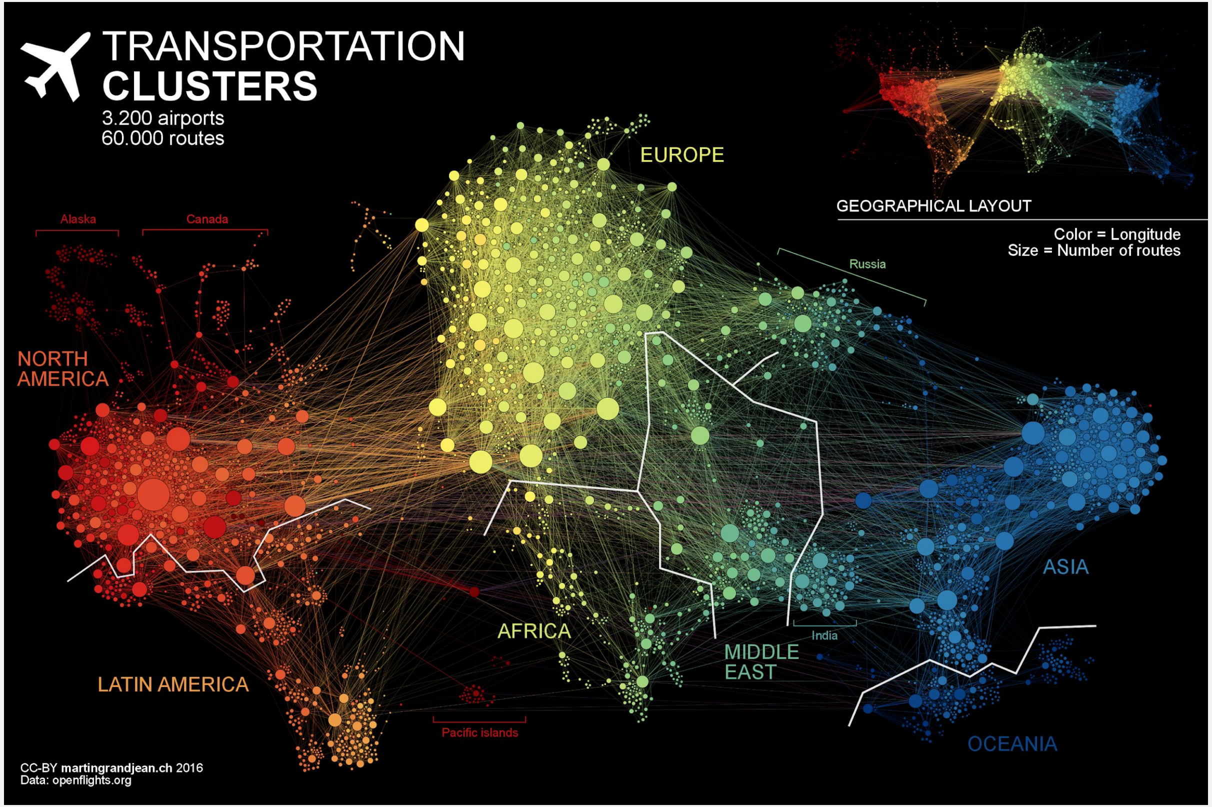
GRAPH EXAMPLES

Partial graph of the Internet (in 2005), where each “node” is an IP address, and the “edges” between them reveal connectivity delays (shorter lines = closer IP addresses)



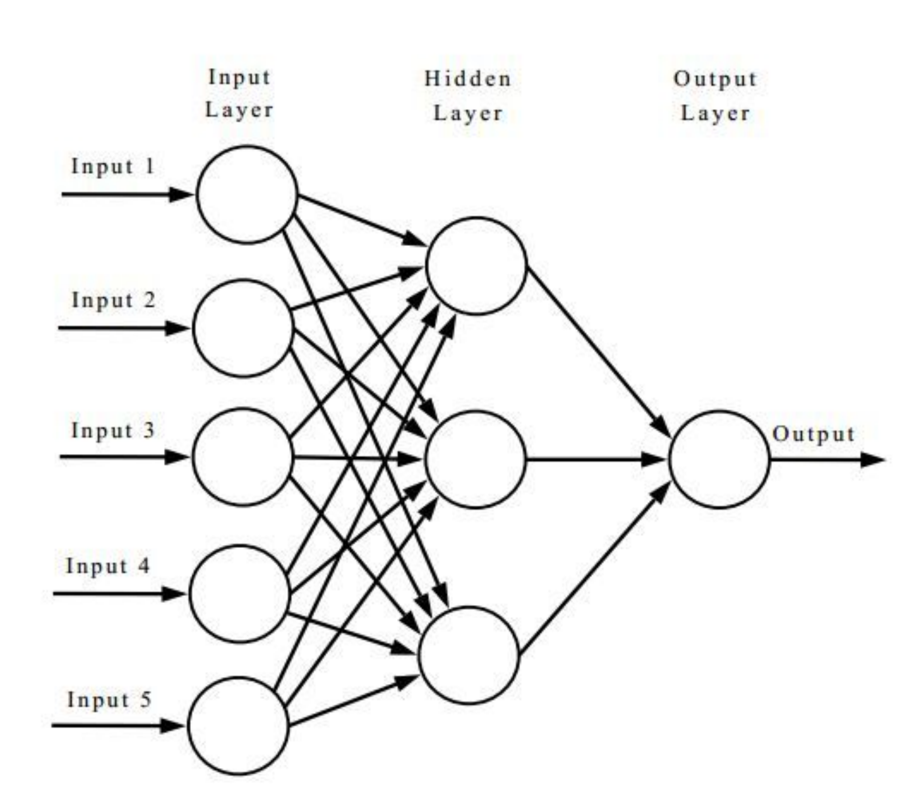
GRAPH EXAMPLES

Each “node” is an airport, and flight routes are represented by the “edge” in between them



GRAPH EXAMPLES

Neural networks! Each "node" represents a module of the neural network, and "edge" represent output/input relationships



GRAPH EXAMPLES

Graph of characters in the third book of Game of Thrones, where each “node” is a character, and “edge” reveal frequency of interaction (i.e. 2 names appearing within 15 words of one another).

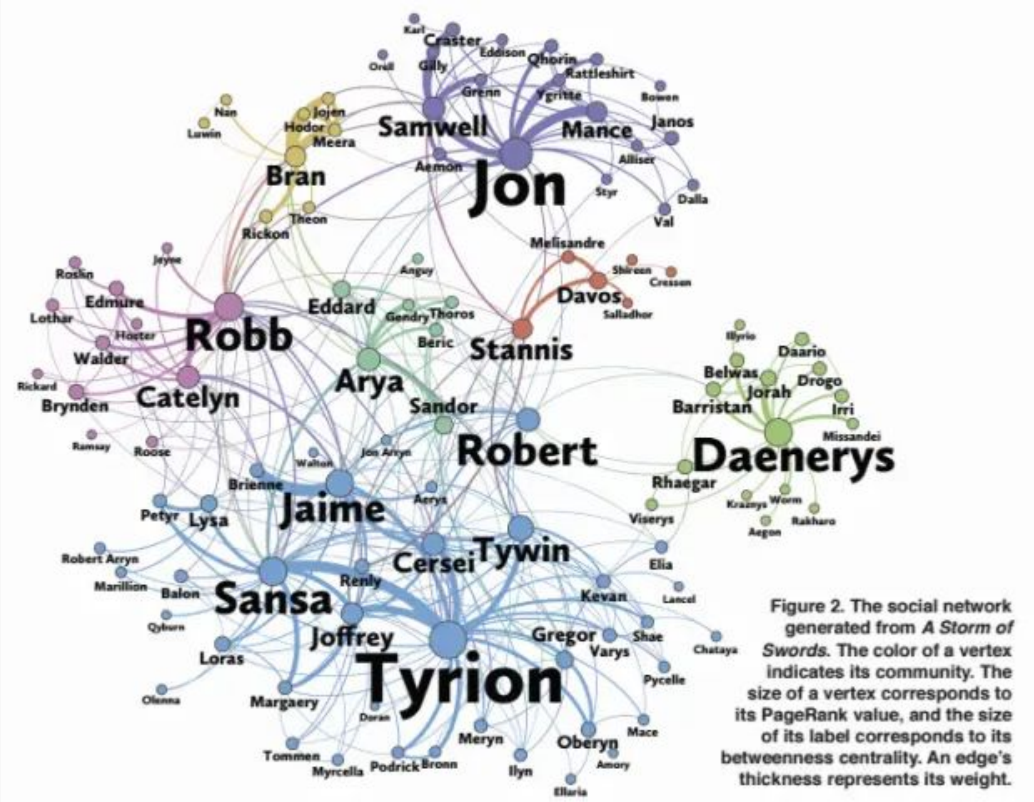
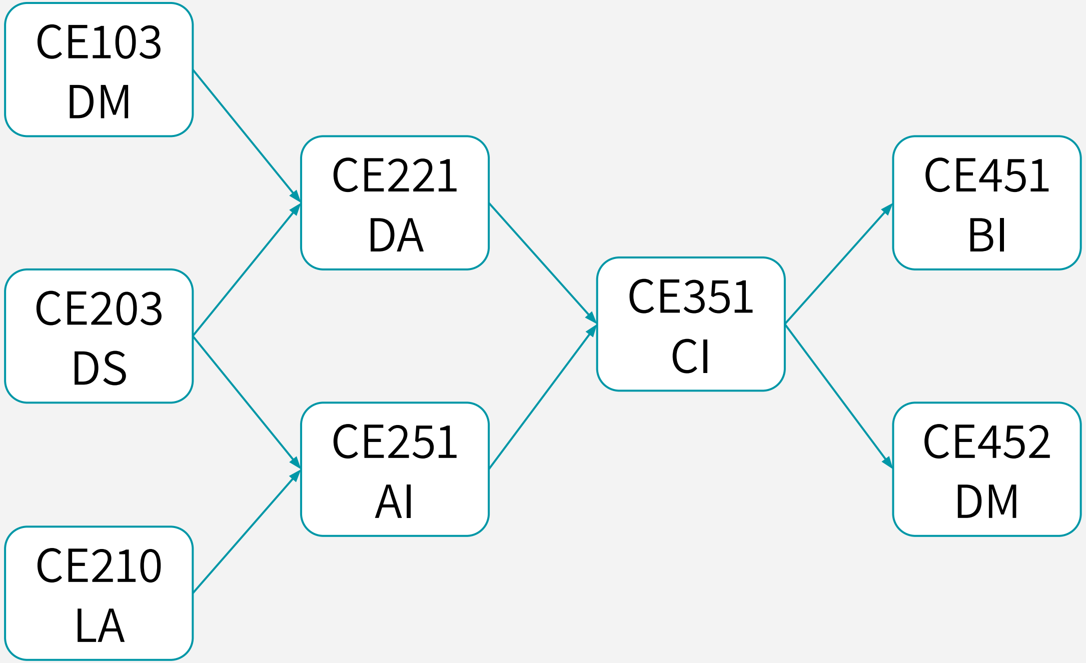


Figure 2. The social network generated from *A Storm of Swords*. The color of a vertex indicates its community. The size of a vertex corresponds to its PageRank value, and the size of its label corresponds to its betweenness centrality. An edge's thickness represents its weight.

GRAPH EXAMPLES

CE prerequisites!
“nodes” are classes
and an “edge” from
class A to class B
means “class B
depends on class A”



WHAT ARE GRAPHS USED FOR?

- There are a lot of diverse problems that can be represented as graphs, and we want to answer questions about them
- For example:
  - How do we most efficiently route packets across the internet?
  - Are there natural “clusters” or “communities” in a graph?
  - Which character(s) are least related with ______?
  - How should I sign up for classes without violating pre-req constraints?

But first off, some terminology!

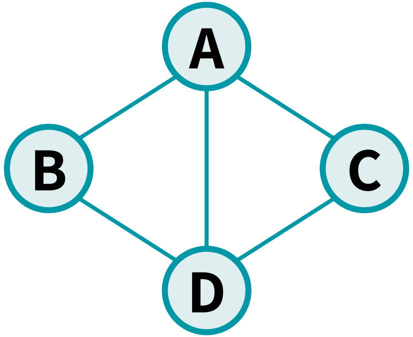
2 KINDS OF GRAPHS

We'll deal with both kinds of graphs in this class.

UNDIRECTED GRAPHS

An undirected graph has
a set of vertices (V) & a set of edges (E)

Formally,
G = (V, E)



V = {A, B, C, D}
E = { {A, B}, {A, C}, {A, D}, {B, D}, {C, D}}

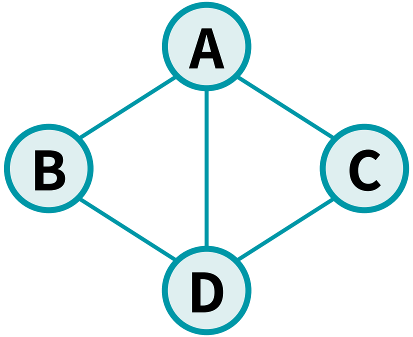
2 KINDS OF GRAPHS

We'll deal with both kinds of graphs in this class.

UNDIRECTED GRAPHS

An undirected graph has
a set of vertices (V) & a set of edges (E)

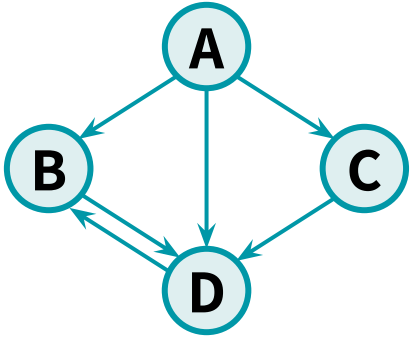
Formally,
G = (V, E)



V = {A, B, C, D}
E = { {A, B}, {A, C}, {A, D}, {B, D}, {C, D}}

DIRECTED GRAPHS

A directed graph has
a set of vertices (V) & a set of **DIRECTED** edges (E)



Formally,
G = (V, E)

V = {A, B, C, D}
E = { [A, B], [A, C], [A, D], [B, D], [C, D], [D, B]}

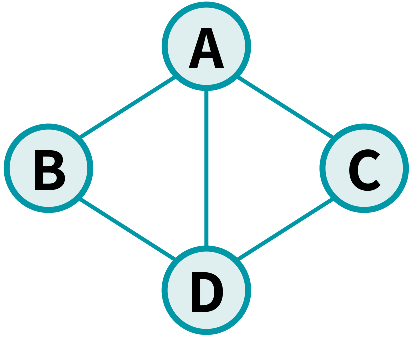
2 KINDS OF GRAPHS

We'll deal with both kinds of graphs in this class.

UNDIRECTED GRAPHS

An undirected graph has
a set of vertices (V) & a set of edges (E)

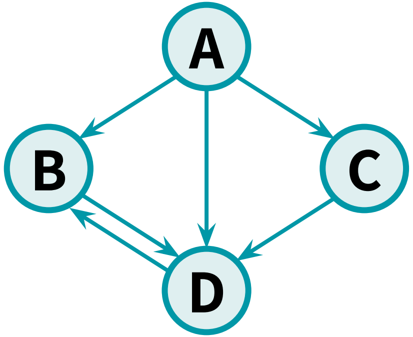
Formally,
G = (V, E)



The **degree** of vertex D is 3
Vertex D's **neighbors** are A, B, and C

DIRECTED GRAPHS

A directed graph has
a set of vertices (V) & a set of **DIRECTED** edges (E)



Formally,
G = (V, E)

The **in-degree** of vertex D is 3. The **out-degree** of vertex D is 1.
Vertex D's **incoming neighbors** are A, B, & C
Vertex D's **outgoing neighbor** is B

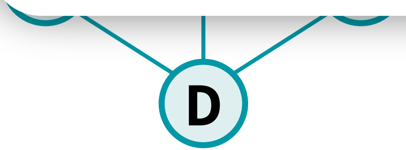
2 KINDS OF GRAPHS

We'll deal with both kinds of graphs in this class.

UNDIRECTED GRAPHS

a set

Formally,
G = (V, E)

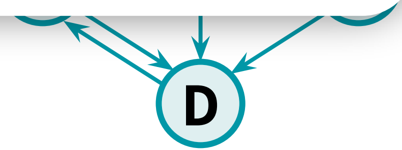


The **degree** of vertex D is 3
Vertex D's **neighbors** are A, B, and C

DIRECTED GRAPHS

edges (E)

Formally,
G = (V, E)

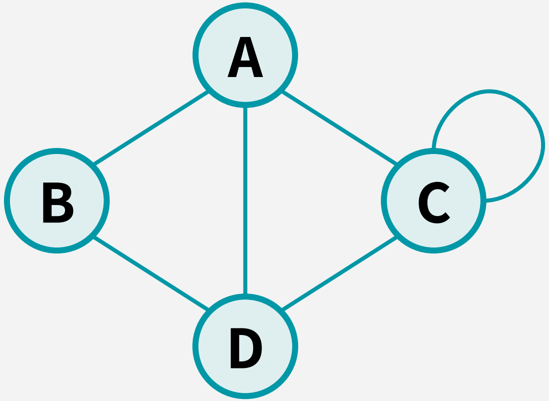


The **in-degree** of vertex D is 3. The **out-degree** of vertex D is 1.
Vertex D's **incoming neighbors** are A, B, & C
Vertex D's **outgoing neighbor** is B

Today, we're only working with ***unweighted*** graphs. These are graphs where edges aren't assigned weights, or all edges are assumed to have the same weight.

GRAPH REPRESENTATIONS

OPTION 1: **ADJACENCY MATRIX**



(An undirected graph)

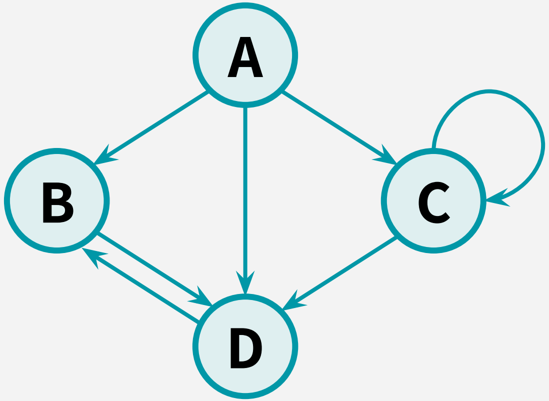
(destination)

(source)

$$\begin{array}{cc} & \begin{array}{cccc} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} \end{array} \\ \begin{array}{c} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \\ \mathbf{D} \end{array} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

GRAPH REPRESENTATIONS

OPTION 1: **ADJACENCY MATRIX**



(A directed graph)

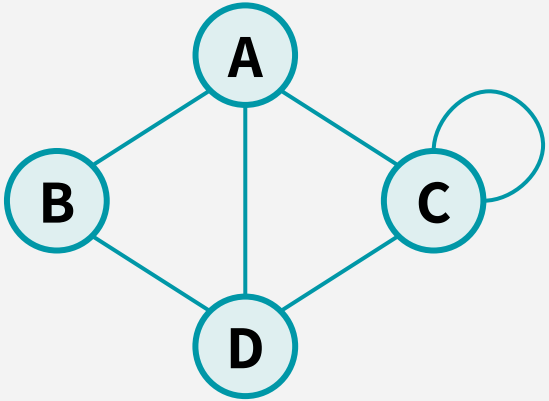
(destination)

(source)

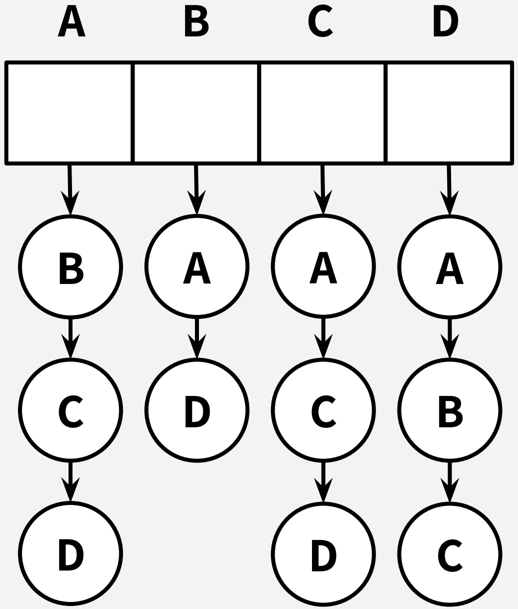
$$
\begin{array}{c c}
 & \begin{array}{cccc} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} \end{array} \\
\begin{array}{c} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \\ \mathbf{D} \end{array} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}
\end{array}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

GRAPH REPRESENTATIONS

OPTION 2: **ADJACENCY LISTS**



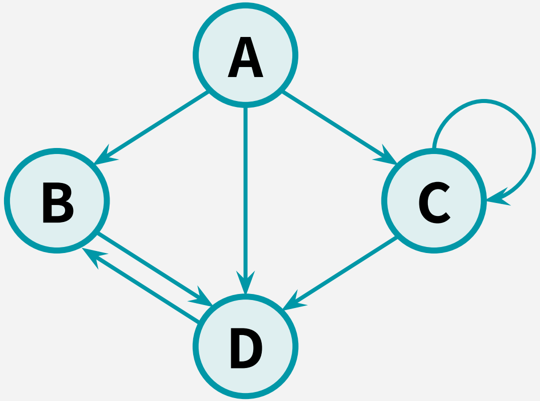
(An undirected graph)



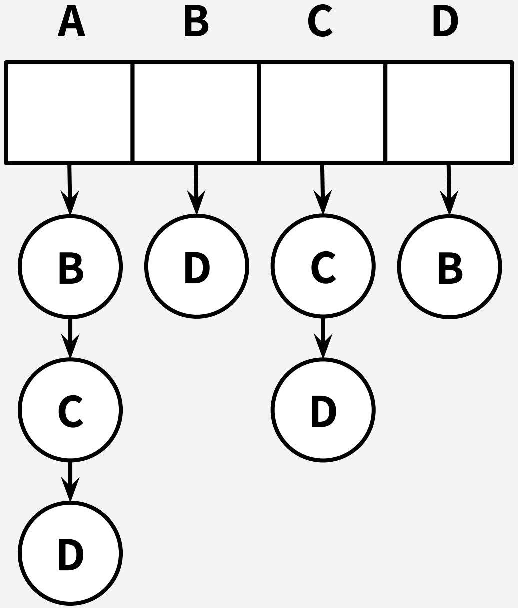
Each list stores a node's neighbors

GRAPH REPRESENTATIONS

OPTION 2: **ADJACENCY LISTS**



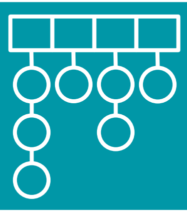
(A directed graph)



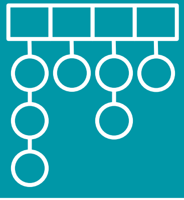
Tracks *outgoing neighbors.*

(You could also do the same for incoming neighbors as well)

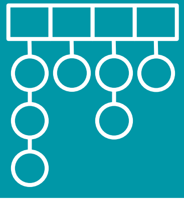
GRAPH REPRESENTATIONS

For a graph G = (V, E) where |V| = **n**, and |E| = **m**	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	
EDGE MEMBERSHIP Is e = {v, w} in E?		
NEIGHBOR QUERY Give me v's neighbors		
SPACE REQUIREMENTS		

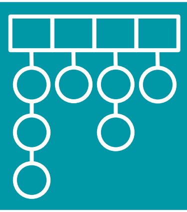
GRAPH REPRESENTATIONS

For a graph G = (V, E) where $\lvert V \rvert$ = **n**, and $\lvert E \rvert$ = **m**	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	
EDGE MEMBERSHIP Is e = {v, w} in E?	**O(1)**	
NEIGHBOR QUERY Give me v's neighbors		
SPACE REQUIREMENTS		

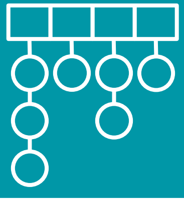
GRAPH REPRESENTATIONS

For a graph G = (V, E) where $\lvert V \rvert = \mathbf{n}$, and $\lvert E \rvert = \mathbf{m}$	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	
EDGE MEMBERSHIP Is e = {v, w} in E?	**O(1)**	**O(deg(v))** or **O(deg(w))**
NEIGHBOR QUERY Give me v's neighbors		
SPACE REQUIREMENTS		

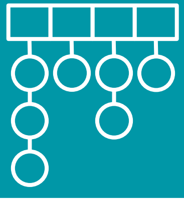
GRAPH REPRESENTATIONS

For a graph G = (V, E) where |V| = **n**, and |E| = **m**	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	
EDGE MEMBERSHIP Is e = {v, w} in E?	**O(1)**	**O(deg(v))** or **O(deg(w))**
NEIGHBOR QUERY Give me v's neighbors	**O(n)**	
SPACE REQUIREMENTS		

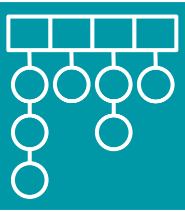
GRAPH REPRESENTATIONS

For a graph $G = (V, E)$ where $\|V\| = \mathbf{n}$, and $\|E\| = \mathbf{m}$	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	
EDGE MEMBERSHIP Is e = {v, w} in E?	**O(1)**	**O(deg(v))** or **O(deg(w))**
NEIGHBOR QUERY Give me v's neighbors	**O(n)**	**O(deg(v))**
SPACE REQUIREMENTS		

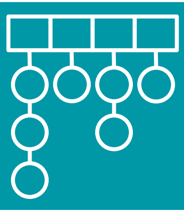
GRAPH REPRESENTATIONS

For a graph G = (V, E) where $|V| = \mathbf{n}$, and $|E| = \mathbf{m}$	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	
EDGE MEMBERSHIP Is e = {v, w} in E?	**O(1)**	**O(deg(v))** or **O(deg(w))**
NEIGHBOR QUERY Give me v's neighbors	**O(n)**	**O(deg(v))**
SPACE REQUIREMENTS	**$O(n^2)$**	

GRAPH REPRESENTATIONS

For a graph G = (V, E) where $\lvert V \rvert$ = **n**, and $\lvert E \rvert$ = **m**	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	
EDGE MEMBERSHIP Is e = {v, w} in E?	**O(1)**	**O(deg(v))** or **O(deg(w))**
NEIGHBOR QUERY Give me v's neighbors	**O(n)**	**O(deg(v))**
SPACE REQUIREMENTS	**$O(n^2)$**	**O(n + m)**

GRAPH REPRESENTATIONS

For a graph $G = (V, E)$ where $\lvert V \rvert = \mathbf{n}$, and $\lvert E \rvert = \mathbf{m}$	$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	
EDGE MEMBERSHIP Is $e = \{v, w\}$ in E?	**O(1)**	**O(deg(v))** or **O(deg(w))**
NEIGHBOR QUERY Give me v's neighbors	**O(n)**	**O(deg(v))**
SPACE REQUIREMENTS	**$O(n^2)$**	**O(n + m)**

Generally, better for sparse graphs (where $m \ll n^2$).

We'll assume this representation, unless otherwise stated.

سوال؟