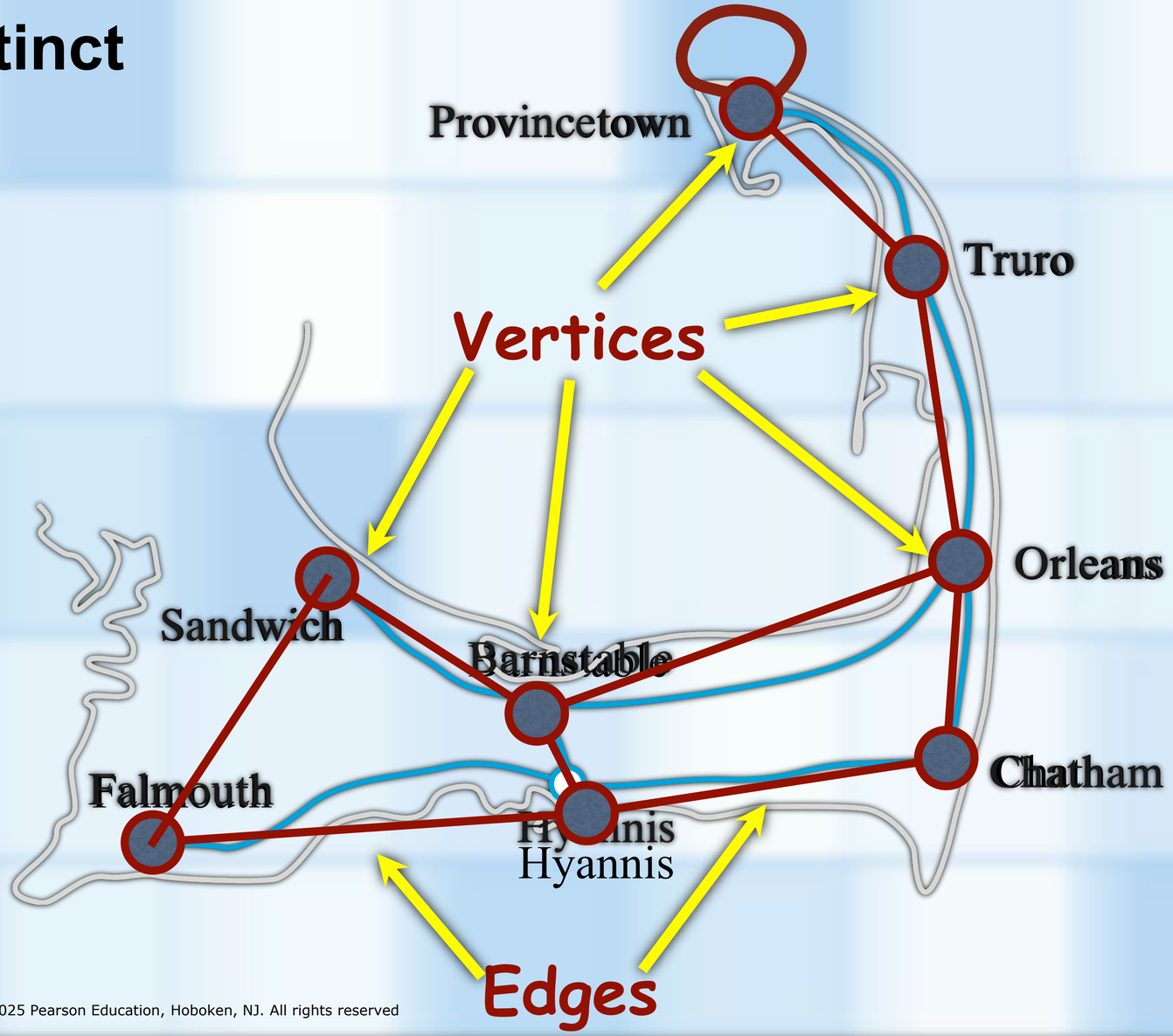
Graph Concepts and Terminology

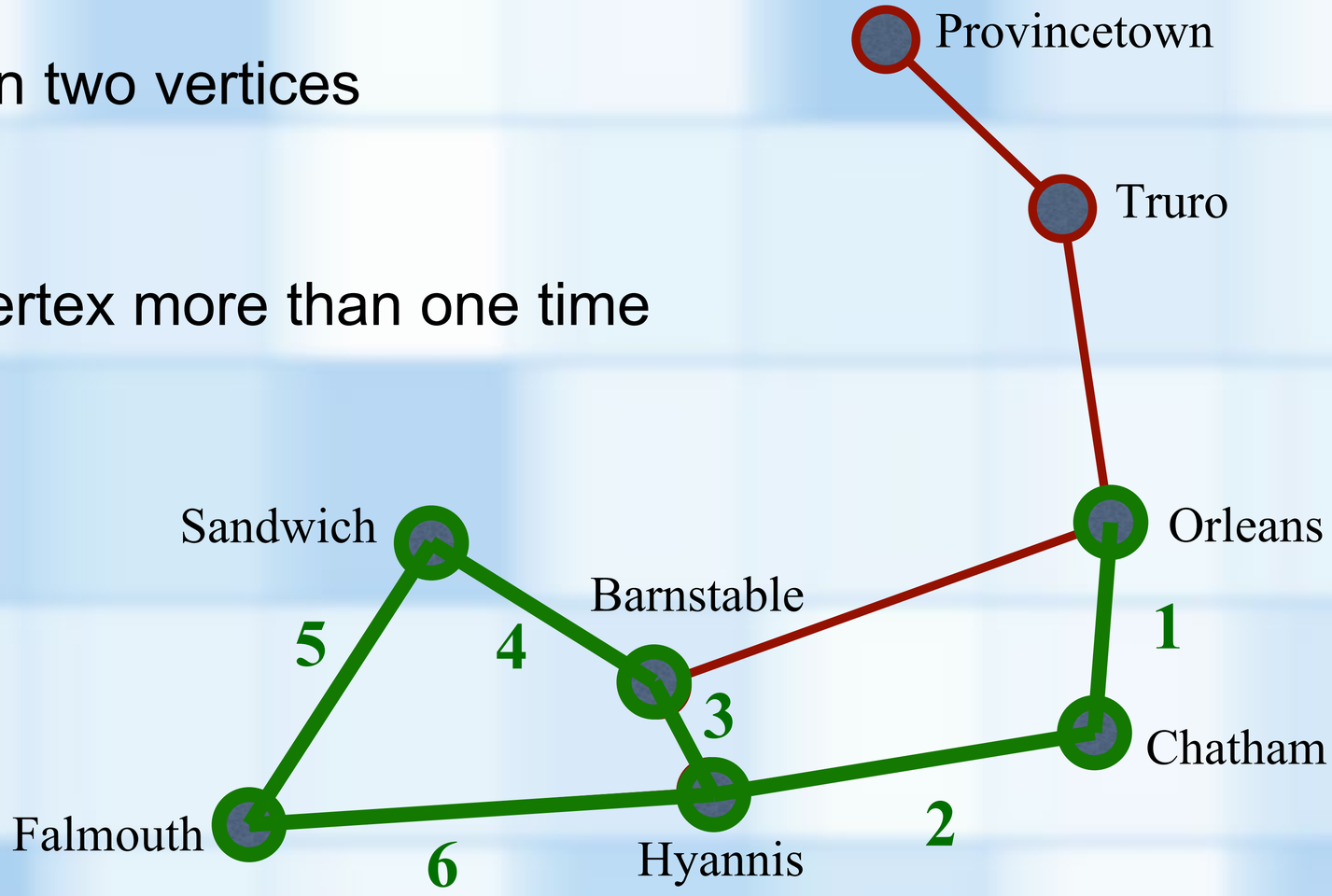
Graph Concepts

- A **Graph** is a collection of distinct **vertices** and distinct **edges**
  - A subgraph is a portion of a graph



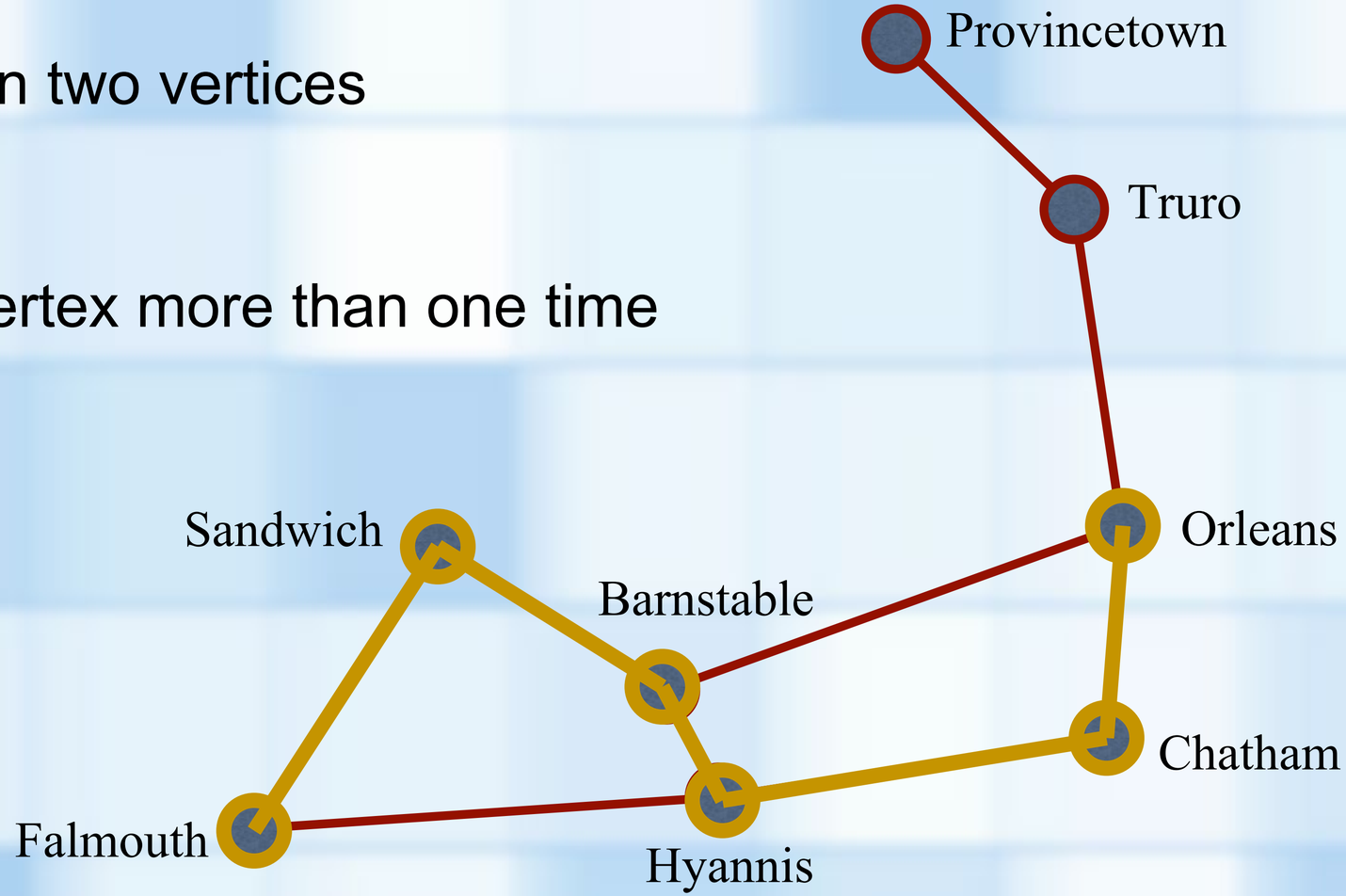
Graph Paths

- **Path**
  - A sequence of edges between two vertices
- **Simple Path**
  - Does not pass through any vertex more than one time



Graph Paths

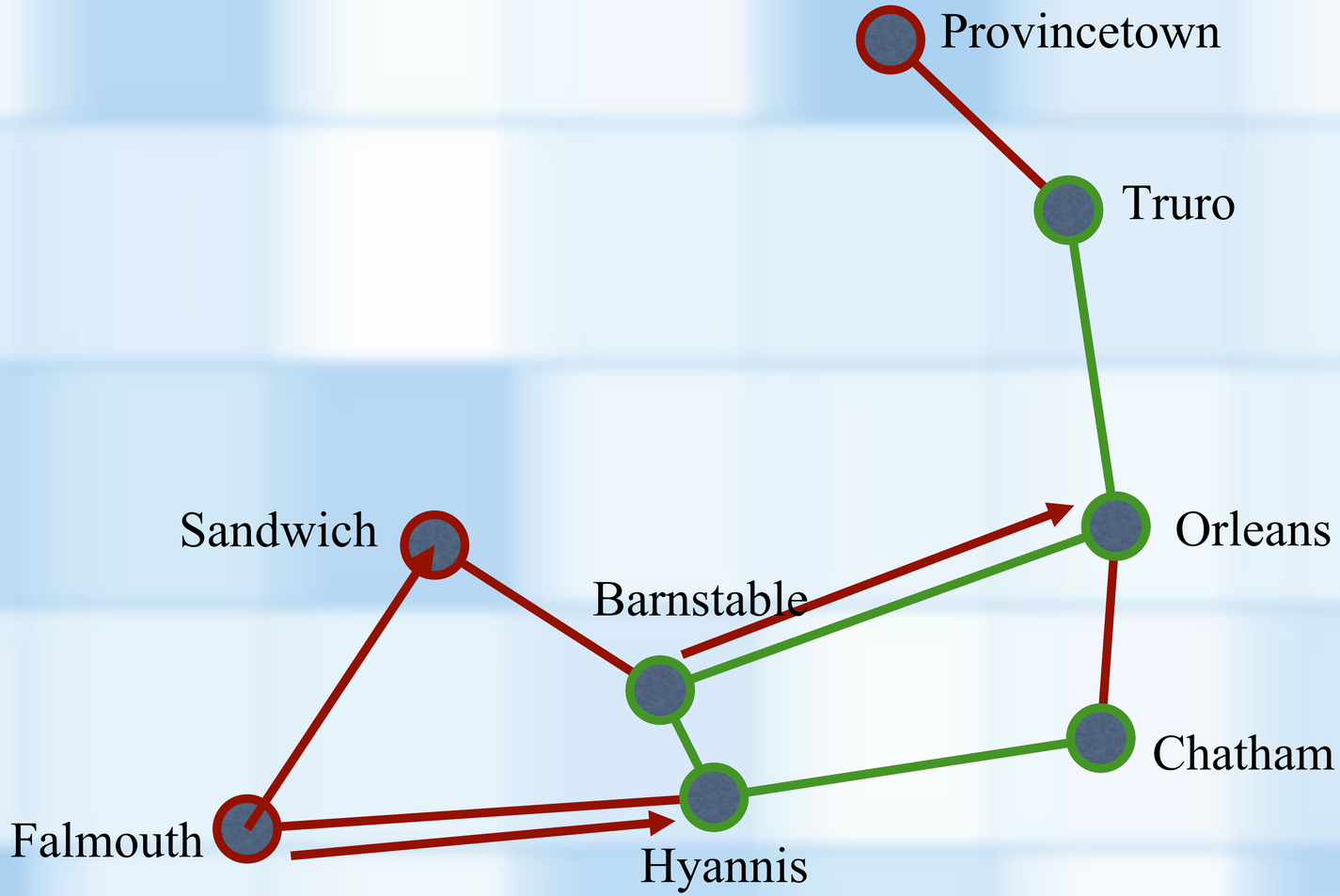
- **Path**
  - A sequence of edges between two vertices
- **Simple Path**
  - Does not pass through any vertex more than one time



Pearson

Directed Graphs

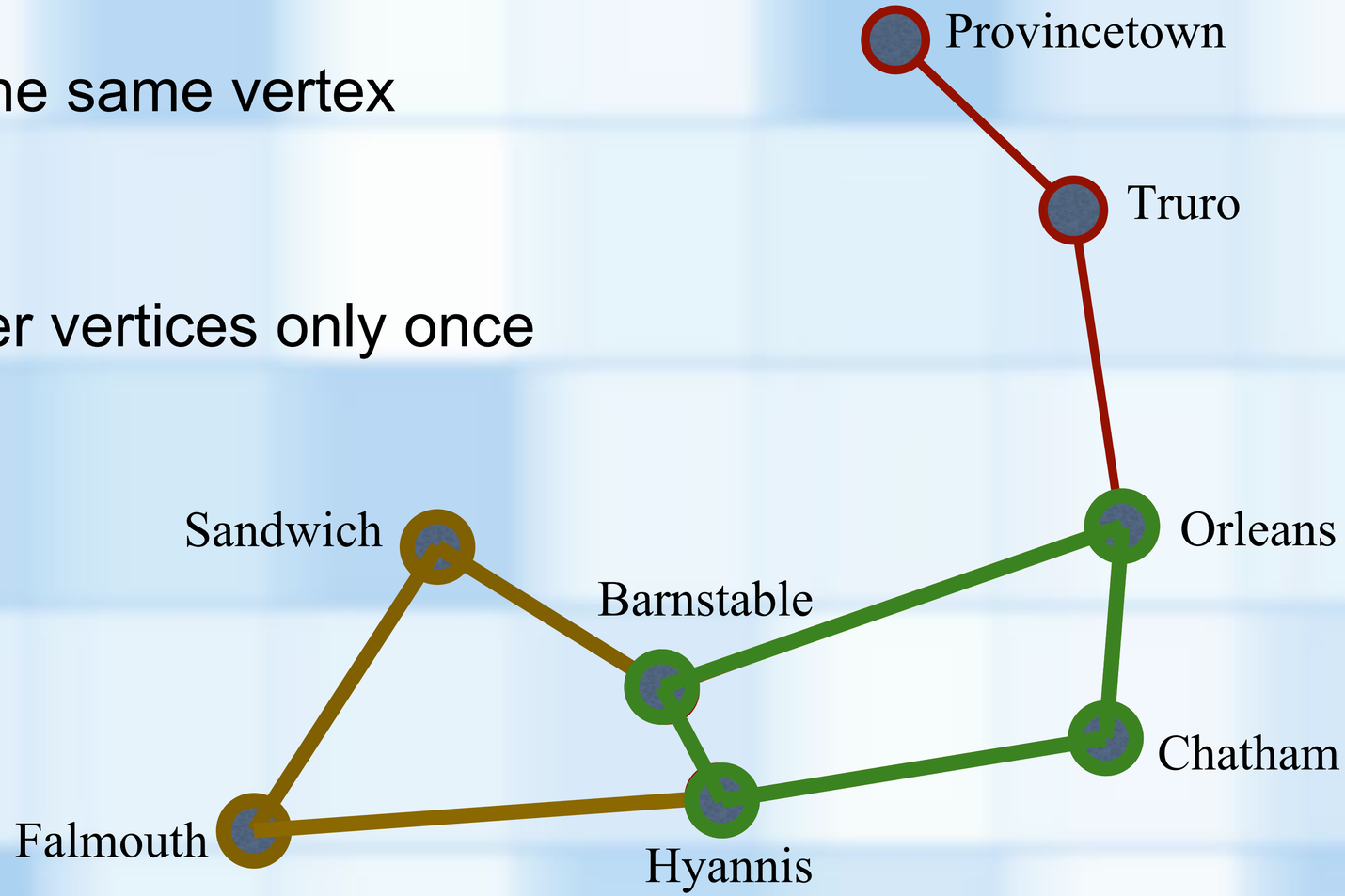
- **Directed Graph (digraph)**
  - Graph with directed edges
- **Directed Path**
  - Path in a directed graph
  - Must consider edge directions



Pearson

Graph Cycles

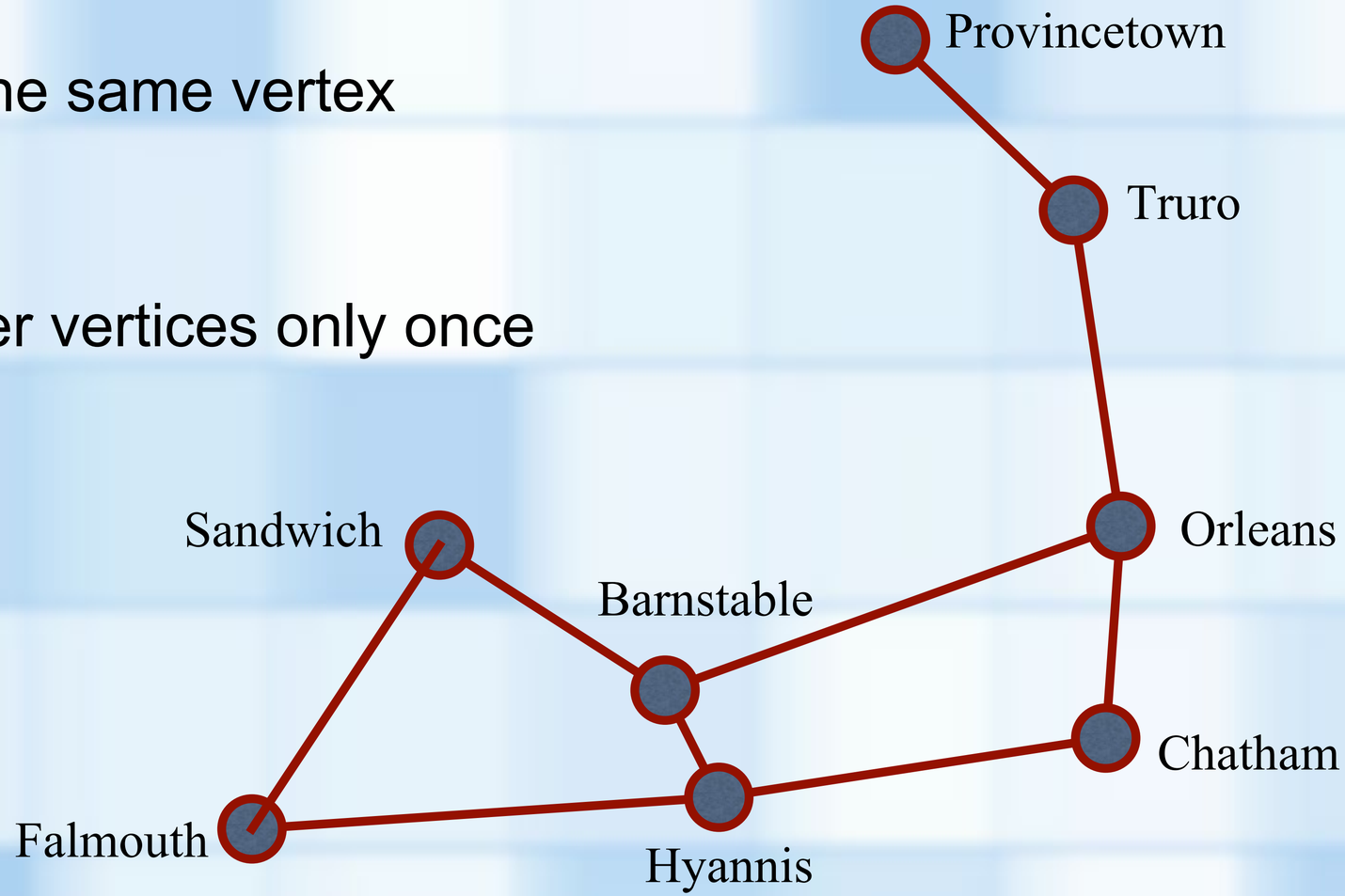
- **Cycle**
  - Path that begins and ends at the same vertex
- **Simple Cycle**
  - Cycle that passes through other vertices only once
- **Acyclic Graph**
  - A graph with no cycles



Pearson

Graph Cycles

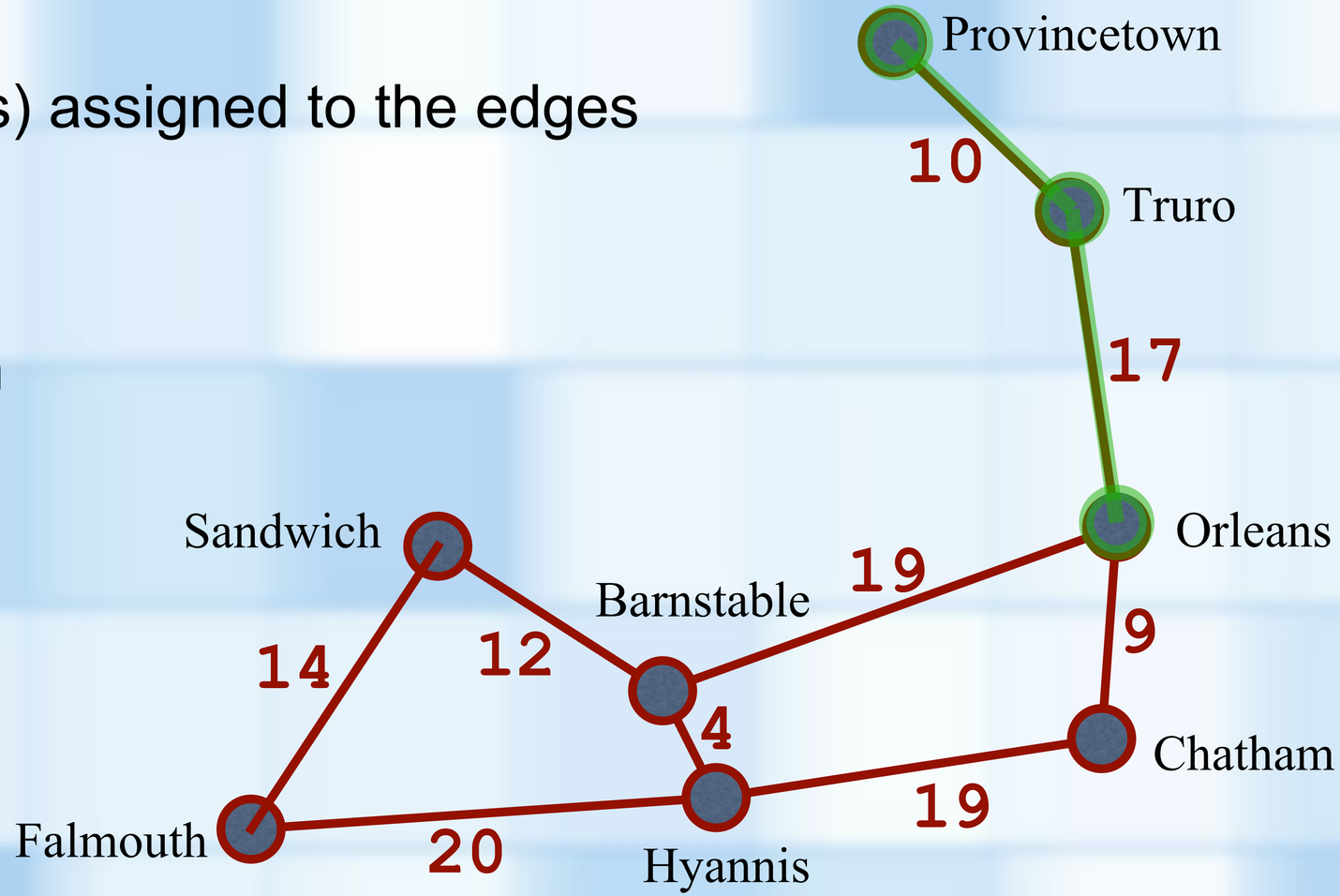
- **Cycle**
  - Path that begins and ends at the same vertex
- **Simple Cycle**
  - Cycle that passes through other vertices only once
- **Acyclic Graph**
  - A graph with no cycles



Pearson

Weighted Graphs

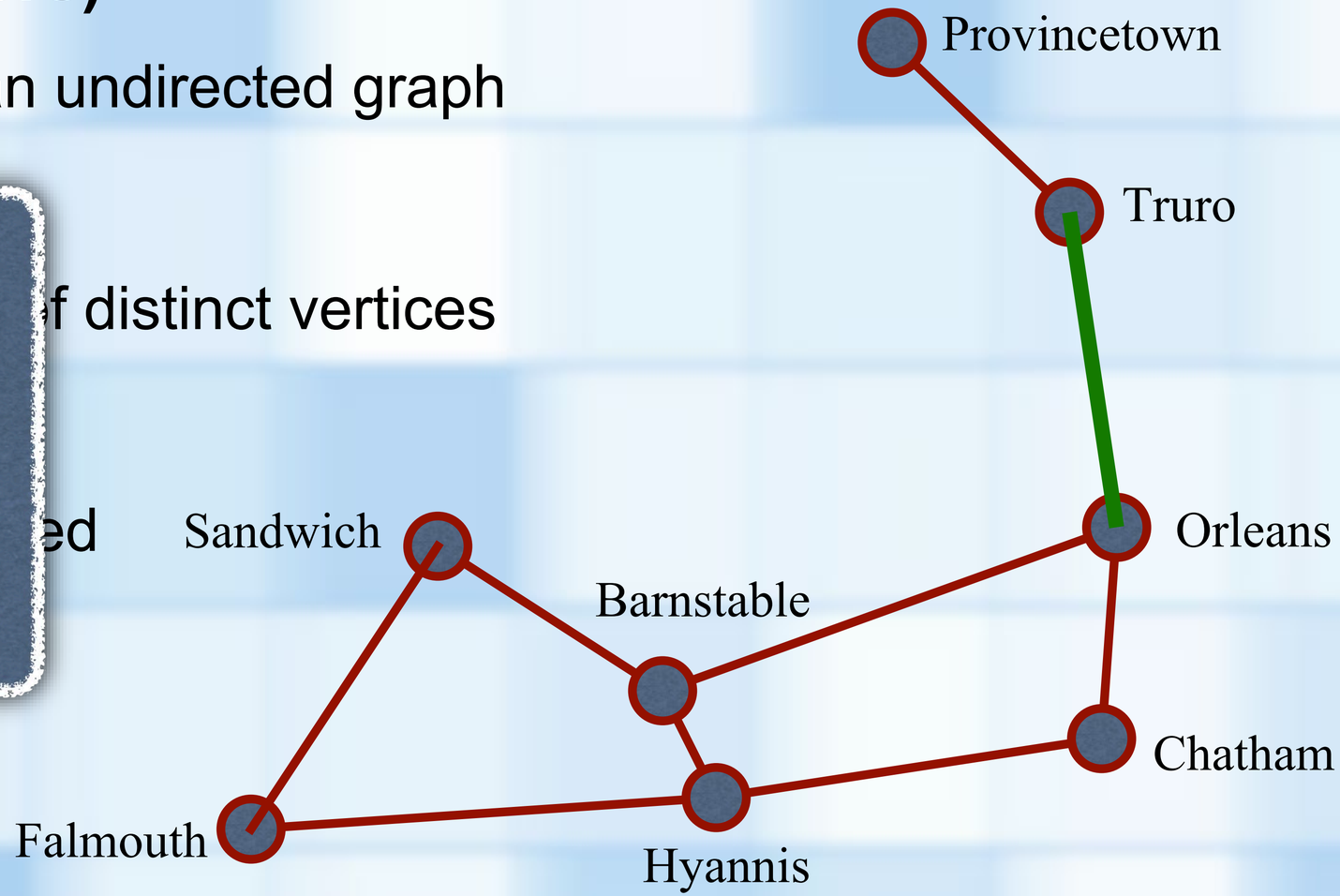
- **Weighted Graph**
  - Graph that has values (weights) assigned to the edges
- **Weighted Path**
  - Path through a weighted graph



Adjacency and Connectedness

- **Adjacent Vertices (neighbors)**
  - Vertices joined by an edge in an undirected graph
- **C**
  - f distinct vertices
- **C**
  - ed

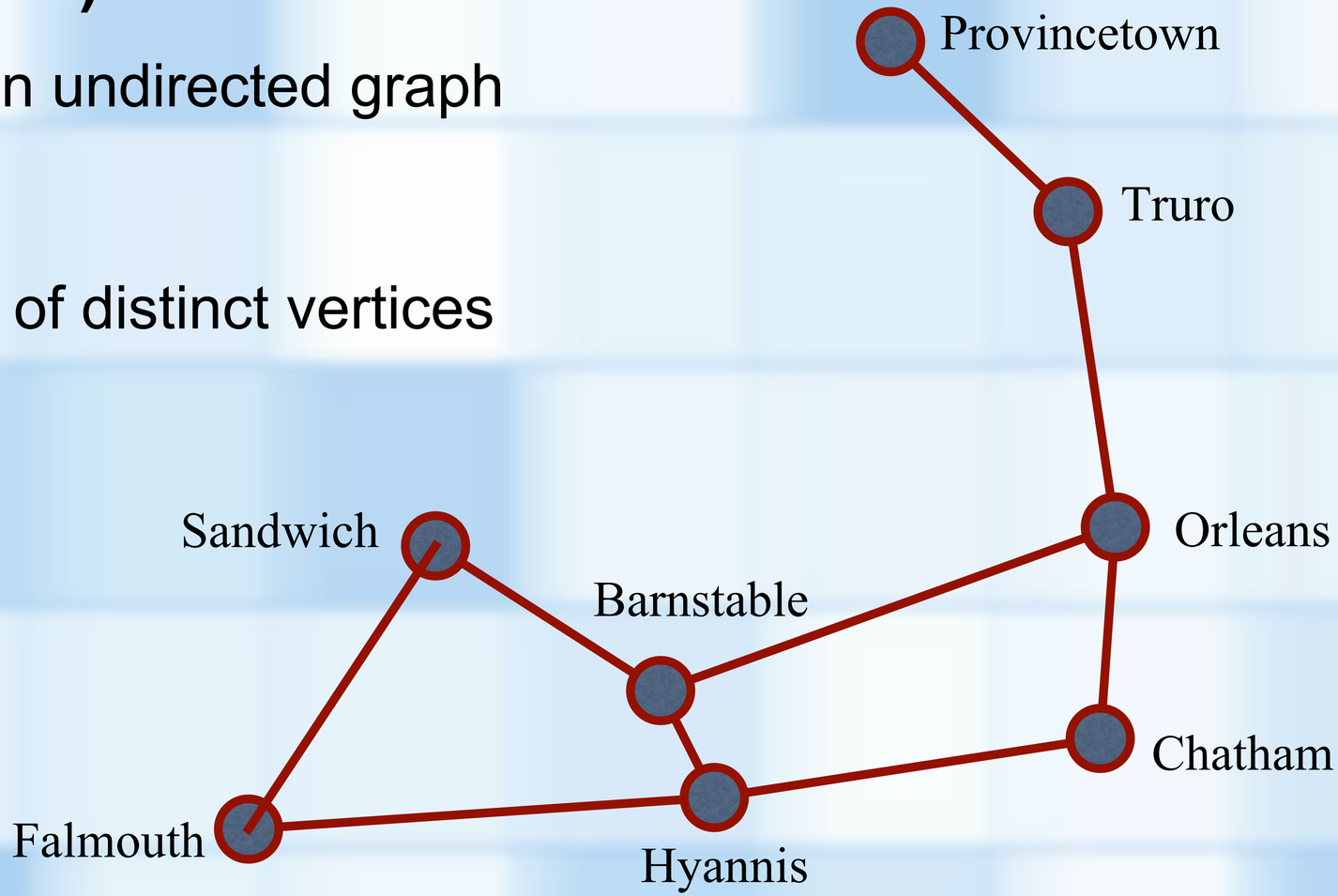
In a directed graph, vertex *i* is adjacent to vertex *j* if a directed edge begins at *j* and ends at *i*.



Pearson

Adjacency and Connectedness

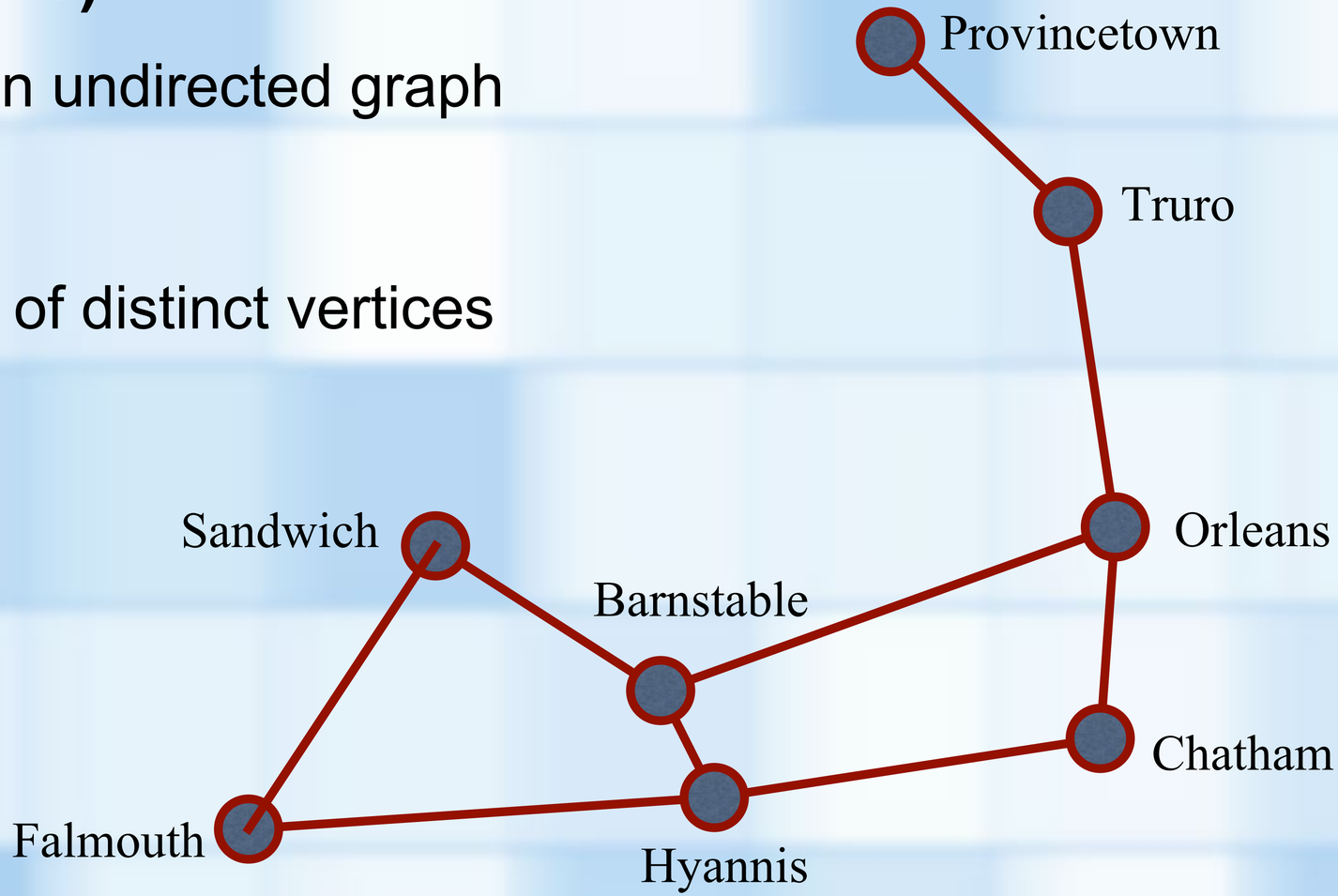
- **Adjacent Vertices (neighbors)**
  - Vertices joined by an edge in an undirected graph
- **Connected Graph**
  - Has a path between every pair of distinct vertices
- **Disconnected Graph**
  - No path from certain vertices to others



Pearson

Adjacency and Connectedness

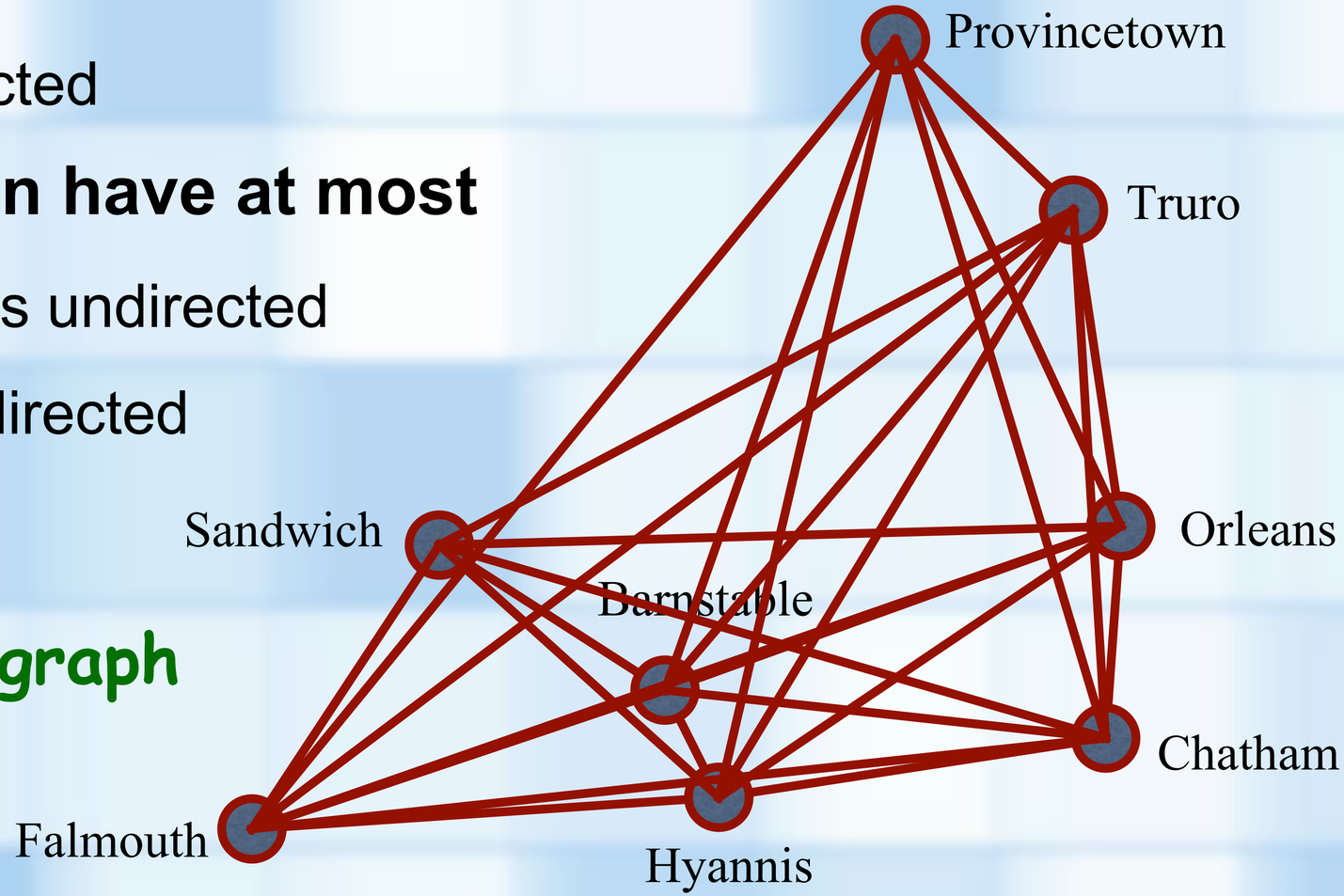
- **Adjacent Vertices (neighbors)**
  - Vertices joined by an edge in an undirected graph
- **Connected Graph**
  - Has a path between every pair of distinct vertices
- **Disconnected Graph**
  - No path from certain vertices to others



Pearson

Adjacency and Connectedness

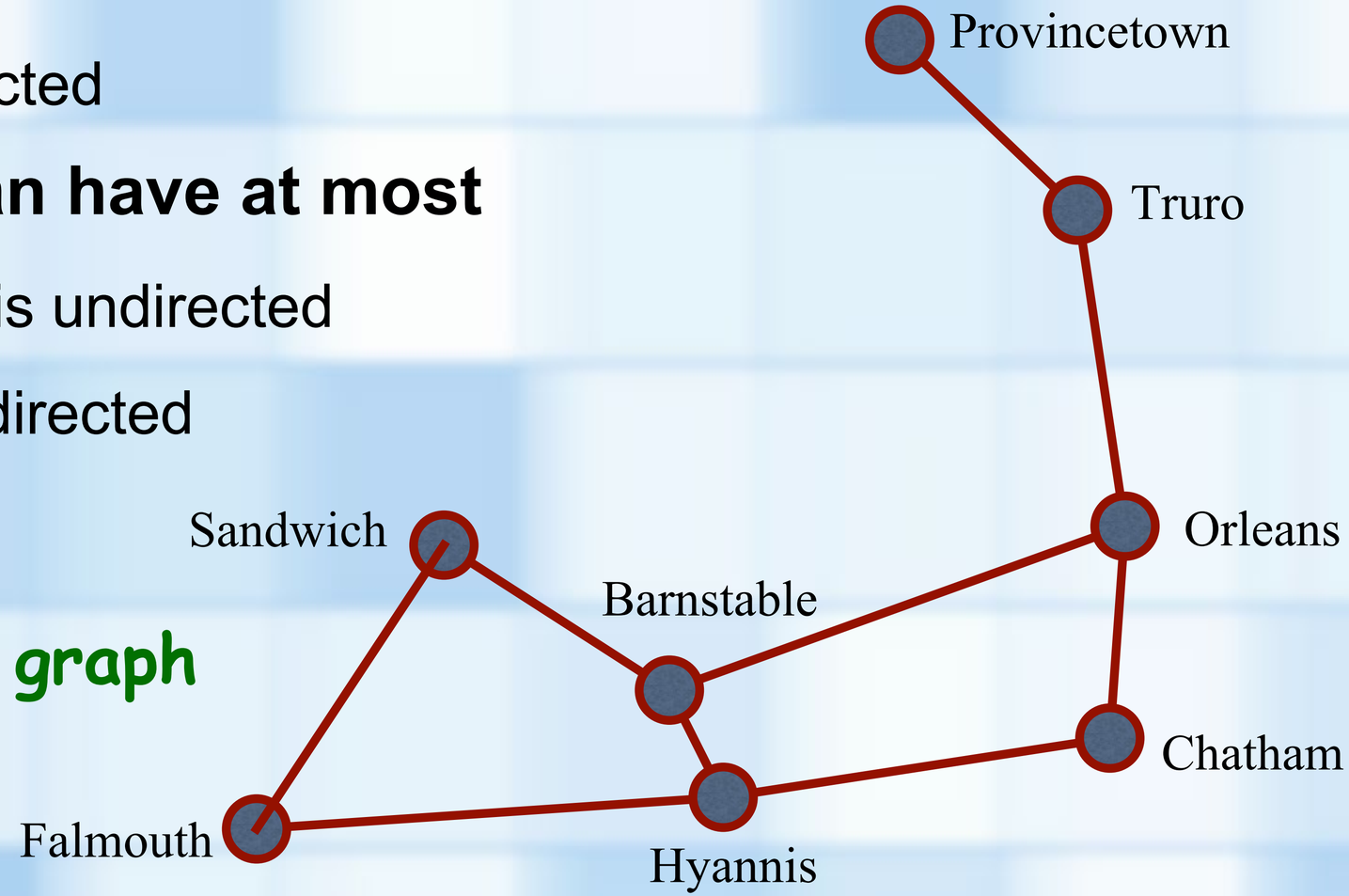
- **Complete Graph**
  - Every pair of vertices is connected
- **A graph with *n* vertices, can have at most**
  - ***n(n-1)/2*** edges if the graph is undirected
  - ***n(n-1)*** edges if the graph is directed



A dense graph

Adjacency and Connectedness

- **Complete Graph**
  - Every pair of vertices is connected
- **A graph with *n* vertices, can have at most**
  - ***n(n-1)/2*** edges if the graph is undirected
  - ***n(n-1)*** edges if the graph is directed



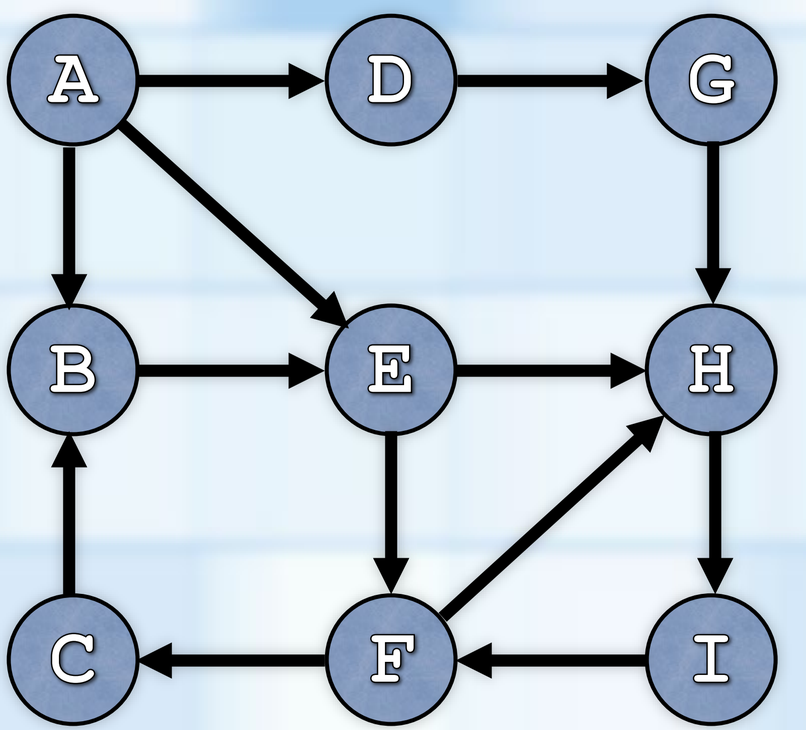
A sparse graph

Pearson

Representing Adjacency

Representing Vertex Relationships

- **Edges connect vertices**
- **Connections represent relationship among vertices**
- **Adjacency Matrix**
  - Two-dimensional array
- **Adjacency List**
  - List of lists



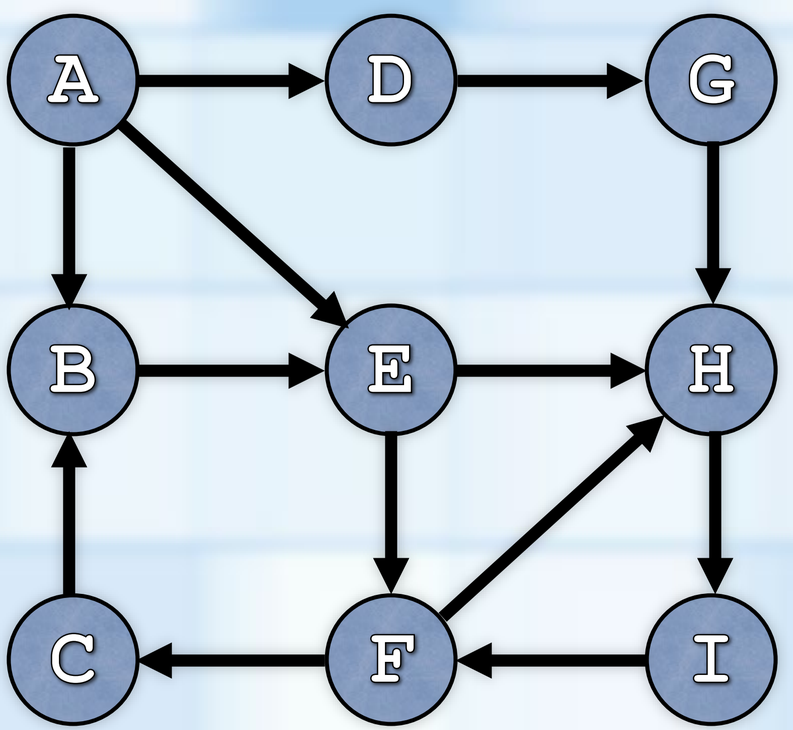
Directed Graph

The Adjacency Matrix

is connected to

Vertex

		0	1	2	3	4	5	6	7	8
		A	B	C	D	E	F	G	H	I
0	A		1		1	1				
1	B					1				
2	C		1							
3	D							1		
4	E						1		1	
5	F			1					1	
6	G								1	
7	H									1
8	I						1			



Directed Graph

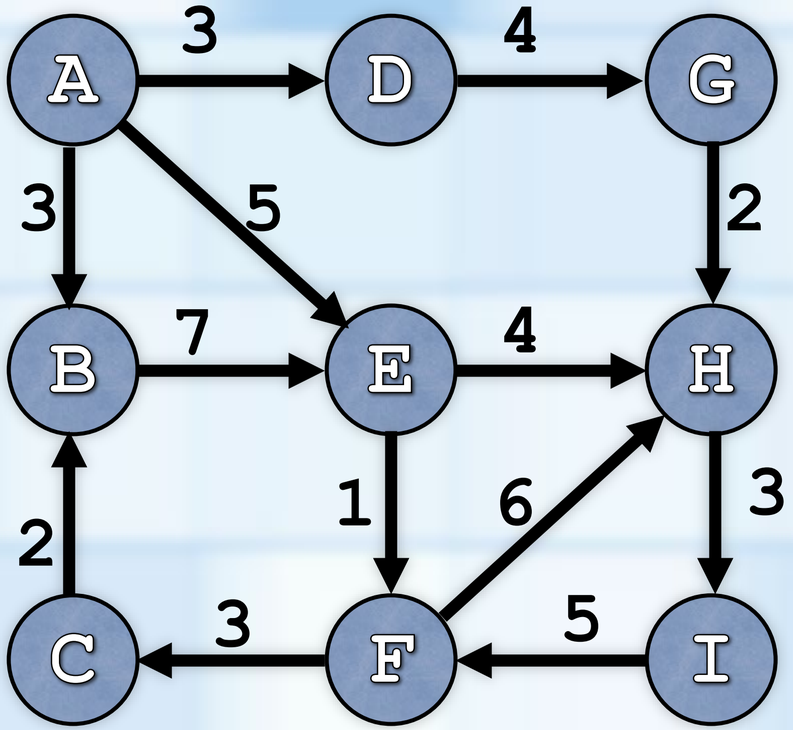
Pearson

The Adjacency Matrix

is connected to

Vertex

	A	B	C	D	E	F	G	H	I
A	∞	3	∞	3	5	∞	∞	∞	∞
B	∞	∞	∞	∞	7	∞	∞	∞	∞
C	∞	2	∞	∞	∞	∞	∞	∞	∞
D	∞	∞	∞	∞	∞	∞	4	∞	∞
E	∞	∞	∞	∞	∞	1	∞	4	∞
F	∞	∞	3	∞	∞	∞	∞	6	∞
G	∞	∞	∞	∞	∞	∞	∞	2	∞
H	∞	∞	∞	∞	∞	∞	∞	∞	3
I	∞	∞	∞	∞	∞	5	∞	∞	∞



Weighted Directed Graph

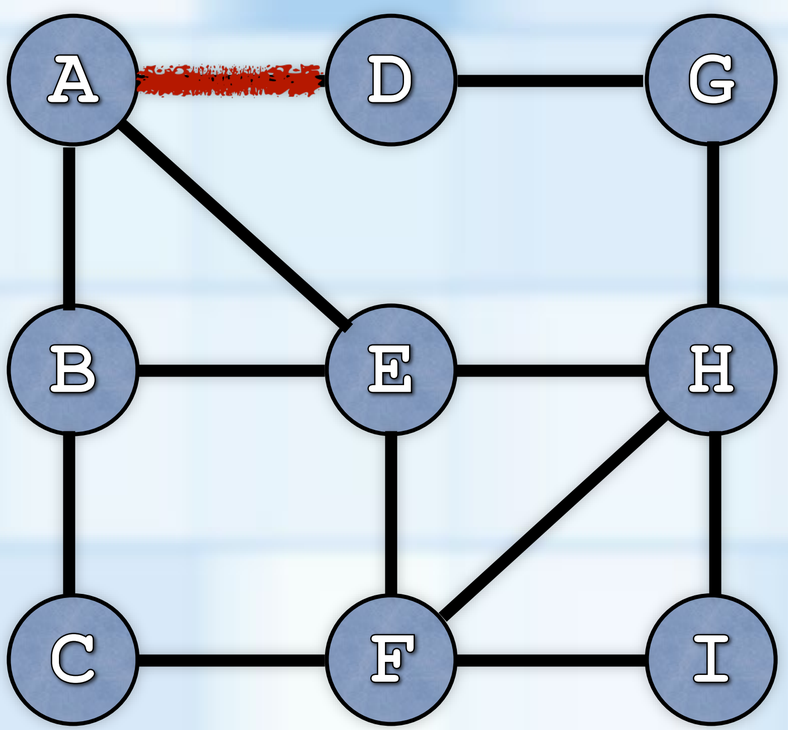
Pearson

The Adjacency Matrix

is connected to

Vertex

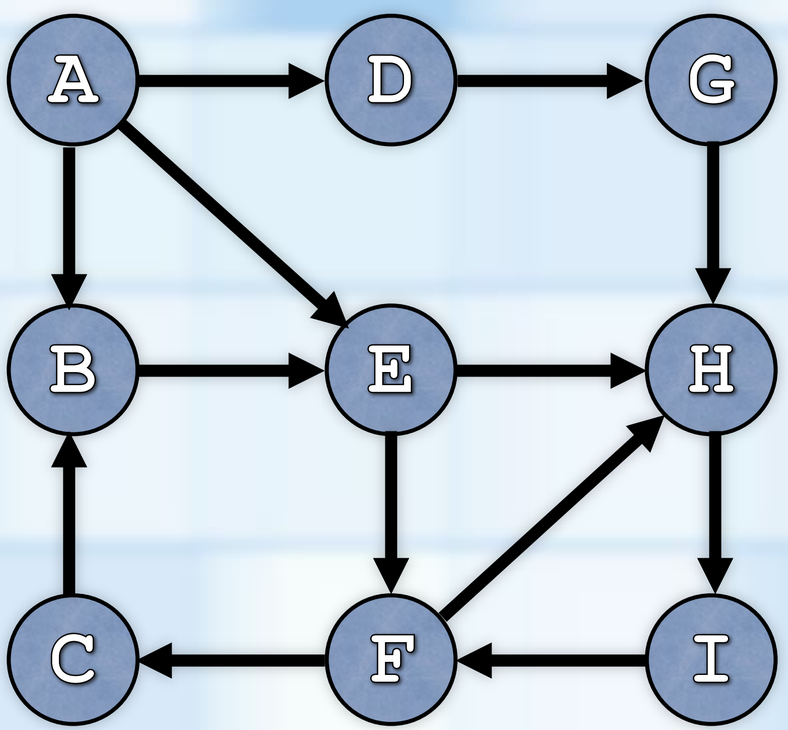
	A	B	C	D	E	F	G	H	I
A		1		1	1				
B	1		1		1				
C		1				1			
D	1						1		
E	1	1				1		1	
F			1		1			1	1
G				1				1	
H					1	1	1		1
I						1		1	



Undirected Graph

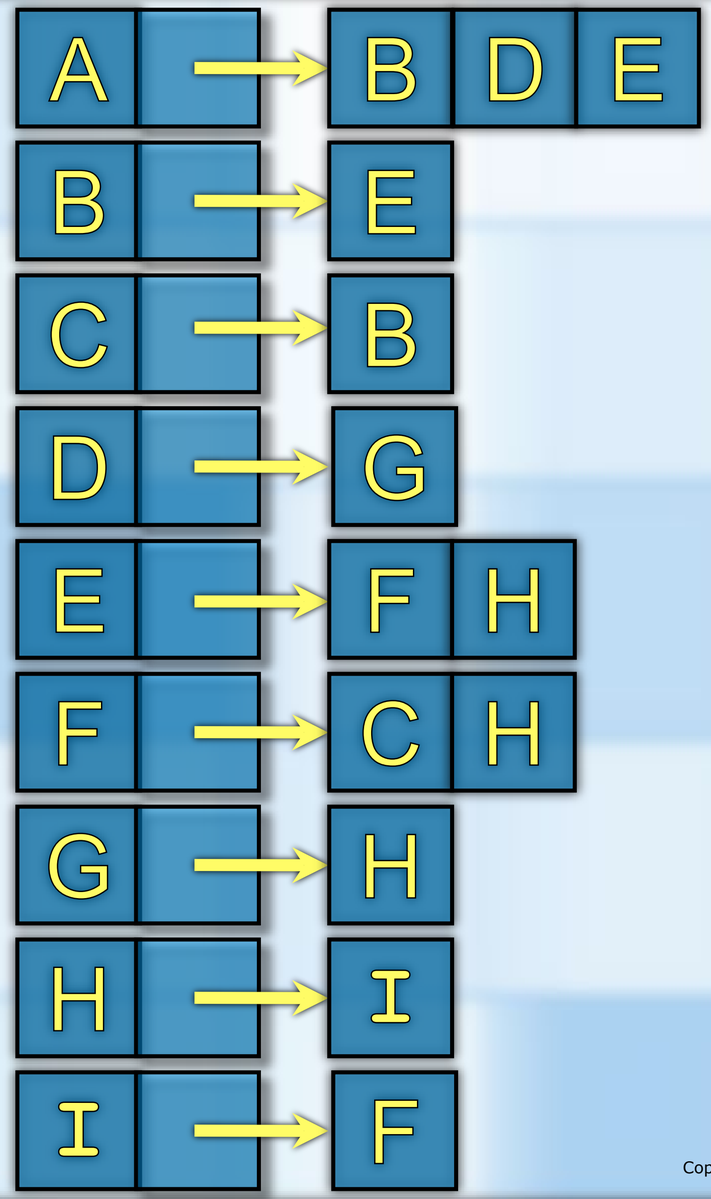
Adjacency Matrix Operations

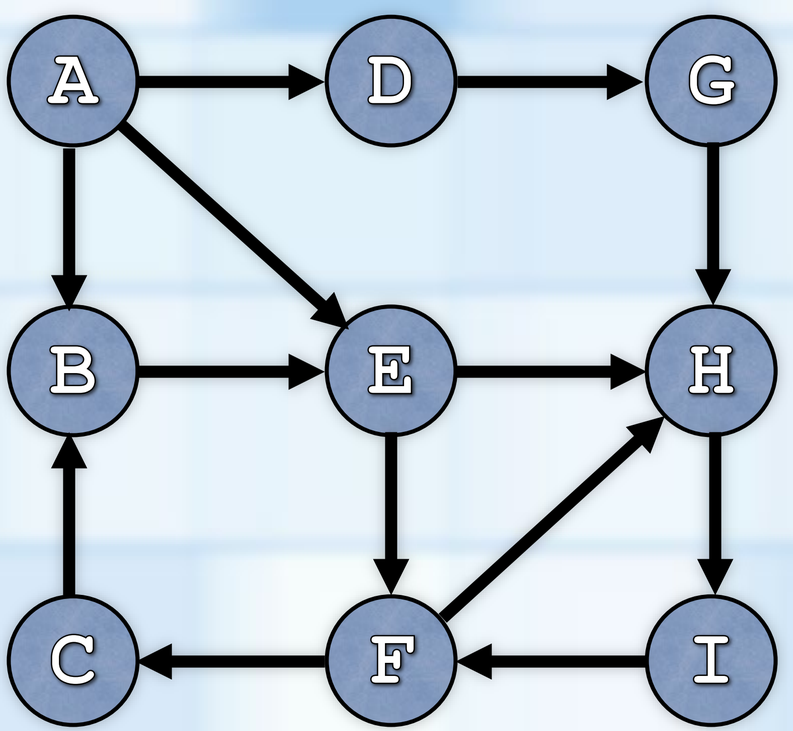
- Determining existence of an edge
  - O(1)
- Determining weight of an edge
  - O(1)
- Determining all neighbors of a vertex
  - O(n)
- Requires storage for n^2 values



Pearson

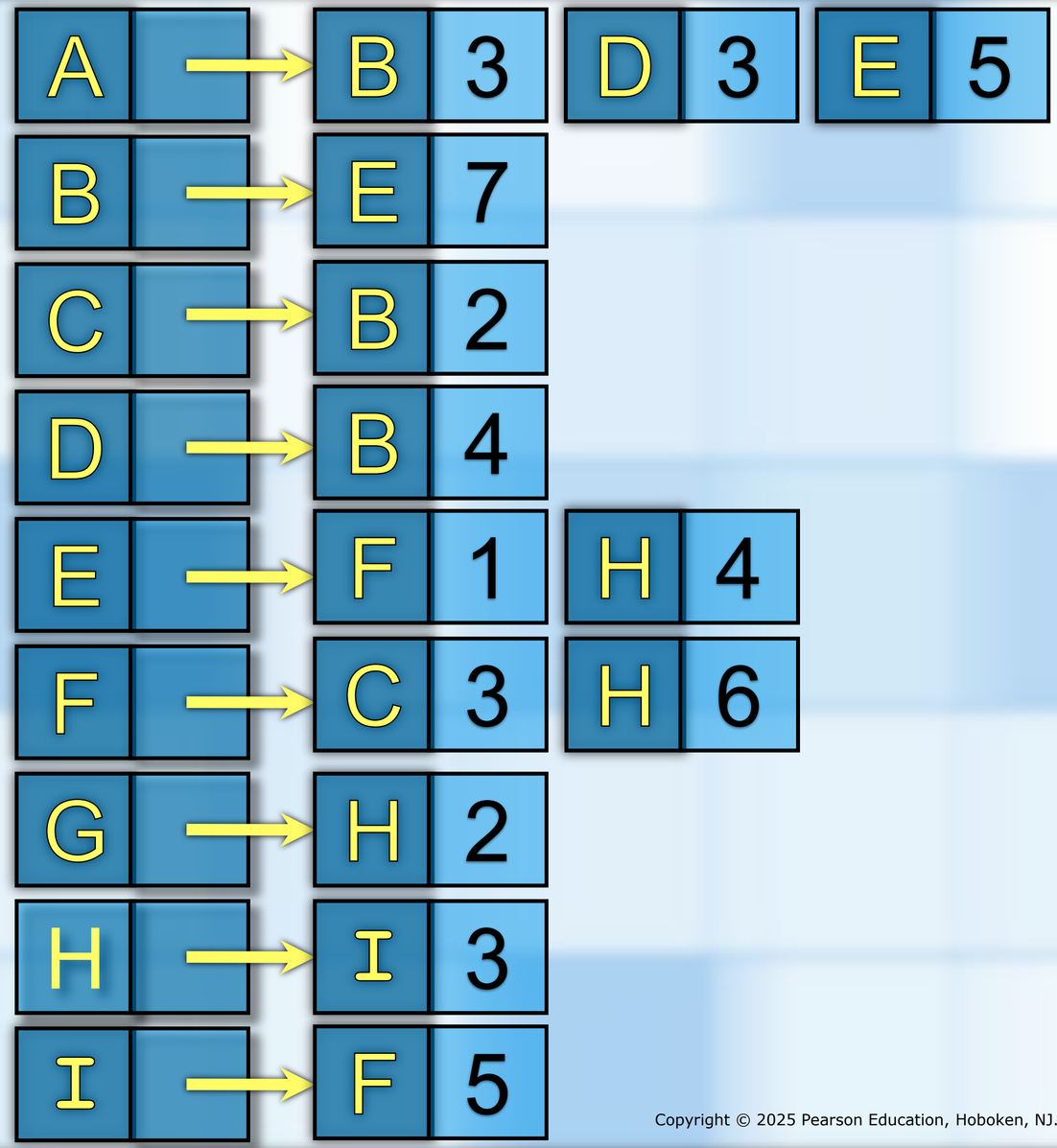
The Adjacency List

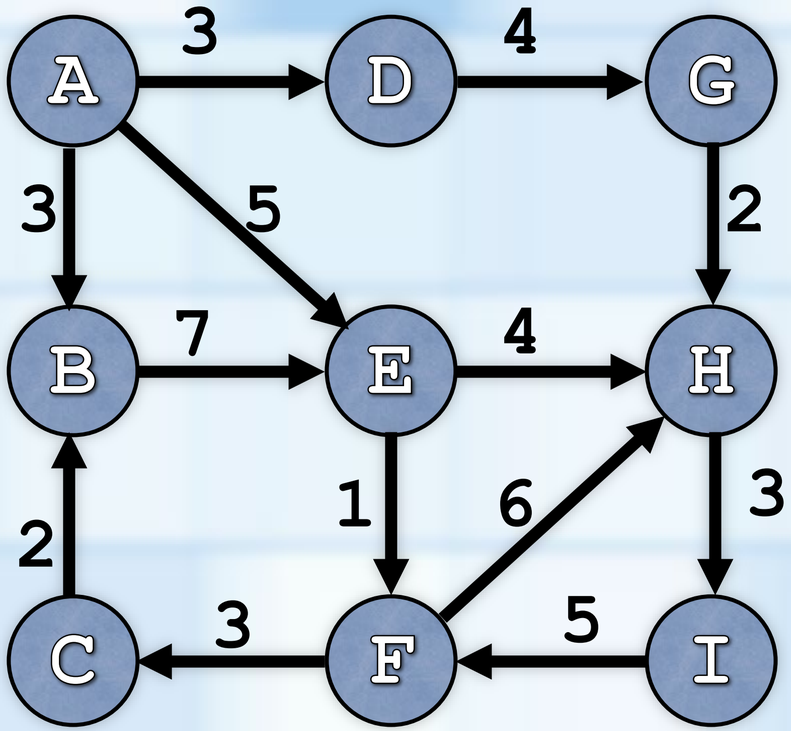




Directed Graph

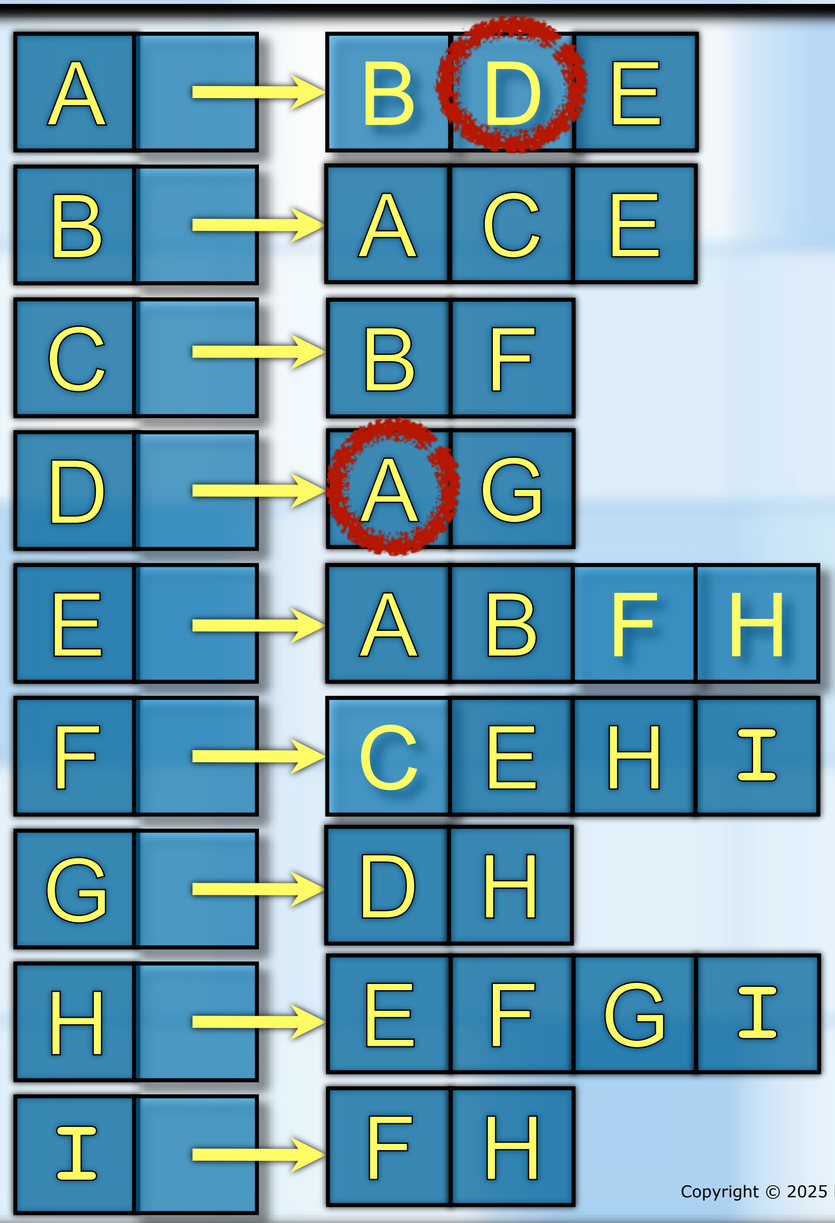
The Adjacency List

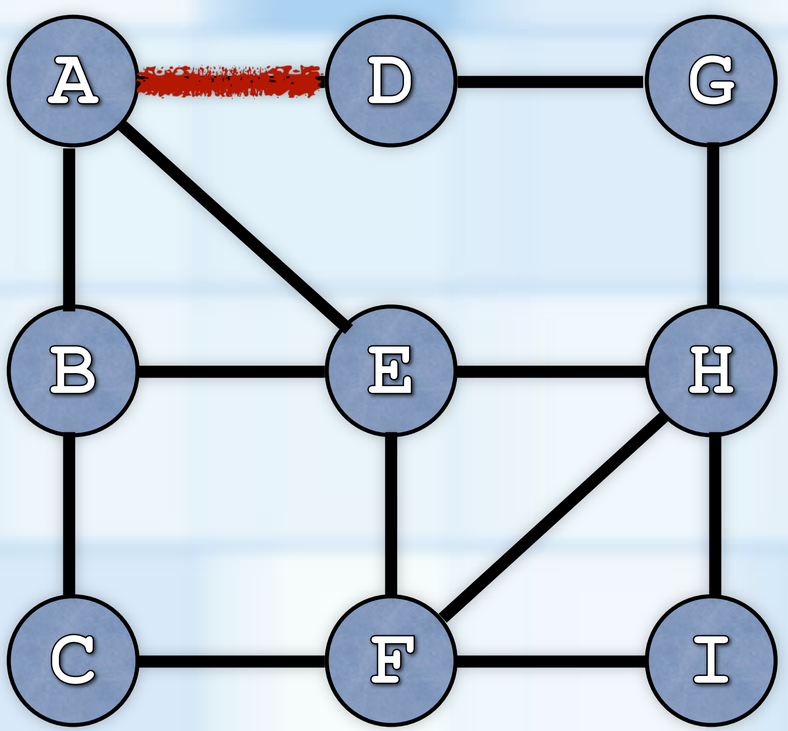




Weighted Directed Graph

The Adjacency List





Undirected Graph

Adjacency List Operations

- **Determining existence of an edge**
  - O(*n*) or O(log*n*) (*implementation dependent*)
- **Determining weight of an edge**
  - O(*n*) or O(log*n*) (*implementation dependent*)
- **Determining all neighbors of a vertex**
  - O(*n*)
- **Requires storage proportional to the number of edges**