НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА

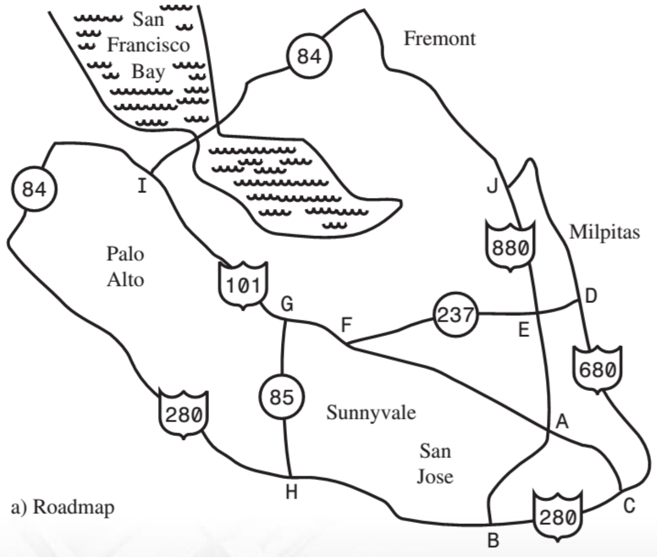
Data Structures and Organization

(p.7 – Graphs)

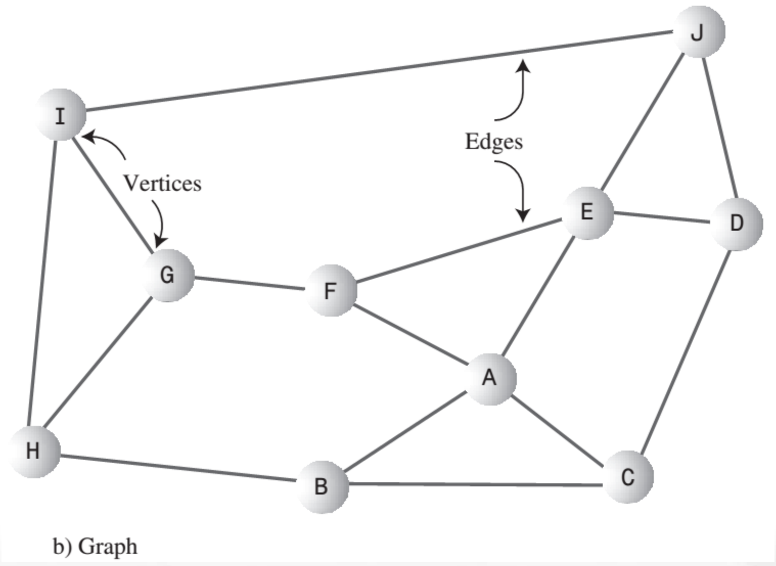
Yevhen Berkunskyi,
Computer Science dept., NUoS
eugeny.berkunsky@gmail.com
http://www.berkut.mk.ua

Graphs

- Figure **A** shows a simplified map of the freeways in the vicinity of San Jose, California.
- Figure **B** shows a graph that models these freeways.



a) Roadmap



b) Graph

Introduction to Graphs

- In the graph, circles represent freeway interchanges, and straight lines connecting the circles represent freeway segments.
- The circles are *vertices*, and the lines are *edges*.
- The vertices are usually labeled in some way—often, as shown here, with letters of the alphabet. Each edge is bounded by the two vertices at its ends

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

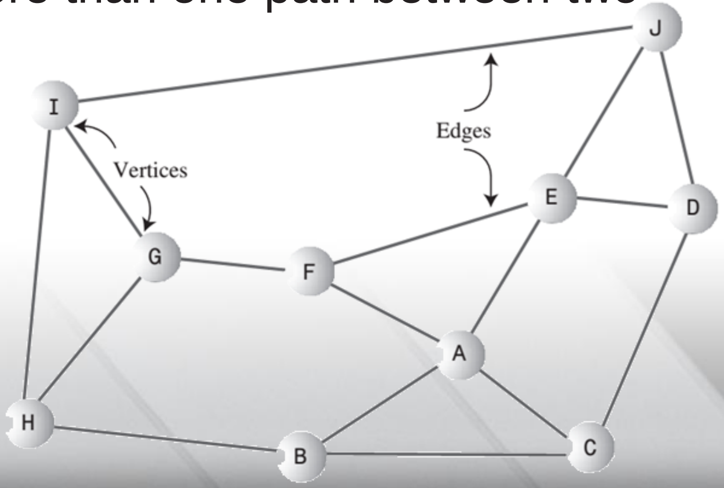
Graphs

Adjacency

- Two vertices are said to be *adjacent* to one another if they are connected by a single edge.
- Thus, in Figure, vertices I and G are adjacent, but vertices I and F are not.
- The vertices adjacent to a given vertex are sometimes said to be its *neighbors*. For example, the neighbors of G are I, H, and F.

Paths

- A *path* is a sequence of edges. Figure shows a path from vertex B to vertex J that passes through vertices A and E.
- We can call this path BAEJ. There can be more than one path between two vertices; another path from B to J is BCDJ.



b) Graph

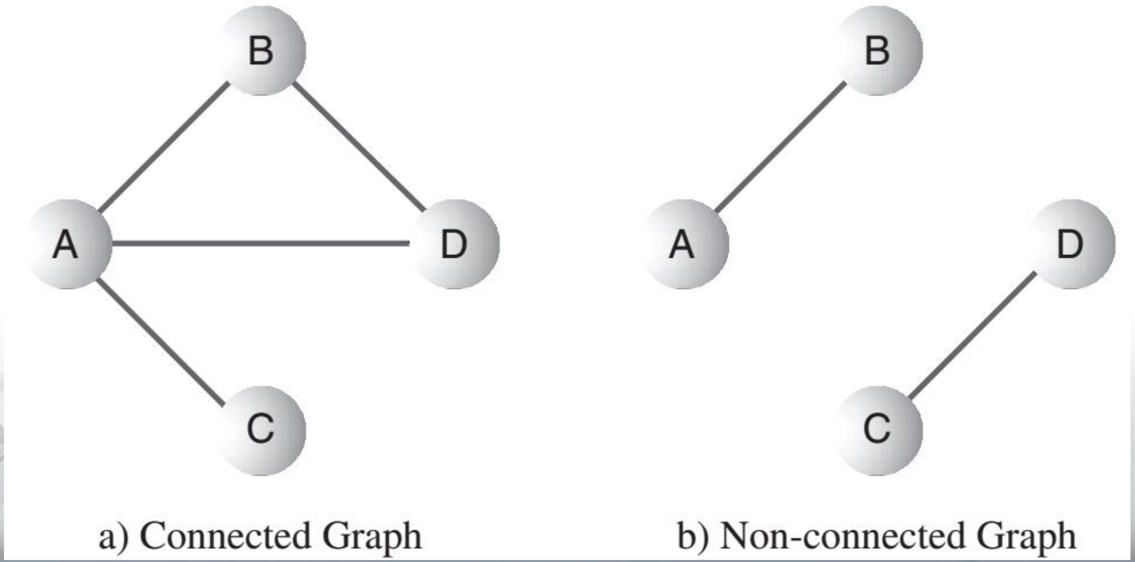
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Graphs

Connected Graphs

A graph is said to be *connected* if there is at least one path from every vertex to every other vertex, as in the graph in Figure **A**.
However, if “You can’t get there from here” (as Vermont farmers traditionally tell city slickers who stop to ask for directions), the graph is not connected, as in Figure **B**.

A non-connected graph consists of several connected components. In Figure 13.2b, A and B are one connected component, and C and D are another.



a) Connected Graph b) Non-connected Graph

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Graphs

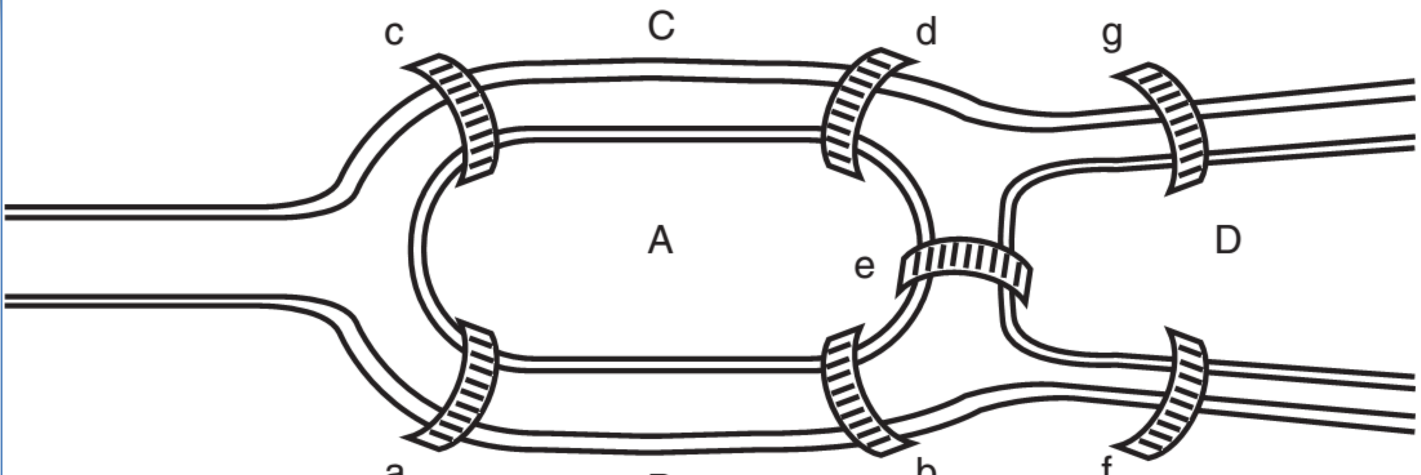
- In some graphs, edges are given a *weight*, a number that can represent the physical distance between two vertices, or the time it takes to get from one vertex to another, or how much it costs to travel from vertex to vertex (on airline routes, for example).
- Such graphs are called *weighted* graphs.

Historical Note

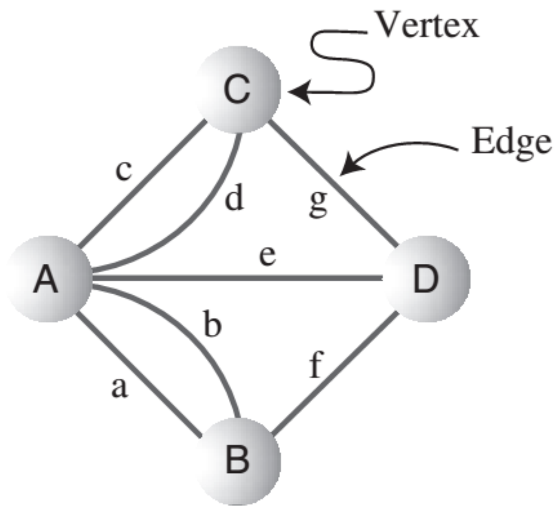
- One of the first mathematicians to work with graphs was Leonhard Euler in the early eighteenth century.
- He solved a famous problem dealing with the bridges in the town of Königsberg, Poland.
- This town included an island and seven bridges, as shown in Figure

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Historical Note



a) Map



b) Graph

Representing a Graph in a Program

Vertices

- In a very abstract graph program you could simply number the vertices 0 to N-1 (where N is the number of vertices). You wouldn’t need any sort of variable to hold the vertices because their usefulness would result from their relationships with other vertices.
- In most situations, however, a vertex represents some real-world object, and the object must be described using data items. If a vertex represents a city in an airline route simulation, for example, it may need to store the name of the city, its altitude, its location, and other such information.
- Thus, it’s usually convenient to represent a vertex by an object of a vertex class.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Representing a Graph in a Program

```
class Vertex {
    public char label; // label (e.g. 'A')
    public boolean wasVisited;

    public Vertex(char label) {
        this.label = label;
        wasVisited = false;
    }
}
```

The vertices might be placed in a list, an array or some other data structure.

Representing a Graph in a Program

Edges

- Two methods are commonly used for graphs: the adjacency matrix and the adjacency list. (Remember that one vertex is said to be adjacent to another if they're connected by a single edge.)

The Adjacency Matrix

- An adjacency matrix is a two-dimensional array in which the elements indicate whether an edge is present between two vertices. If a graph has N vertices, the adjacency matrix is an NxN array.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

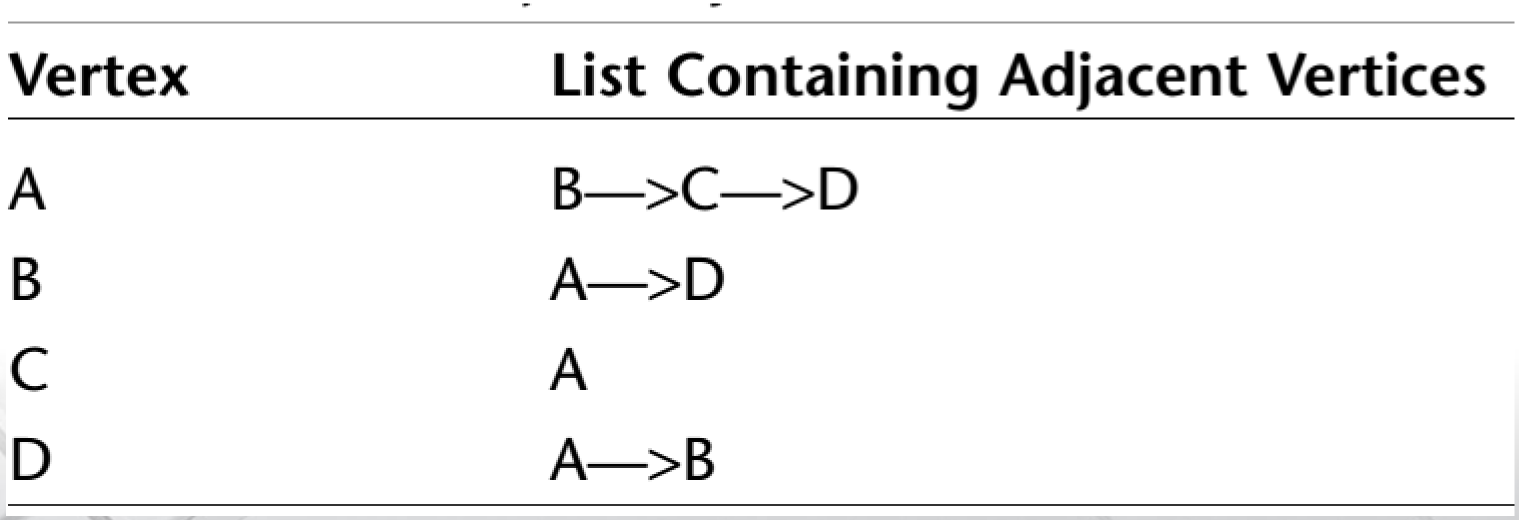
Adjacency Matrix

	A	B	C	D
A	0	1	1	1
B	1	0	0	1
C	1	0	0	0
D	1	1	0	0

Representing a Graph in a Program

The Adjacency List

- The other way to represent edges is with an adjacency list. The list in adjacency list refers to a linked list
- Actually, an adjacency list is an array of lists (or sometimes a list of lists).
- Each individual list shows what vertices a given vertex is adjacent to.



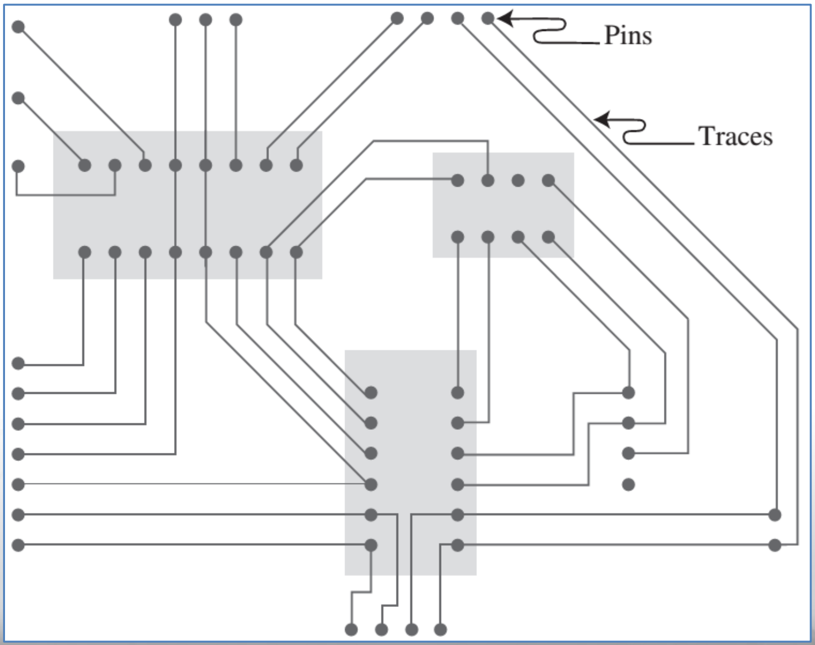
Vertex	List Containing Adjacent Vertices
A	B—>C—>D
B	A—>D
C	A
D	A—>B

Searches

- Imagine trying to find out how many towns in the United States can be reached by passenger train from Kansas City (assuming that you don’t mind changing trains).
- Some towns could be reached. Others couldn’t be reached because they didn’t have passenger rail service.
- Possibly others couldn’t be reached, even though they had rail service, because their rail system (the narrow-gauge Hayfork-Hicksville RR, for example) didn’t connect with the standard-gauge line you started on or any of the lines that could be reached from your line.

Searches

- Imagine that you’re designing a printed circuit board, like the ones inside your computer. Various components — mostly integrated circuits (ICs) — are placed on the board, with pins from the ICs protruding through holes in the board.
- The ICs are soldered in place, and their pins are electrically connected to other pins by traces — thin metal lines applied to the surface of the circuit board, as shown in Figure



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

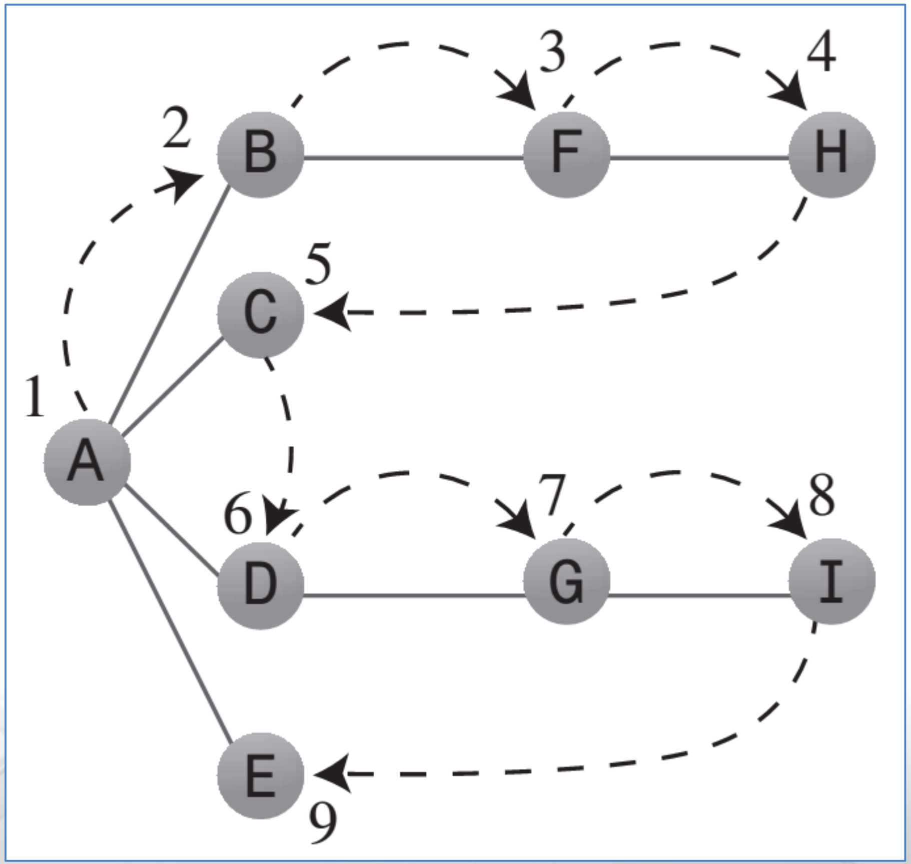
Searches

Assume that you’ve created such a graph. Now you need an algorithm that provides a systematic way to start at a specified vertex and then move along edges to other vertices in such a way that, when it’s done, you are guaranteed that it has visited every vertex that’s connected to the starting vertex.

- There are two common approaches to searching a graph: **depth-first search (DFS)** and **breadth-first search (BFS)**. Both will eventually reach all connected vertices.
- The **depth-first search** is implemented with a ***stack***, whereas the **breadth-first search** is implemented with a ***queue***.
- These mechanisms result, as we’ll see, in the graph being searched in different ways.

Depth-First Search

The depth-first search uses a stack to remember where it should go when it reaches a dead end.



Depth-First Search

RULE 1

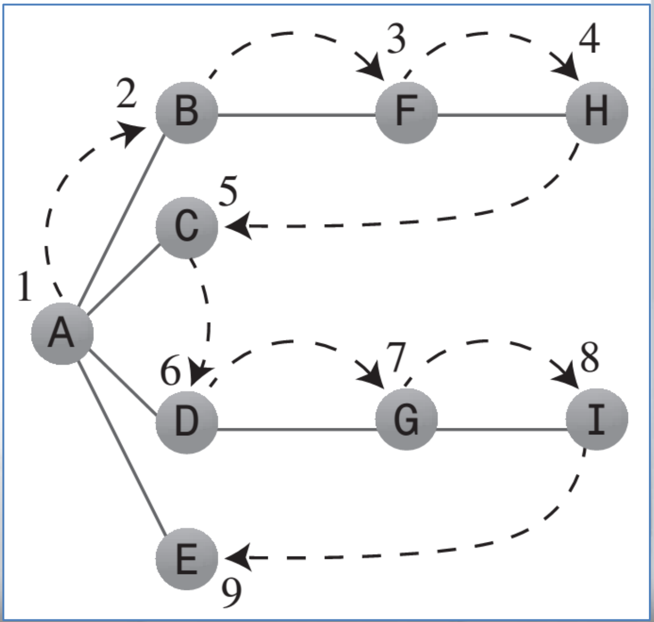
If possible, visit an adjacent unvisited vertex, mark it, and push it on the stack.

RULE 2

If you can't follow Rule 1, then, if possible, pop a vertex off the stack.

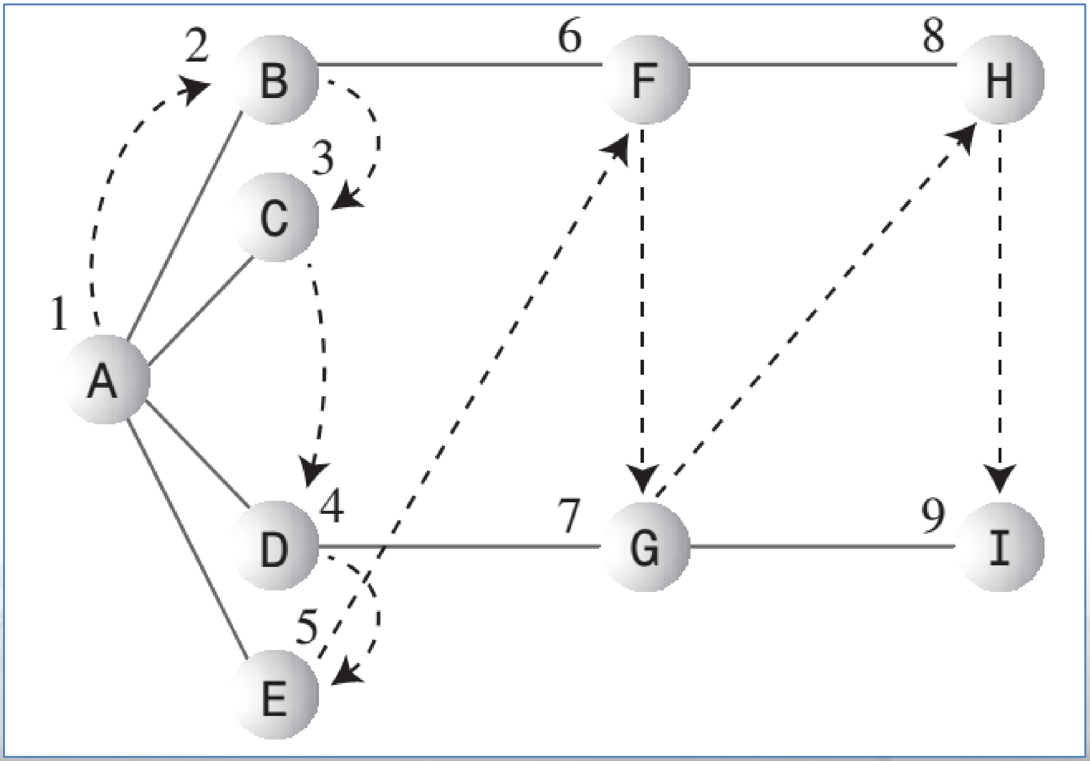
RULE 3

If you can't follow Rule 1 or Rule 2, you're done.



Breadth-First Search

- In the breadth-first search the algorithm likes to stay as close as possible to the starting point.
- It visits all the vertices adjacent to the starting vertex, and only then goes further afield. This kind of search is implemented using a queue instead of a stack.



Breadth-First Search

RULE 1

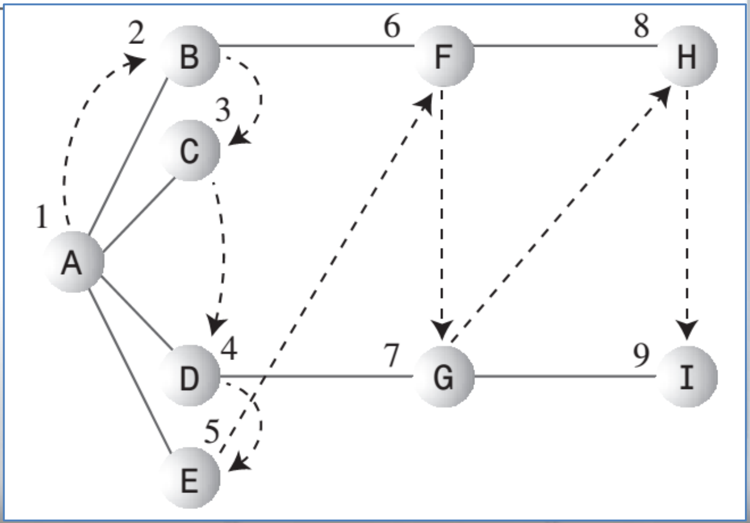
Visit the next unvisited vertex (if there is one) that's adjacent to the current vertex, mark it, and insert it into the queue.

RULE 2

If you can't carry out Rule 1 because there are no more unvisited vertices, remove a vertex from the queue (if possible) and make it the current vertex.

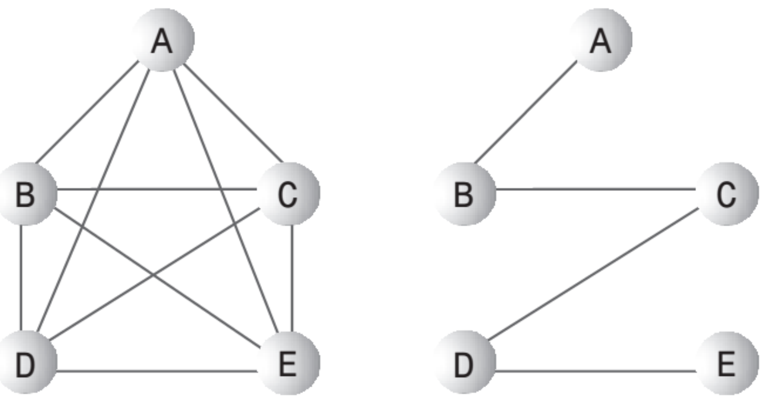
RULE 3

If you can't carry out Rule 2 because the queue is empty, you're done.



Minimum Spanning Trees

- Suppose that you've designed a printed circuit board like the one shown in Figure, and you want to be sure you've used the minimum number of traces. That is, you don't want any extra connections between pins; such extra connections would take up extra room and make other circuits more difficult to lay out.
- It would be nice to have an algorithm that, for any connected set of pins and traces (vertices and edges, in graph terminology), would remove any extra traces. The result would be a graph with the minimum number of edges necessary to connect the vertices.



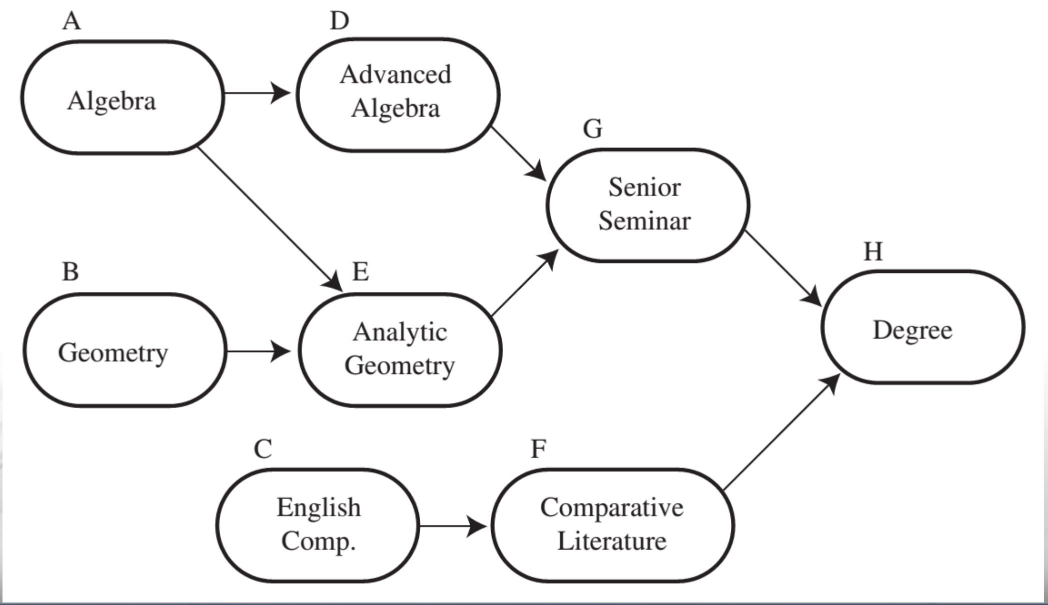
a) Extra Edges

b) Minimum Number of Edges

Topological Sorting with Directed Graphs

An Example: Course Prerequisites

- In high school and college, students find (sometimes to their dismay) that they can’t take just any course they want.
- Some courses have prerequisites — other courses that must be taken first. Indeed, taking certain courses may be a prerequisite to obtaining a degree in a certain field.
- Figure shows a somewhat fanciful arrangement of courses necessary for graduating with a degree in mathematics



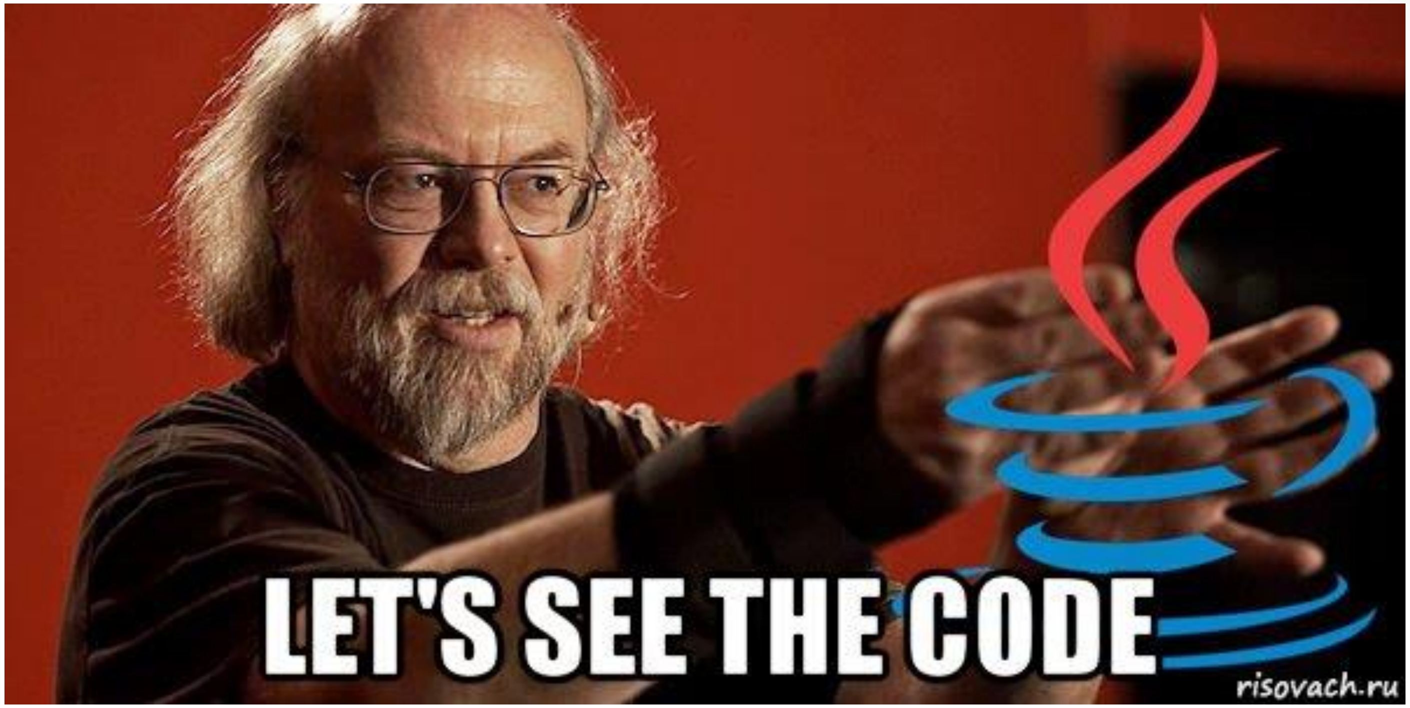
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Directed Graphs

- Previous Figure shows, a graph can represent this sort of arrangement. However, the graph needs a feature we haven’t seen before: The edges need to have a direction.
- When this is the case, the graph is called a directed graph. In a directed graph you can proceed only one way along an edge. The arrows in the figure show the direction of the edges.

To be continued…

Example



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Questions?

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Data Structures and Organization

(p.7 – Graphs)

Yevhen Berkunskyi,
Computer Science dept., NUoS
eugeny.berkunsky@gmail.com
http://www.berkut.mk.ua