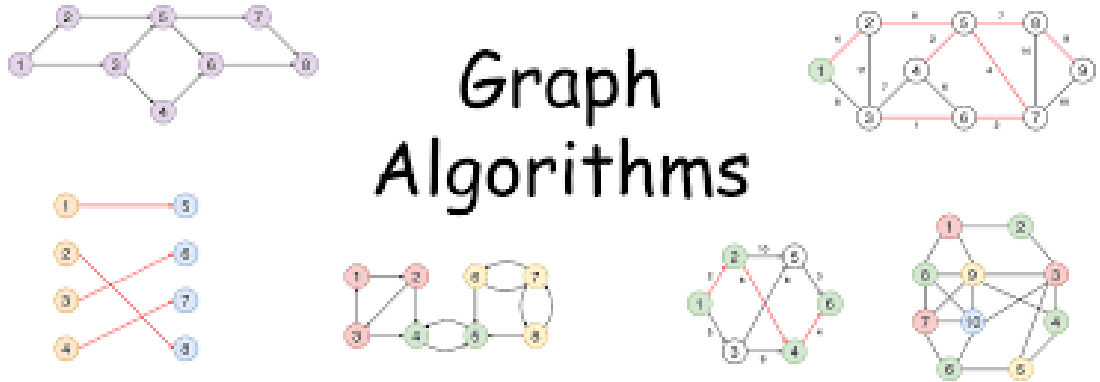
Introduction to Graphs

COMP215: Design & Analysis of Algorithms

Today

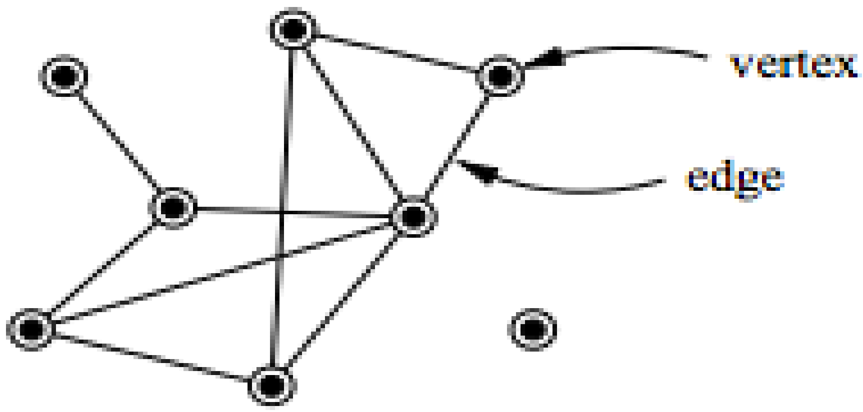


- Graph: Some Vocabulary.
- A Few Applications.
- Measuring the Size of a Graph.
- Representing a Graph

Graph: Some Vocabulary

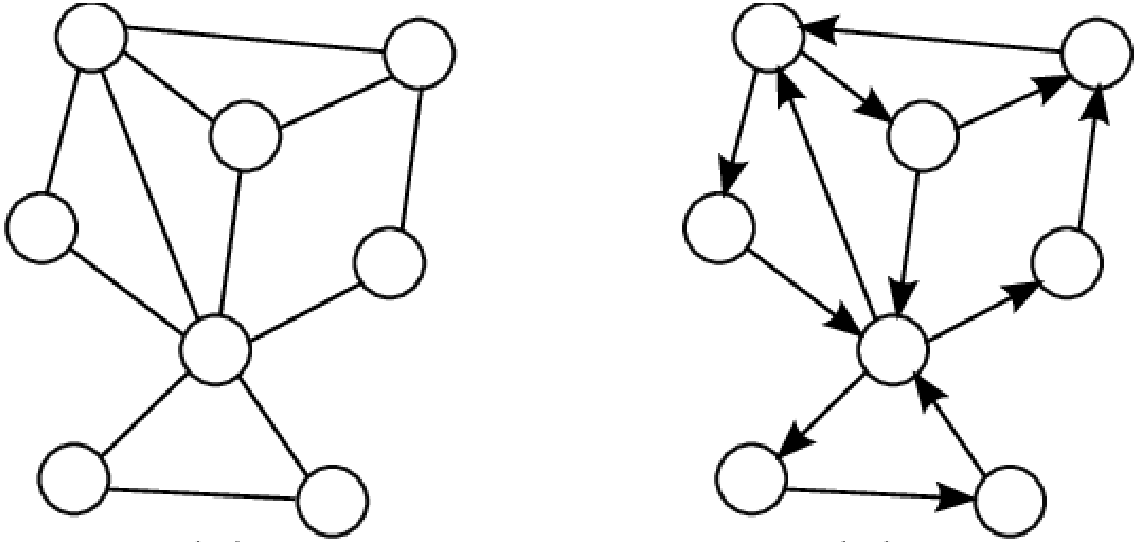
- Graph has two ingredients
  - The objects being represented, vertices (V)
  - Their pairwise relationships, edges (E)

- G = (V,E) to mean the graph G with vertices V and edges E.



Graph: Some Vocabulary

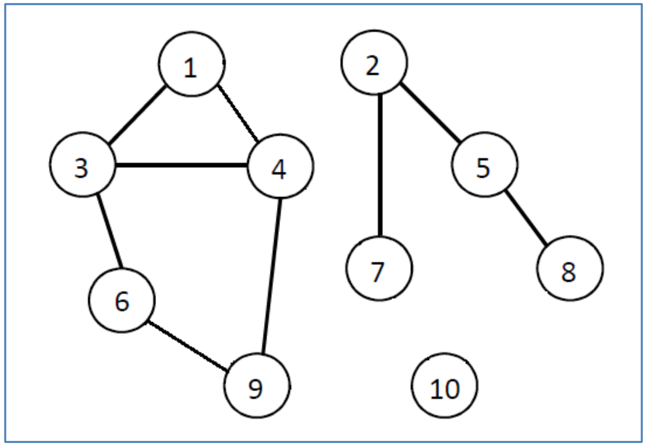
- There are two flavors of graphs:
  - Directed
  - Undirected

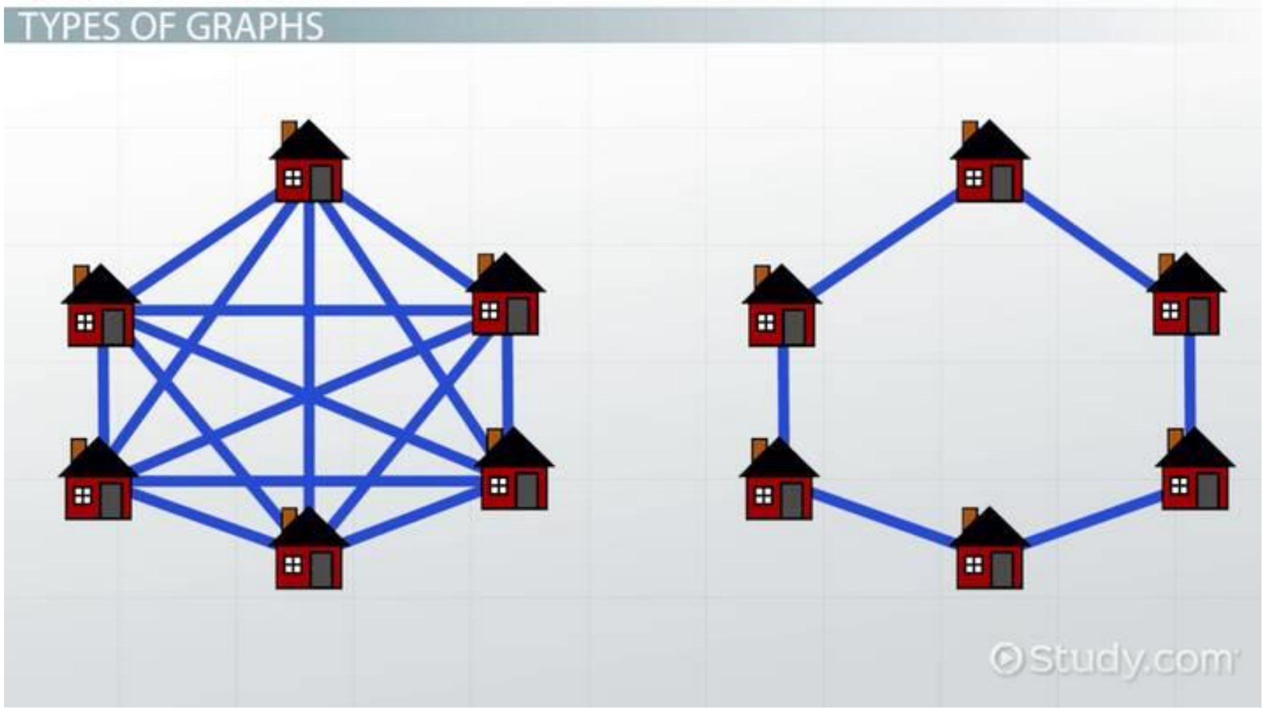


(a) (b)

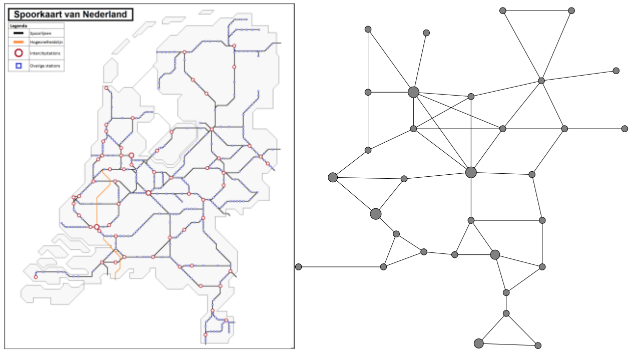
Types of Graphs

- Complete vs. Connected

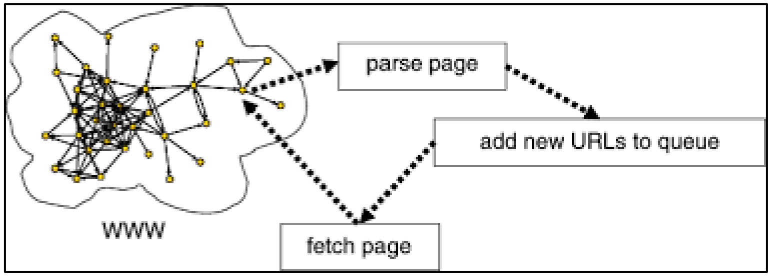




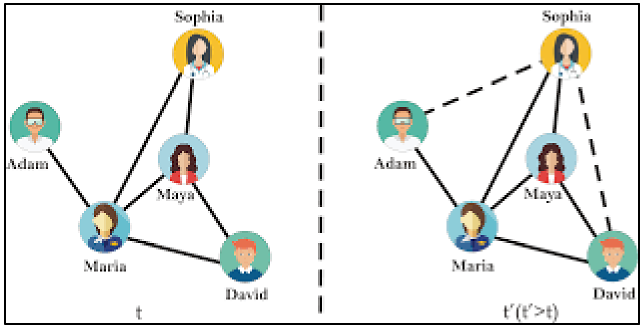
A Few Applications



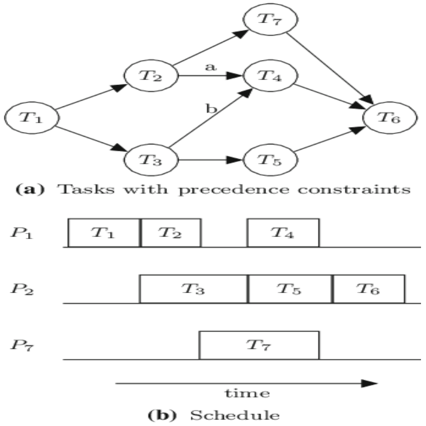
- Road networks.



- The World Wide Web.



- Social networks.



- Precedence constraints.

Measuring the Size of a Graph

- Two parameters control a graph's size
  - The number of vertices
  - The number of edges

Notation for Graphs

For a graph $G = (V, E)$ with vertex set V and edge set E:

- $n = |V|$ denotes the number of vertices; and
- $m = |E|$ denotes the number of edges.[3]

Measuring the Size of a Graph

Quiz 7.1

Consider an undirected graph with n vertices and no parallel edges. Assume that the graph is connected, meaning “in one piece.” What are the minimum and maximum numbers of edges, respectively, that the graph could have?

a) $n - 1$ and $\frac{n(n-1)}{2}$

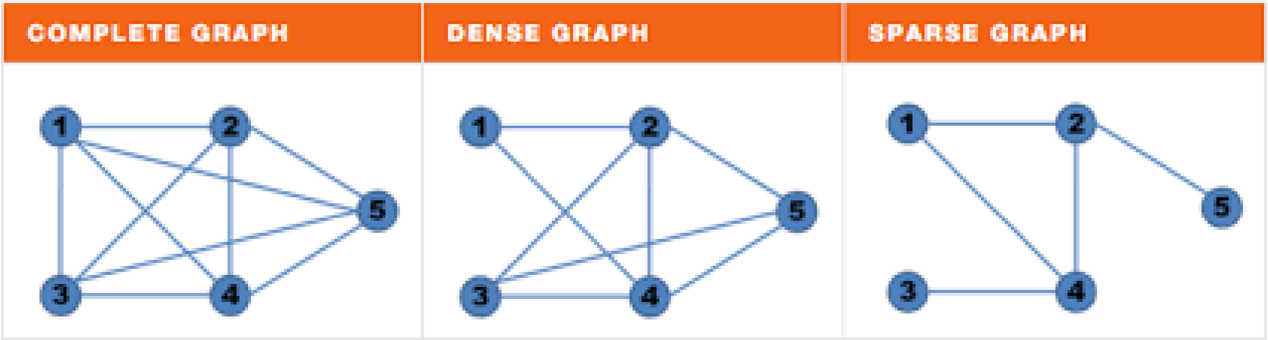
b) $n - 1$ and n^2

c) n and 2^n

d) n and n^n

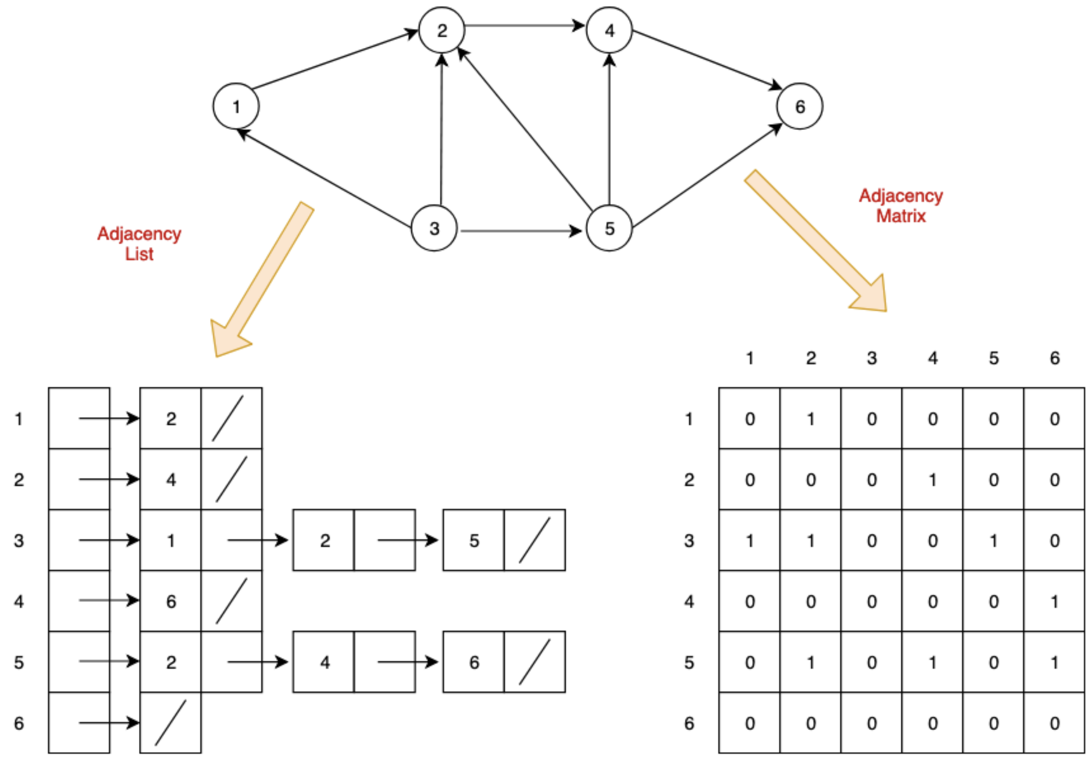
Measuring the Size of a Graph

- Sparse vs. Dense Graphs
  - A graph is sparse if the number of edges is relatively close to linear in the number of vertices, for example, graphs with n vertices and O(n log n) edges are usually considered sparse,
  - A graph is dense if this number is closer to quadratic in the number of vertices, for example, graphs with n vertices $\Omega(n^2 \log n)$ edges are considered dense



Representing a Graph

- There is more than one way to encode a graph for use in an algorithm.
  - Adjacency list
  - Adjacency matrix



Representing a Graph- Adjacency List

- The adjacency list representation is two arrays (or linked lists, if you prefer):
  - one for keeping track of the vertices,
  - one for the edges.
- These two arrays cross-reference each other , with each edge associated with pointers to its endpoints and each vertex with pointers to the edges for which it is an endpoint

Quiz 7.2

How much space does the adjacency list representation of a graph require, as a function of the number n of vertices and the number m of edges?

a) $\Theta(n)$

b) $\Theta(m)$

c) $\Theta(m+n)$

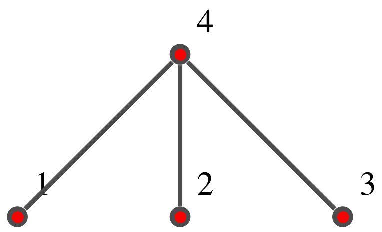
d) $\Theta(n^2)$

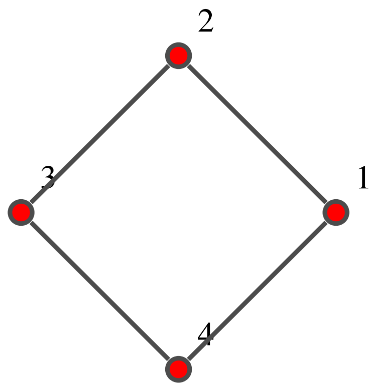
Representing a Graph- Adjacency Matrix

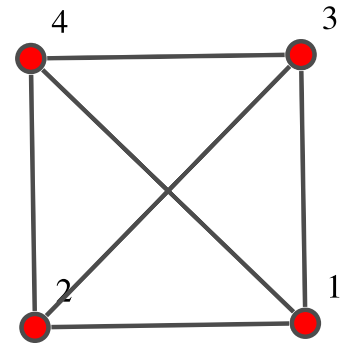
- Consider an undirected graph G = (V,E) with n vertices and no parallel edges, and label its vertices 1, 2, 3,...,n.
- The adjacency matrix representation of G is a **square n × n matrix** A(a two-dimensional array) with only zeroes and ones as entries. Each entry A_{ij} is defined as

$$A_{ij} = \begin{cases} 1 & \text{if edge } (i, j) \text{ belongs to } E \\ 0 & \text{otherwise.} \end{cases}$$

Representing a Graph- Adjacency Matrix







$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Representing a Graph- Adjacency Matrix

Using Adjacency Matrix, it is easy to handle:

- Parallel edges.
- Weighted graphs:
  - If each edge (i, j) has a weight w_{ij}, then each entry A_{ij} stores w_{ij}
- Directed graphs:
  - For a directed graph G, each entry A_{ij} of the adjacency matrix is defined as

$$A_{ij} = \begin{cases} 1 & \text{if edge } (i, j) \text{ belongs to } E \\ 0 & \text{otherwise,} \end{cases}$$

Representing a Graph- Adjacency Matrix

Quiz 7.3

How much space does the adjacency matrix of a graph require, as a function of the number n of vertices and the number m of edges?

a) $\Theta(n)$

b) $\Theta(m)$

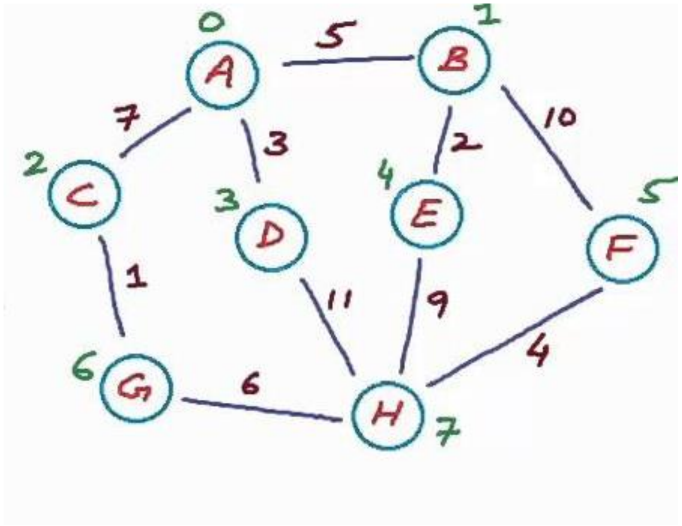
c) $\Theta(m+n)$

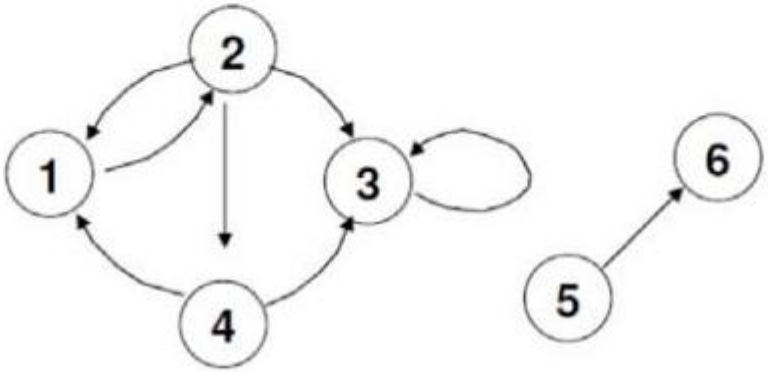
d) $\Theta(n^2)$

KALAMAZOO COLLEGE

Representing a Graph- Adjacency Matrix

- Adjacency Matrix?



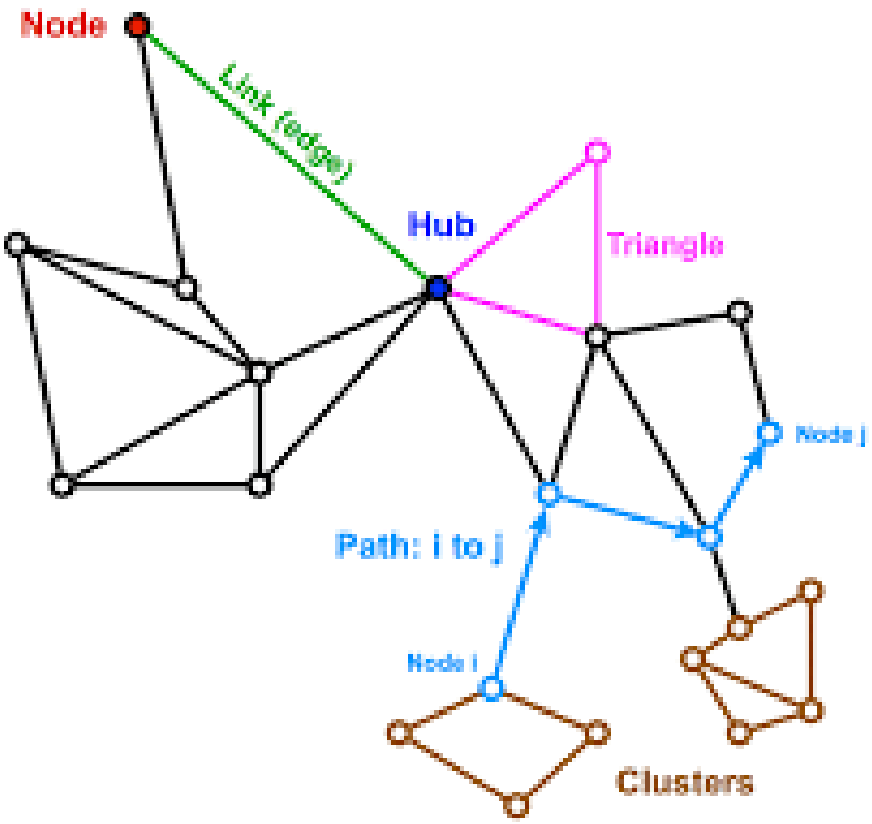


Graph Search and Its Applications

- Checking connectivity.
- Shortest paths.
- Planning.
- Connected components.

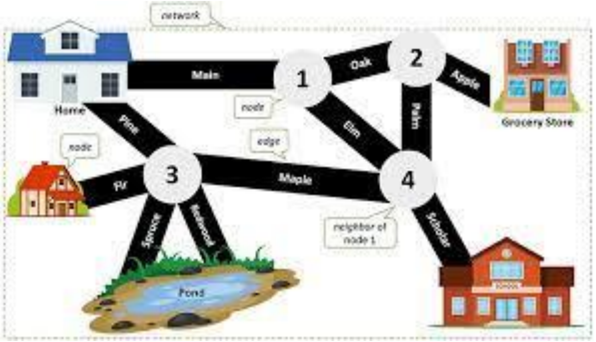
Checking connectivity

- You can get anywhere from anywhere else. Or, for every choice of a point A and a point B, there should be a path in the network from the former to the latter.



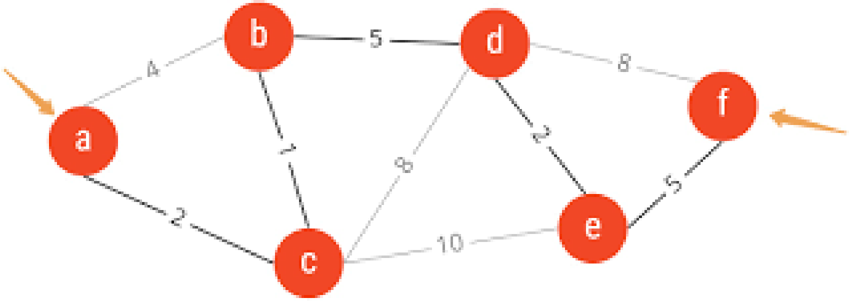
- Represents pairwise relationships between objects.

Shortest paths



- The path that using the fewest number of edges.

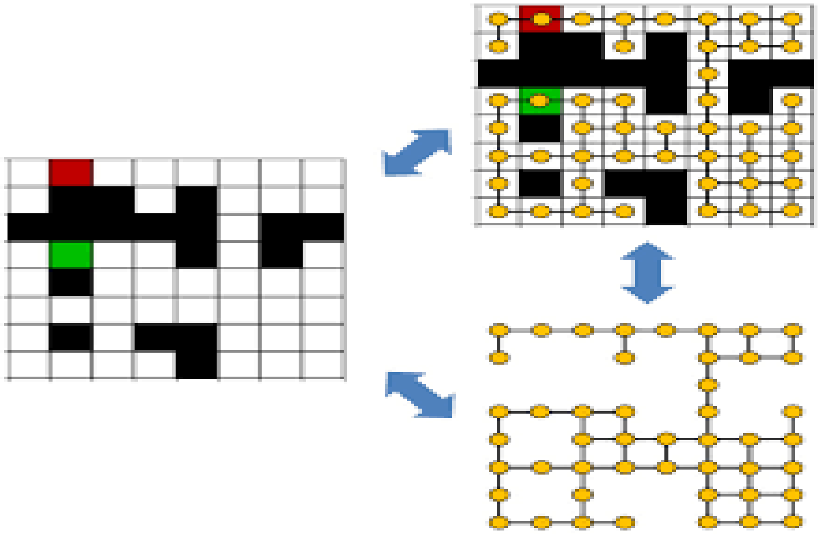
- Minimizing time for driving directions, or money for airline tickets, and so on



2 + 1 + 5 + 2 + 5 = 15

Planning

- A path is a sequence of decisions taking you from one state to another.
  - A robotic hand to grasp a coffee mug is essentially a planning problem



Generic Graph Search

Problem: Graph Search

Input: An undirected or directed graph $G = (V, E)$, and a starting vertex $s \in V$.

Goal: Identify the vertices of V reachable from s in G.

- By a vertex v being "reachable," we mean that there is a sequence of edges in G that travels from s to v

Generic Graph Search

`GenericSearch`

Input: graph $G = (V, E)$ and a vertex $s \in V$.
Postcondition: a vertex is reachable from s if and only if it is marked as "explored."

---

mark s as explored, all other vertices as unexplored
while there is an edge $(v, w) \in E$ with v explored and w unexplored **do**
    choose some such edge (v, w) `// underspecified`
    mark w as explored