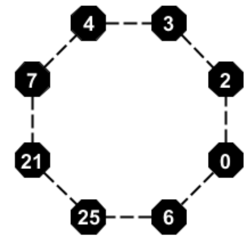
Cammini Minimi
(Shortest Paths)

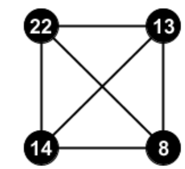
Alessandro Hill

rev. 1.0(AH) – 2024

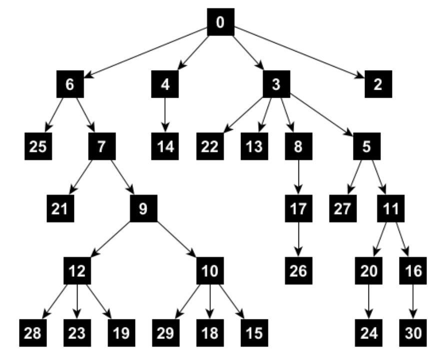
Networks and Notation

A **network** (or **graph**) consists of a set of **vertices** (or **nodes)** and a set of **edges** that connect selected pairs of nodes.



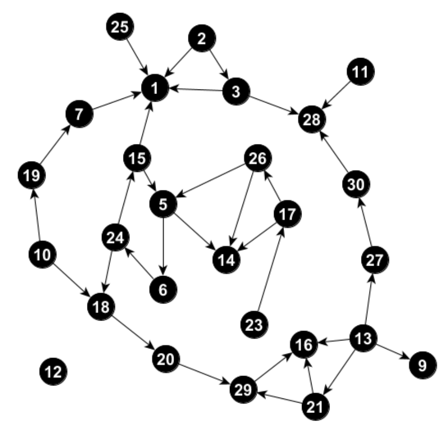


A **directed network** (or **oriented network**) consists of a set of **nodes** and a set of **arcs** (directed edges) that connect selected pairs of nodes.



The **neighbors** of a node are the nodes that are directly connected to the node (i.e., via edges or arcs).

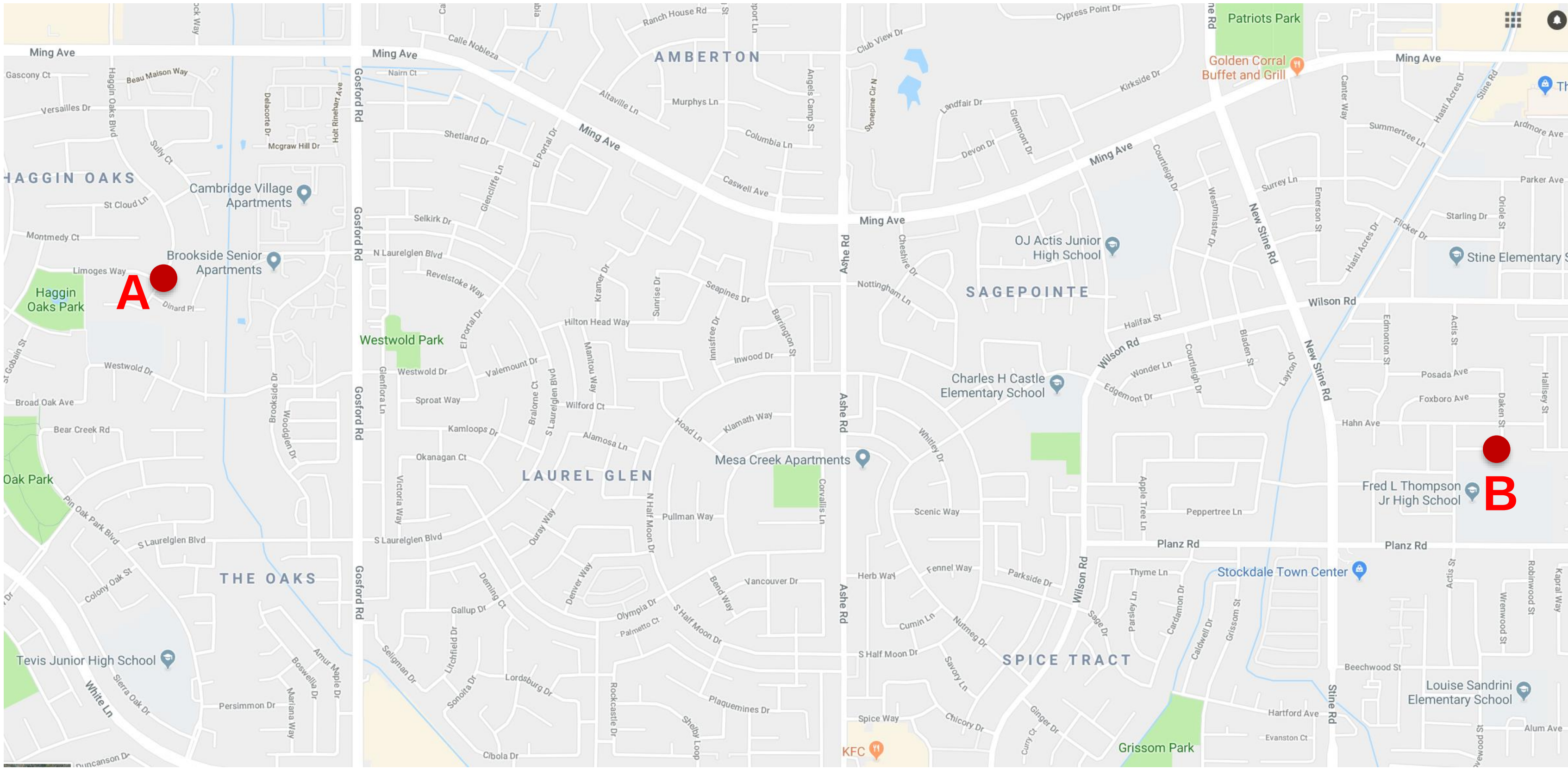
The **degree** of a node is its number of neighbors. For directed networks: **in-degree** and **out-degree**.



Nodes, edges and arcs may have **weights** (costs or profits) associated with them.

The Shortest Path Problem

What is the shortest path from **A** to **B**?



- **Nodes:** Intersections, U-turn locations
- **Arcs:** Street segments between nodes
- **Possible arc weights:** Distance, travel time, etc.

Applications: Route planning, transportation, machine control, etc.

Bike, drive, walk, run?

The Shortest Path Problem

IP Model (shortest path from A to B)

Binary arc variables for each arc that could be part of a shortest path:

$$x_{i,j} = \begin{cases} 1 & \text{if arc } (i,j) \text{ will be used on the shortest path,} \\ 0 & \text{otherwise.} \end{cases}$$

Objective: Minimize $\sum_{(i,j)\in A} w_{i,j} x_{i,j}$

Force source node out-flow:

$x_{A,j_1} + \ldots + x_{A,j_k} = 1$ for out-neighbors $j_1, \ldots, j_k$ of A.

Conserve flow through intermediate nodes:

$$x_{i_1,v} + \ldots + x_{i_k,v} = x_{v,j_1} + \ldots + x_{v,j_k}$$

for in-neighbors $i_1, \ldots, i_k$, out-neighbors $j_1, \ldots, j_k$ and each intermediate node v.

Force sink node in-flow:

$x_{i_1,B} + \ldots + x_{i_k,B} = 1$ for in-neighbors $i_1, \ldots, i_k$ of B.

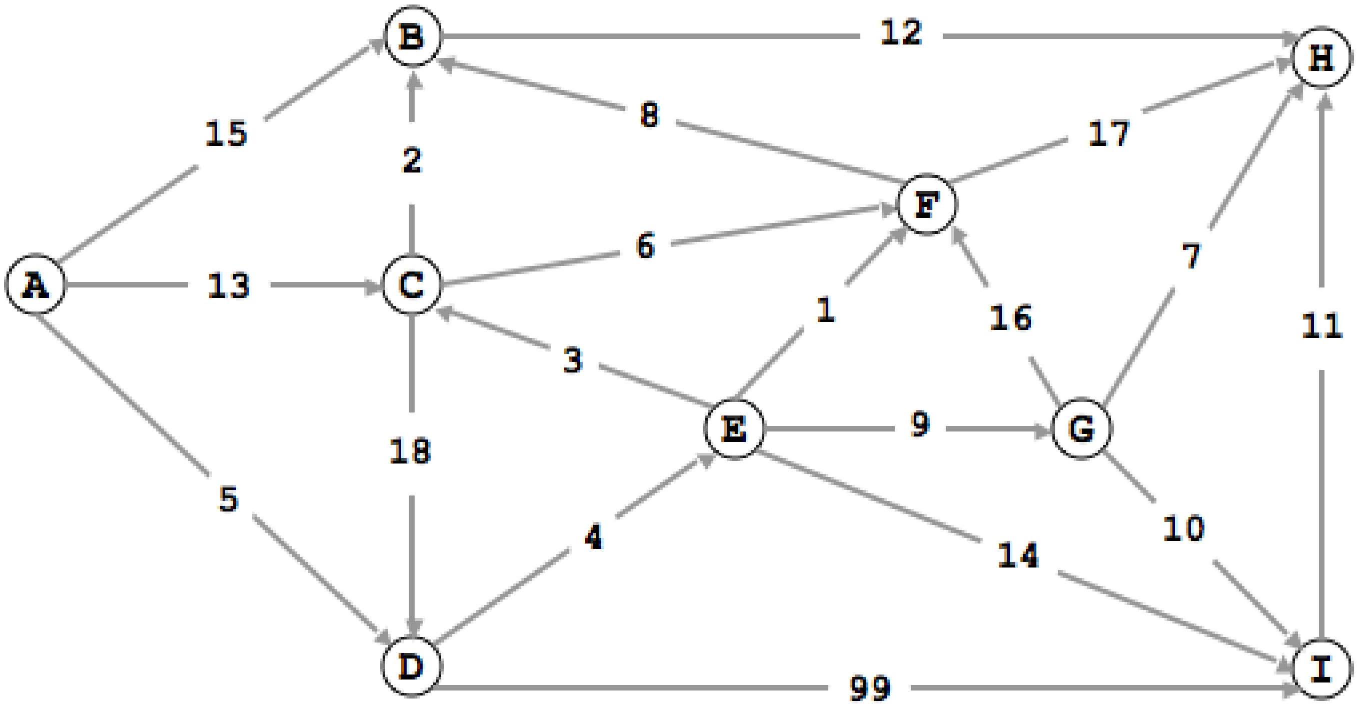
The Shortest Path Problem

Dijkstra's Algorithm, 1956 (shortest path from A to all other nodes)

1. Initialize node distances: $d(i) = \infty$ for each node I
2. Set distance to zero for start node: $d(A) = 0$
3. Mark all nodes as unvisited
4. Pick unvisited node i that has minimum distance
5. For each unvisited node j that can be reached from i:
   - Distance update $d(j) = \min(d(j) , d(i) + w(i,j))$
   - If distance was updated, set i as predecessor of j: PRED(j) = i
   - Mark i as visited
6. Go to 4 unless all nodes are visited
7. Return shortest path tree encoded in PRED

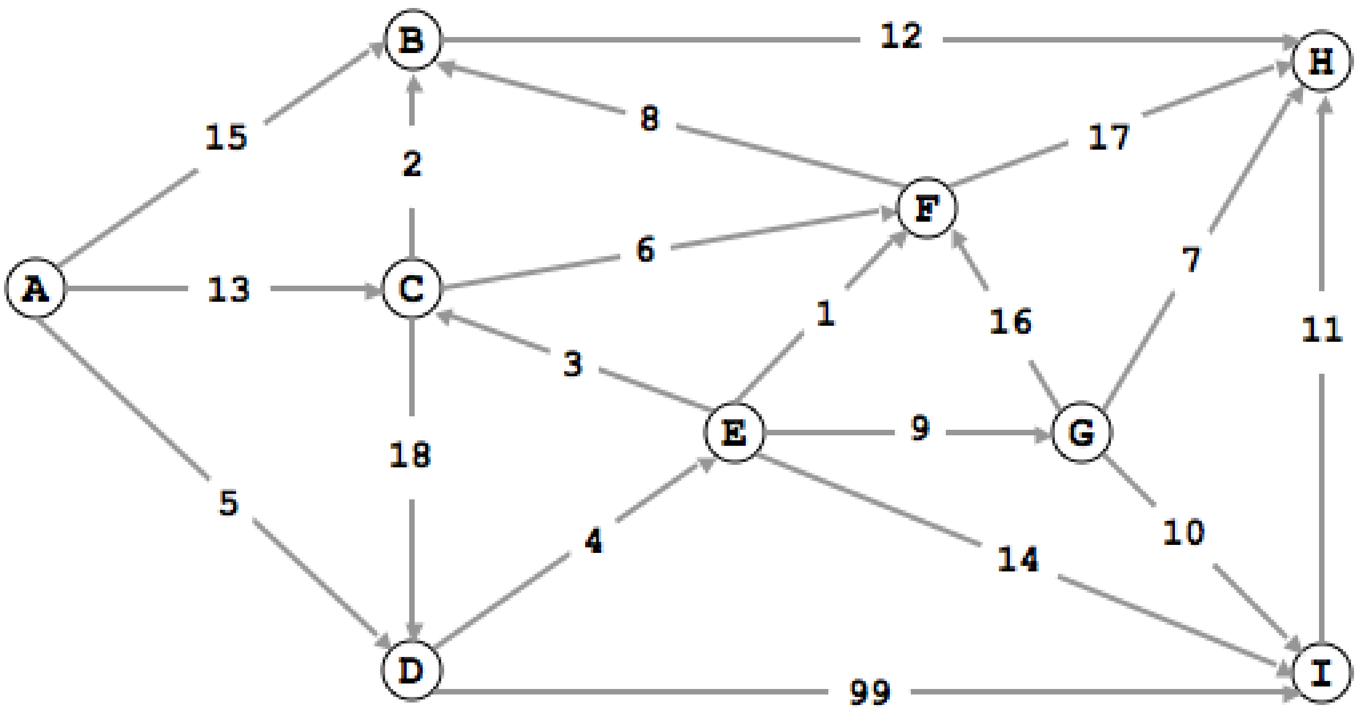
The Shortest Path Problem

1) What is the shortest path from node A to node H in the network given below?



The Shortest Path Problem

2) What is the shortest path from node A to node H in the network given below?

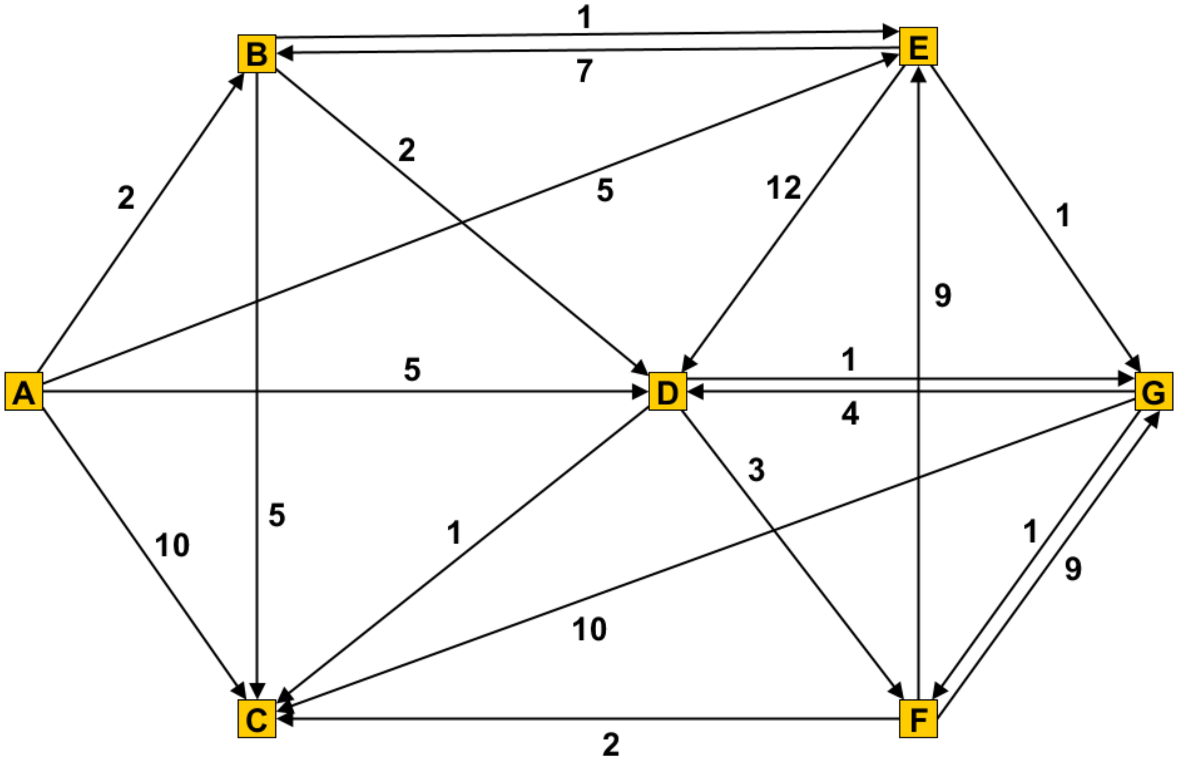


The Shortest Path Problem

1) Find shortest paths using Dijkstra's algorithm from

a) node A

b) node F



2) Build an IP and find the shortest paths from

c) node A to node G

d) node C to node E