CS2040 Data Structures and Algorithms
Lecture Note #11 – Part 1

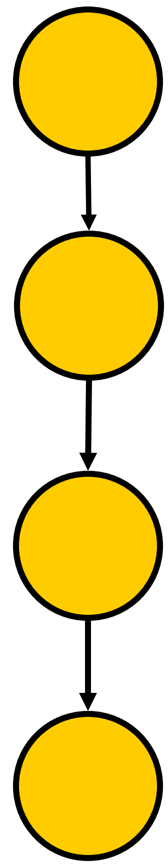
Graphs
Part 1: Introduction

COVID-19 and CS2040

- **Question**: Given a population of N, how many rounds of infections is needed to infect all N people?
  - (Apparently SG $N \approx 6{,}000{,}000$ on 8 Oct. 2022)
  - R (reproduction) number is important

- **Answer**: O(log N)
  (if we consider the spread to be a tree. However, it is not.)

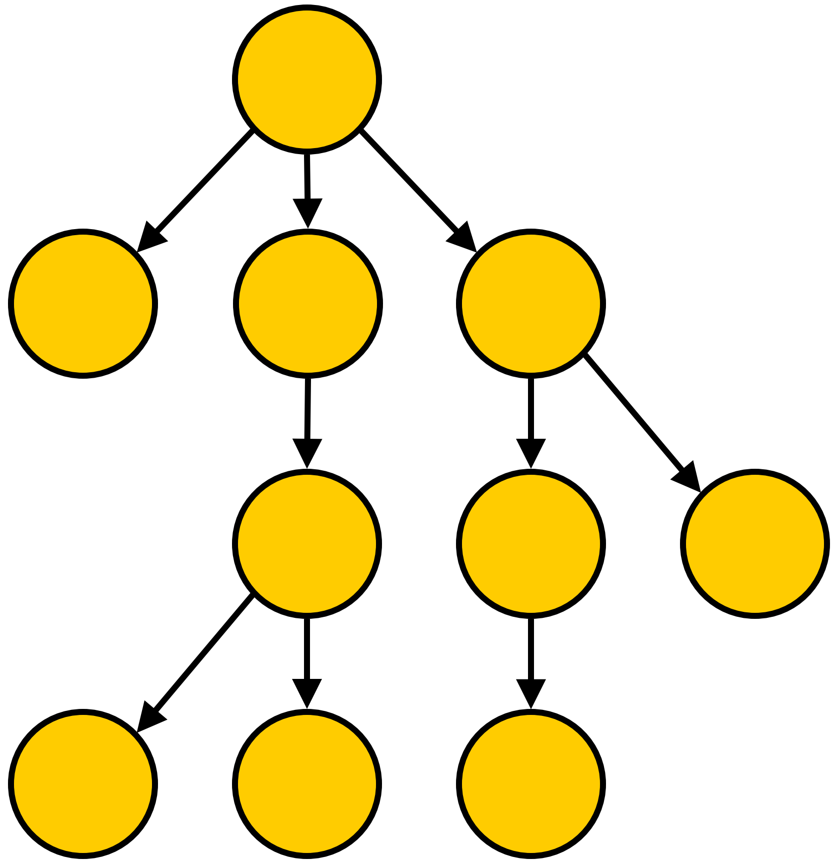
Linked list



Linear data structure

Some operations will take O(n)

Tree

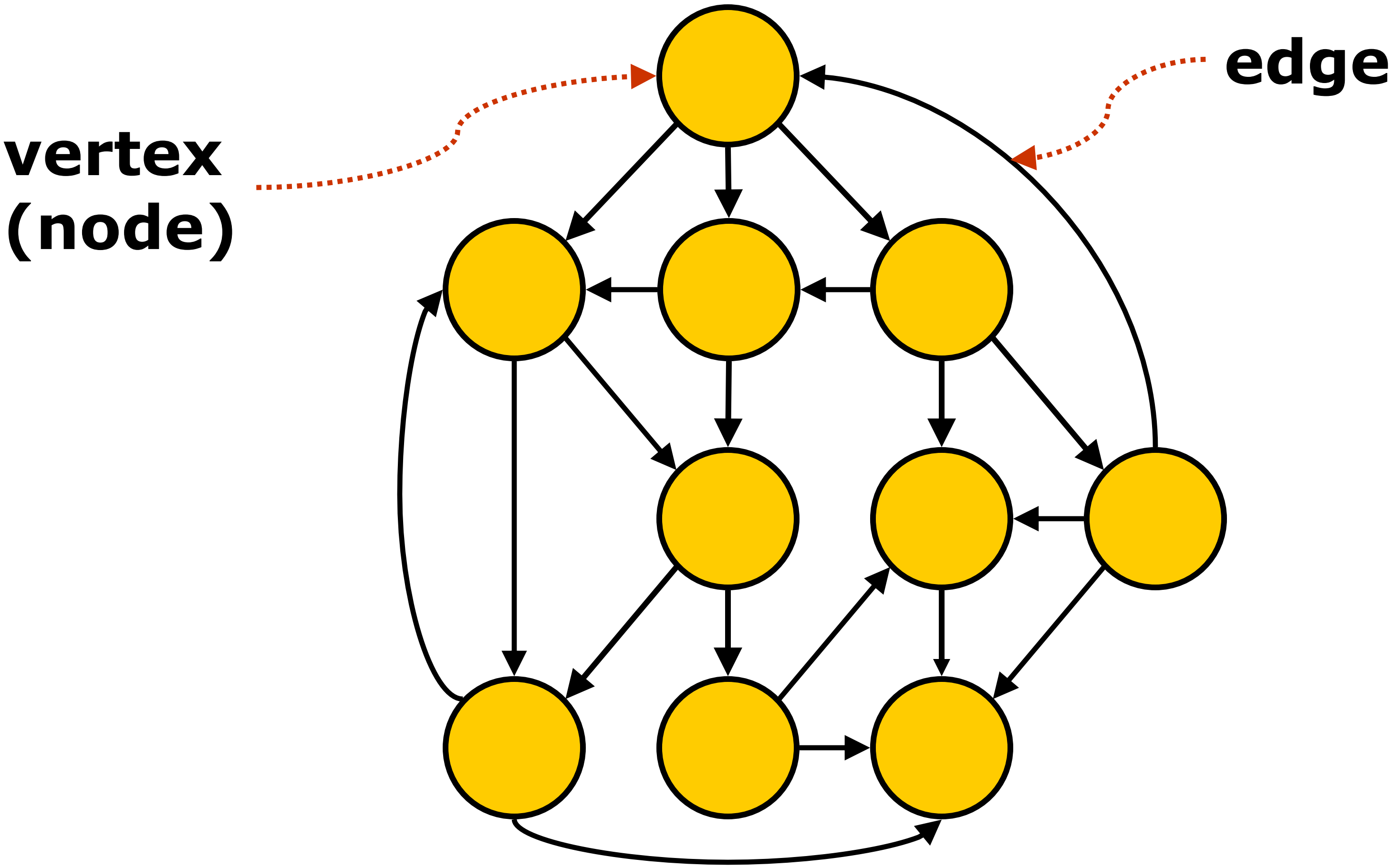


E.g., AVL or Heap

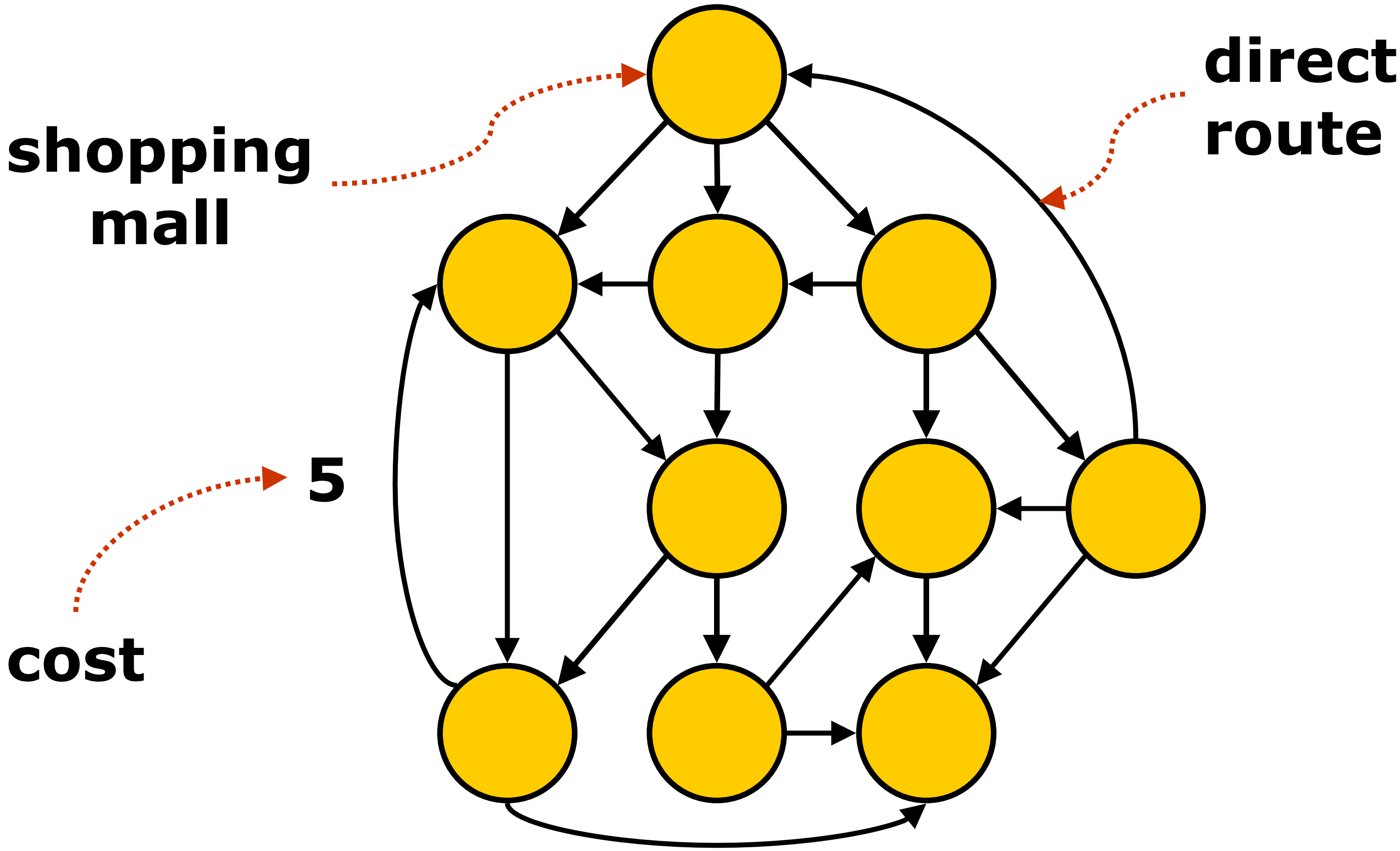
Tree has no loops

Only one path from A to B

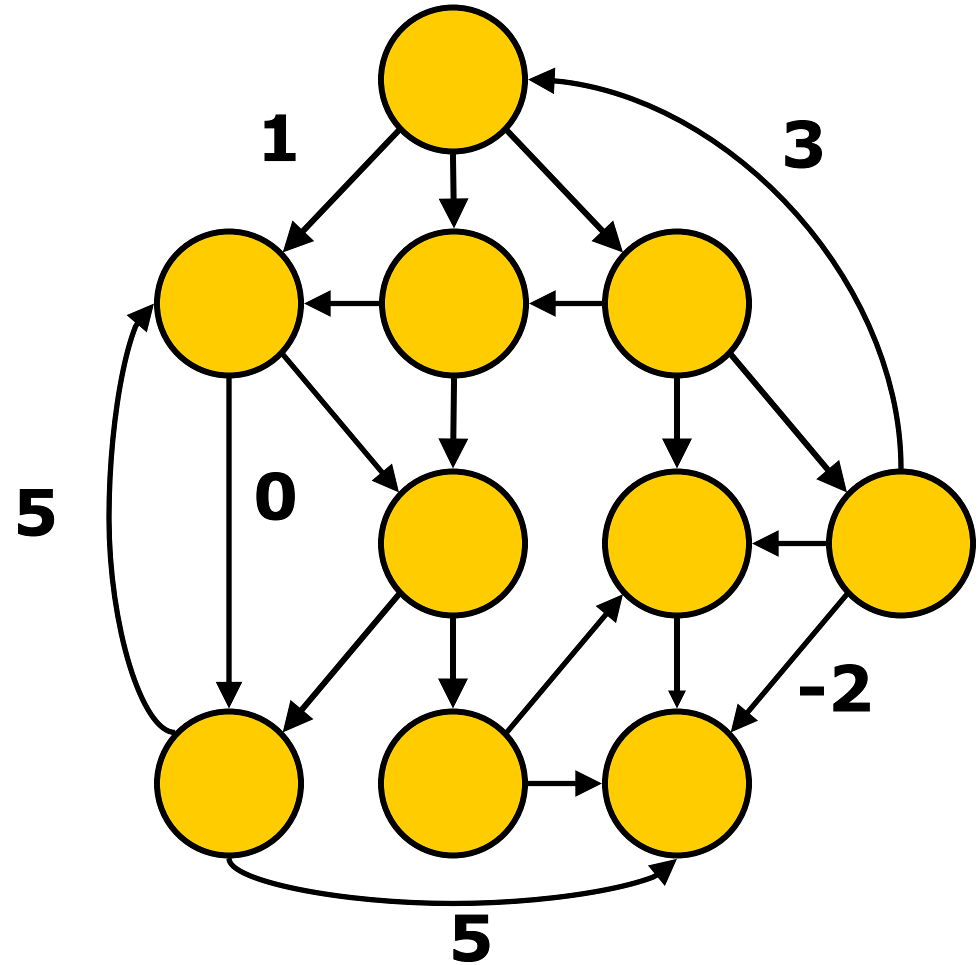
Directed graph



Example: travel planning

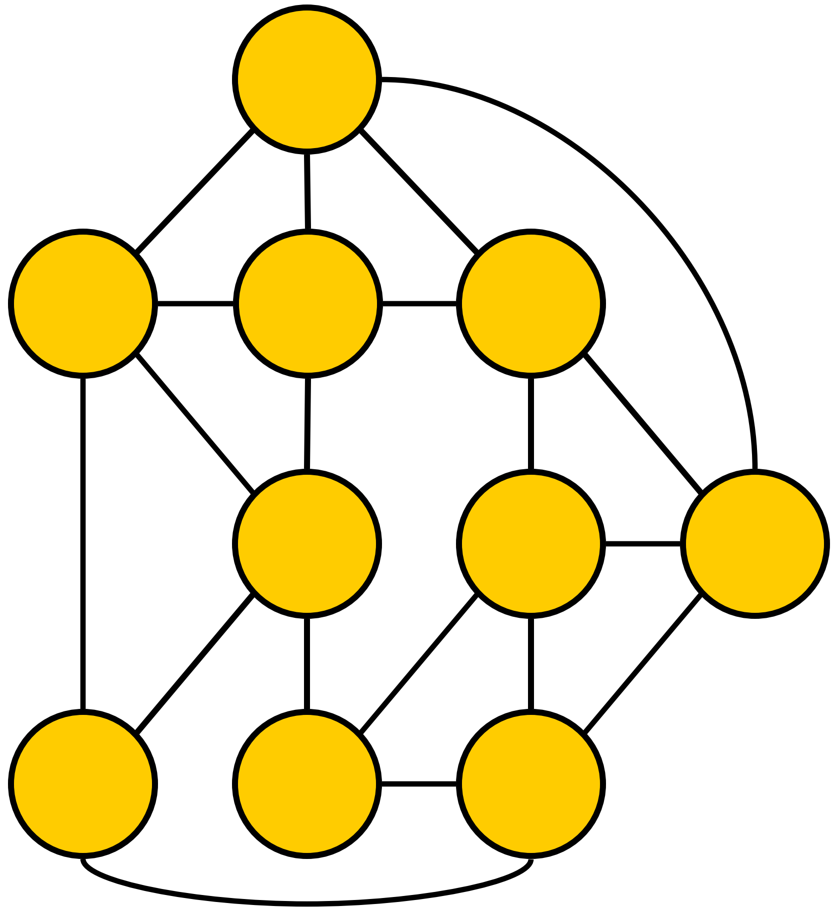


Weighted directed graph



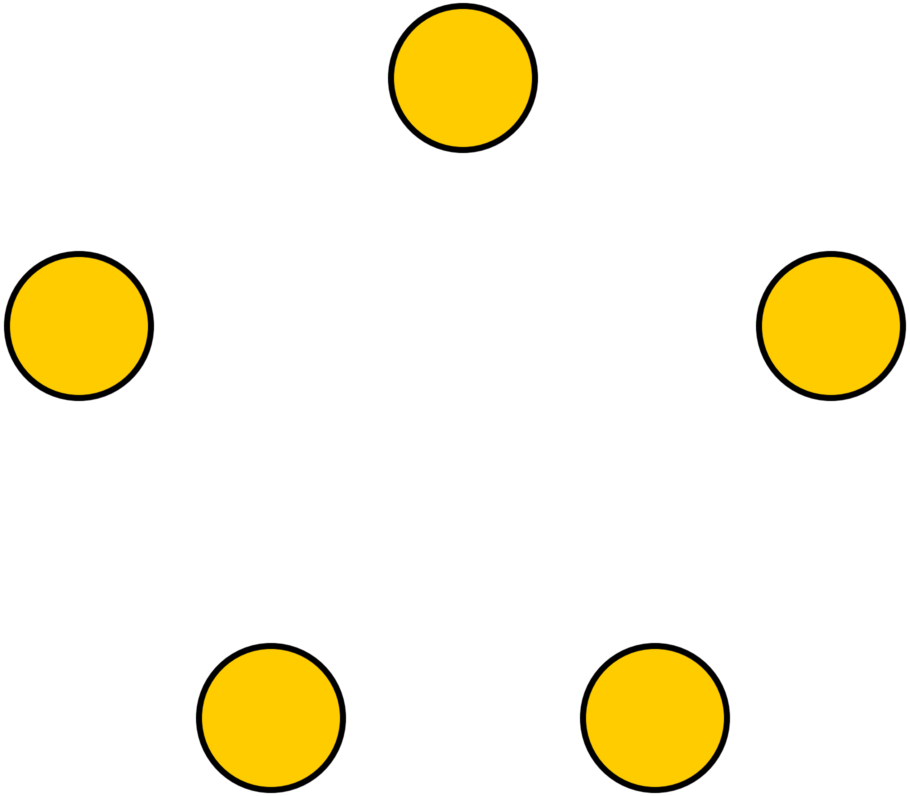
Undirected graph

Ex.: we may want to find the shortest path



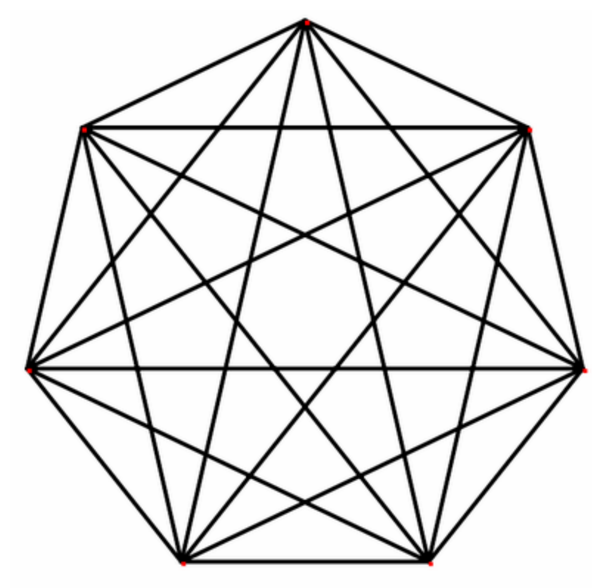
Undirected graph can be weighted Or unweighted

Complete graph



Complete Graph

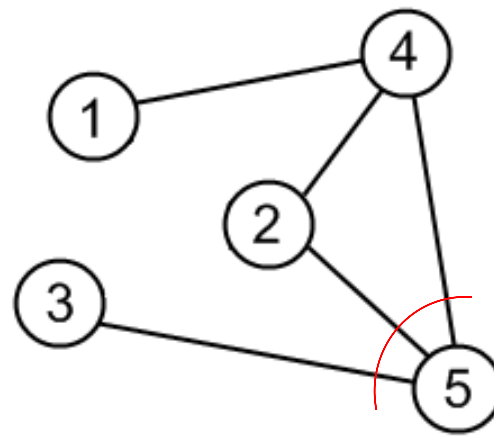
Simple graph with
N vertices and ${}_{N}C_{2}$ edges



Graph Terminologies (2)

More terminologies (simple graph):

- Sparse/Dense
  - Sparse = not so many edges
  - Dense = many edges
  - No guideline for "how many"
- In/Out Degree of a vertex
  - Number of in/out edges from a vertex



Sparse Graph

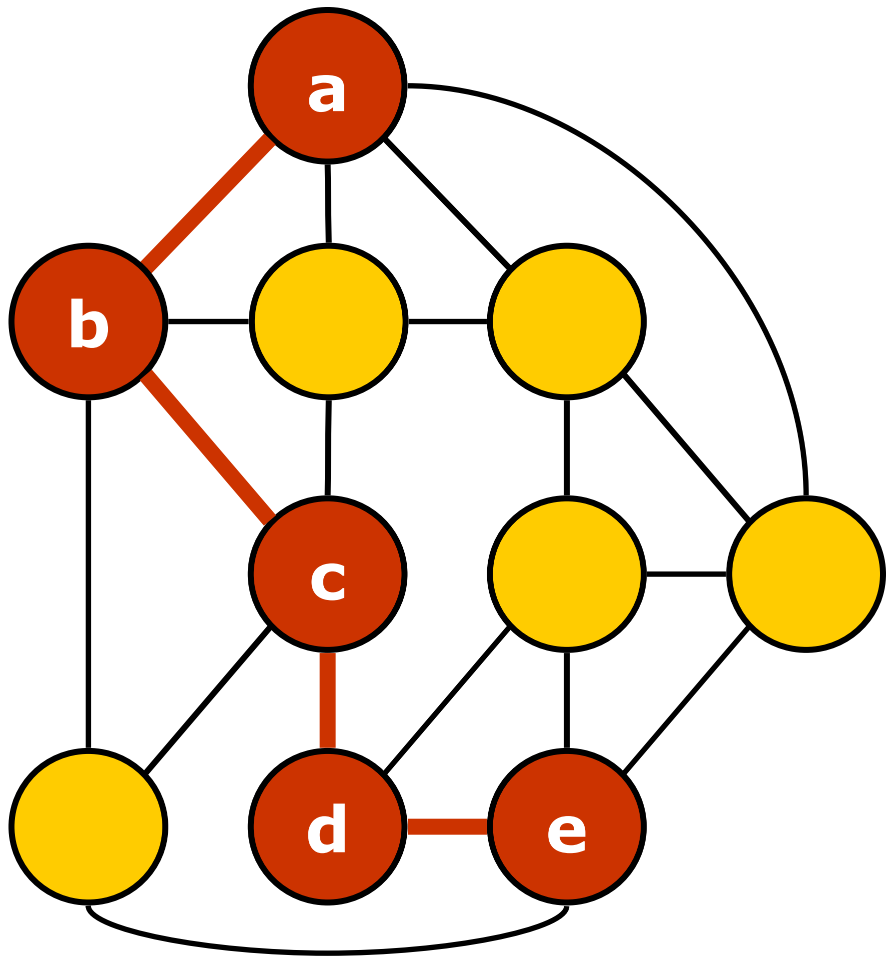
In/out degree of vertex 5 = 3

Dense Graph

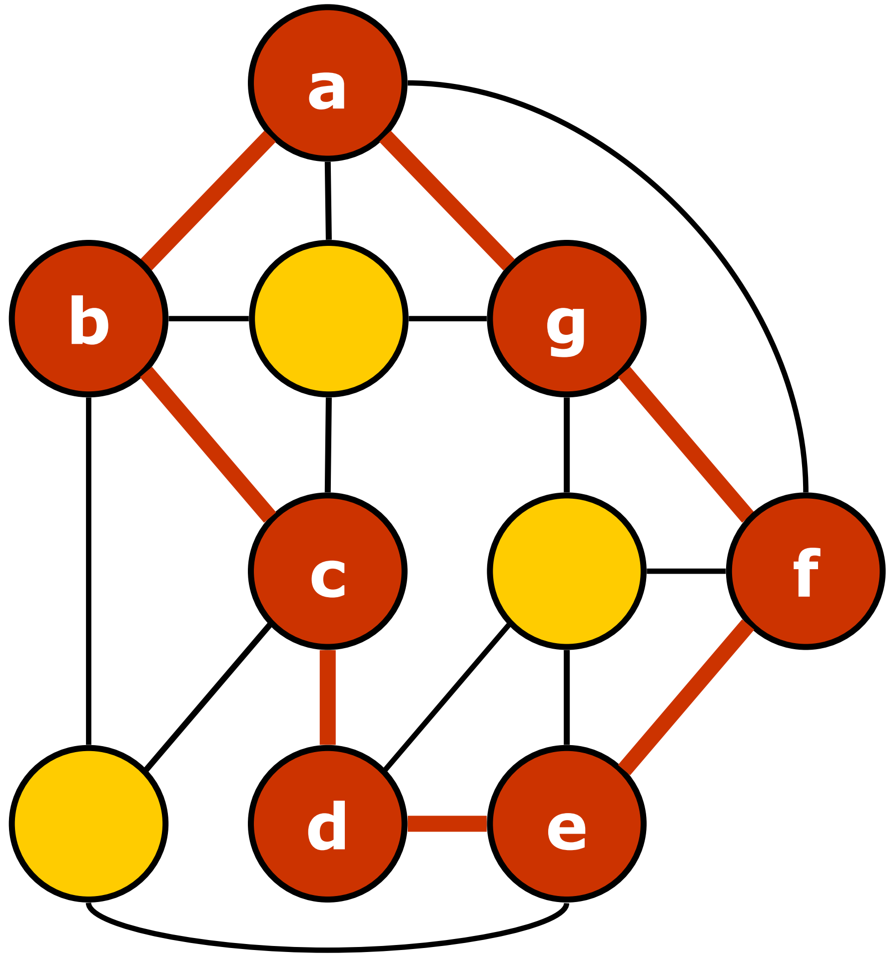
If ≥ n edges: cycle

Complete Graph
7 vertices,
$_7C_2$ = 21 edges

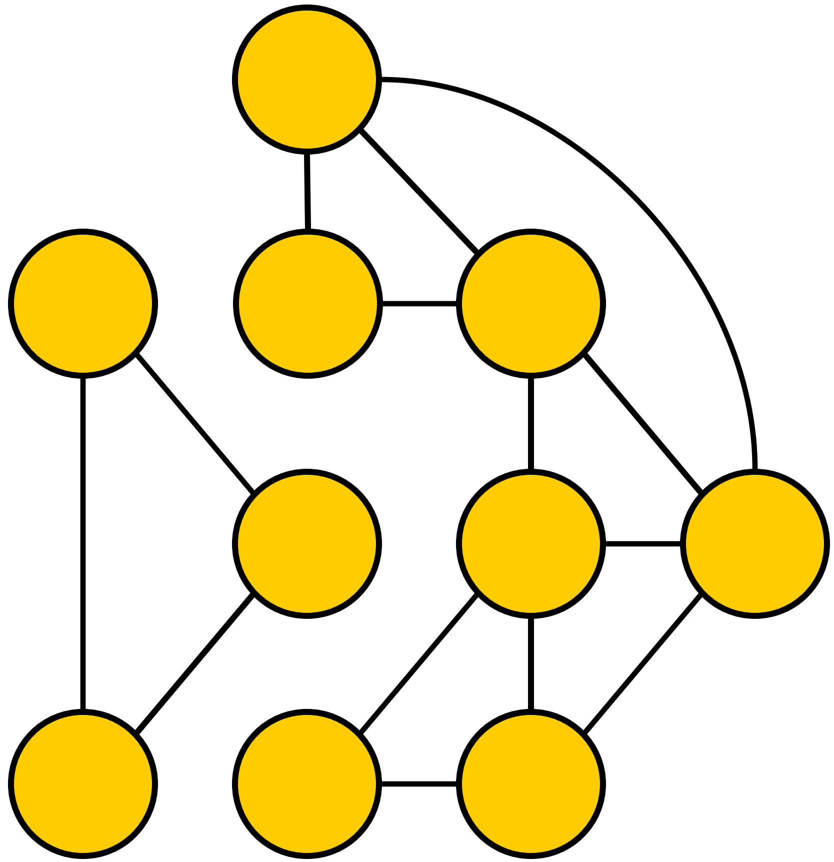
Path



Cycle

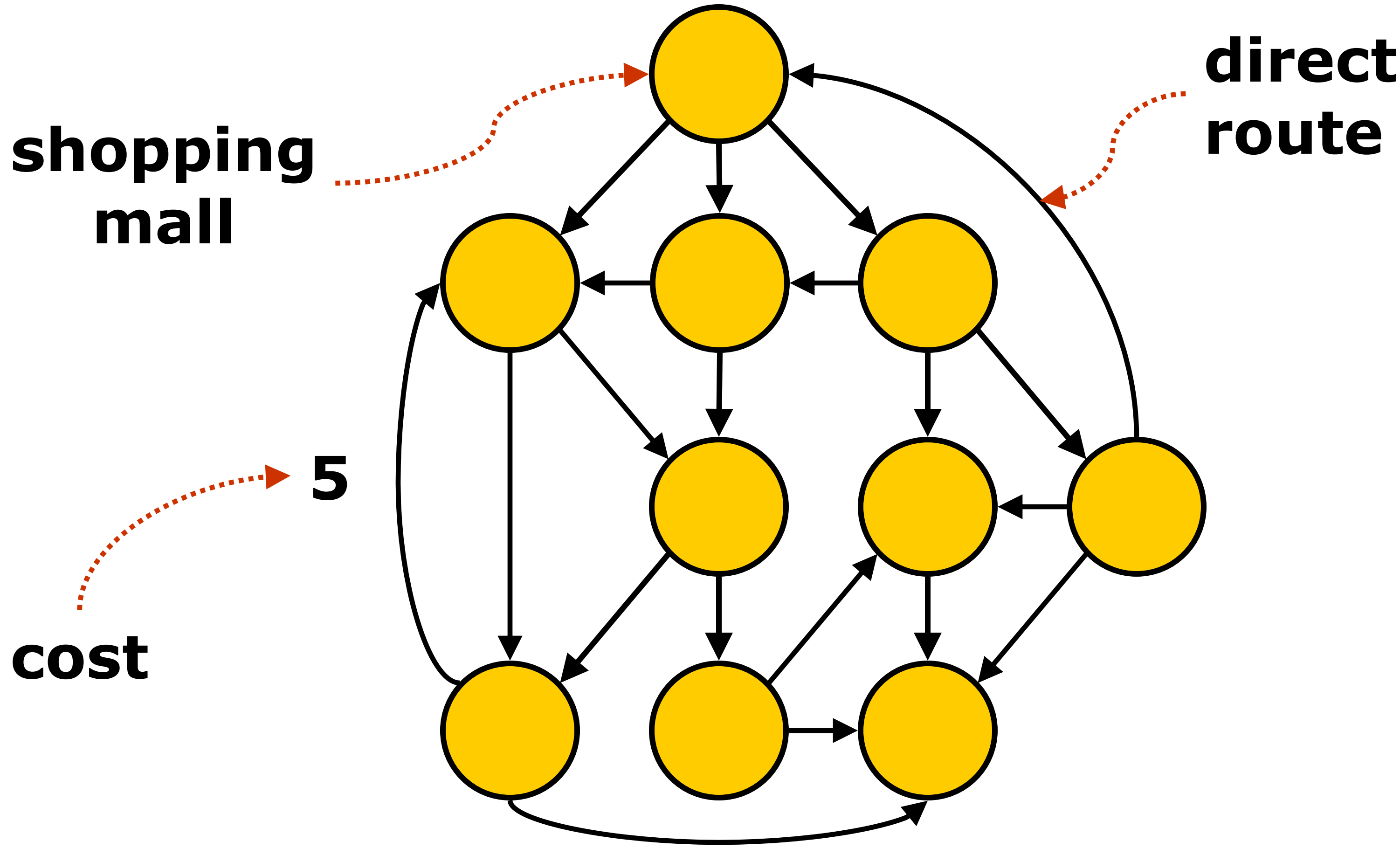


Disconnected graph



Applications

Travel Planning



Question

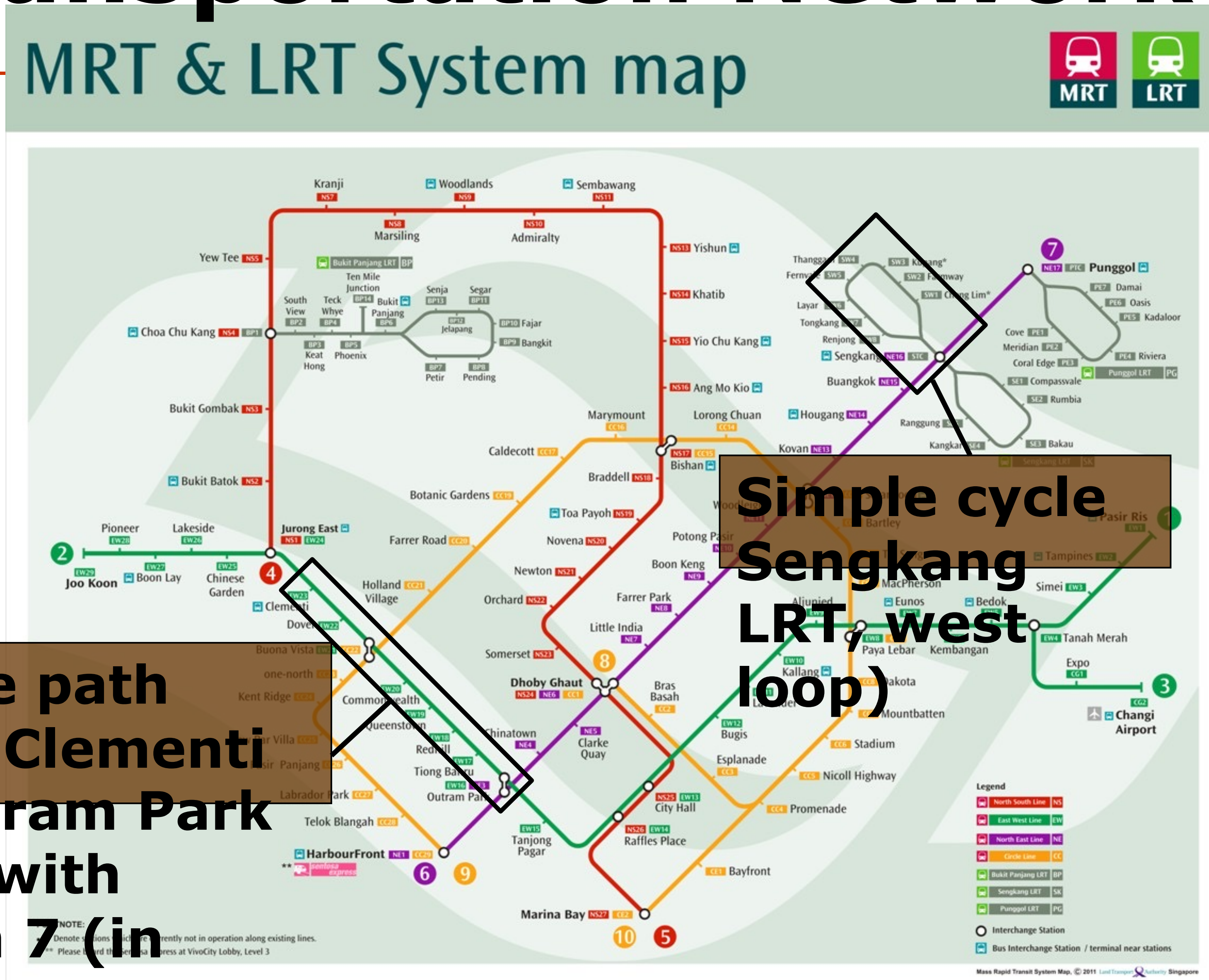
- What is the shortest way to travel between A and B?

  “**SHORTEST PATH PROBLEM**”

- How to minimize the cost of visiting n cities such that we visit each city exactly once, and finishing at the city where we start from?

  “TRAVELING SALESMAN PROBLEM (TSP)”

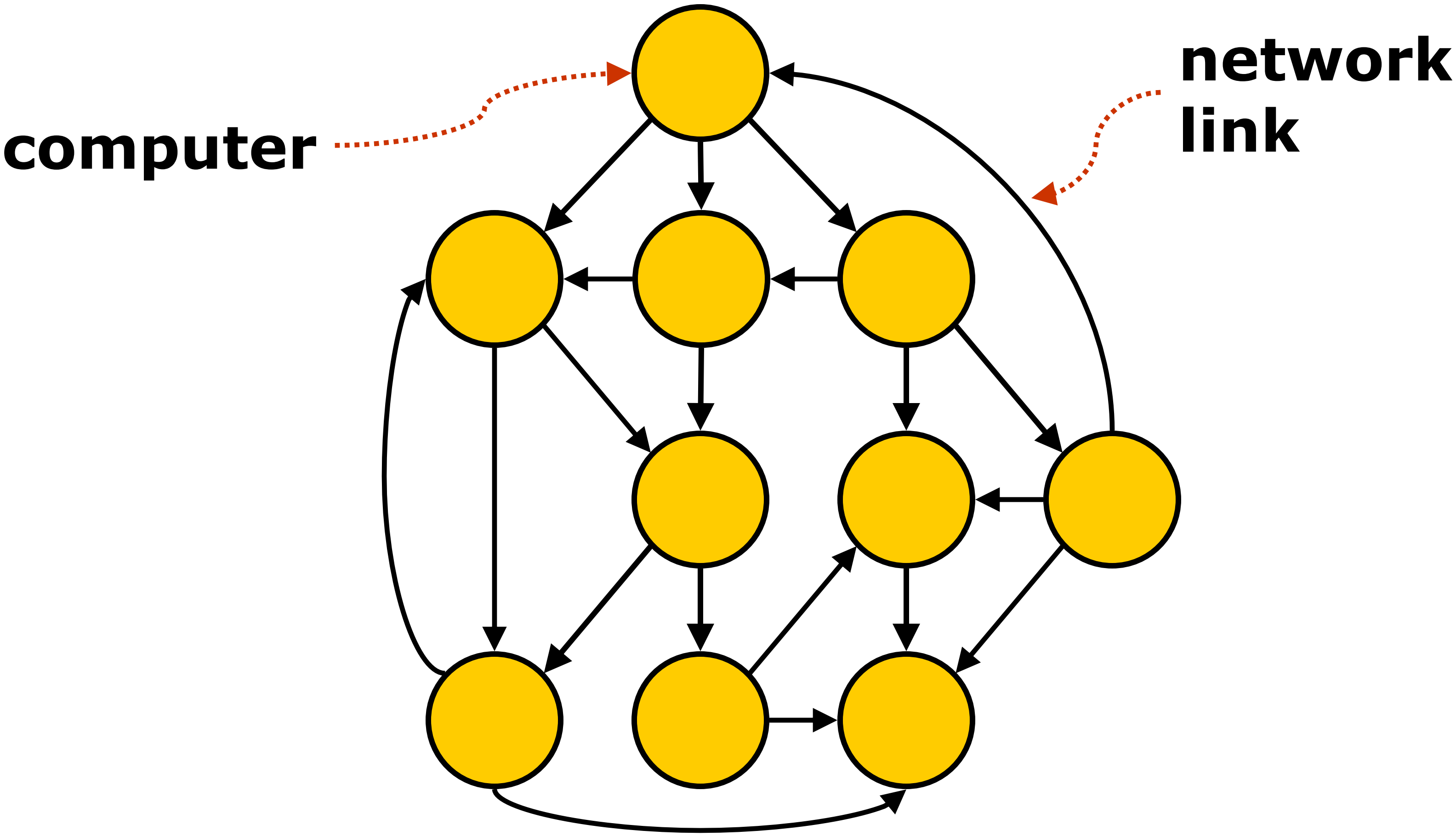
Transportation Network



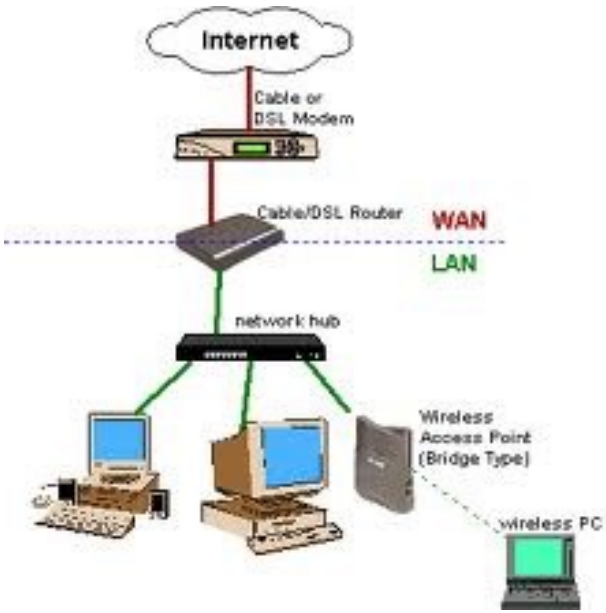
Simple cycle Sengkang LRT, west loop)

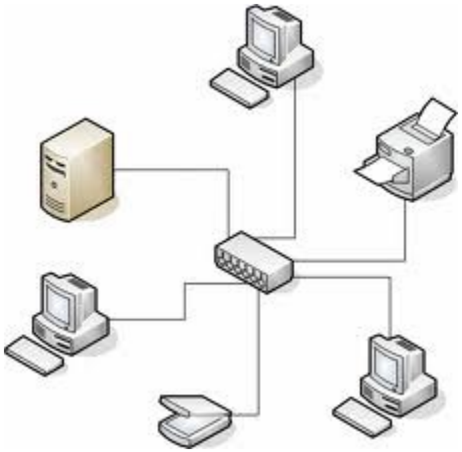
Simple path (from Clementi to Outram Park MRT) with length 7 (in terms of number

Internet

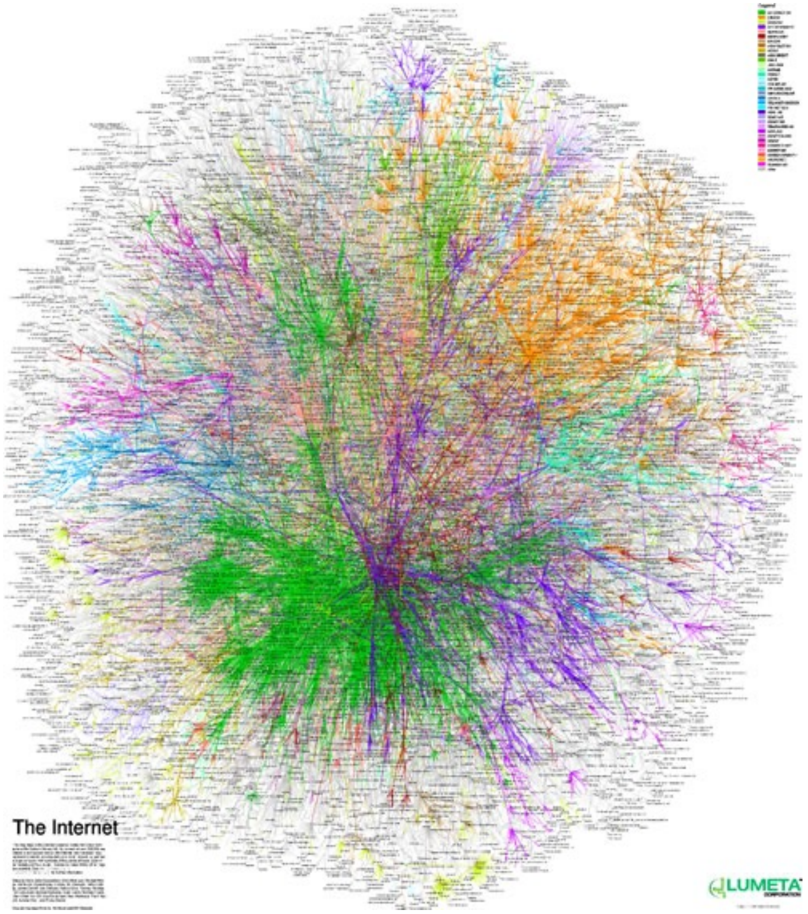


Internet / Computer Networks





Internet / Computer Networks



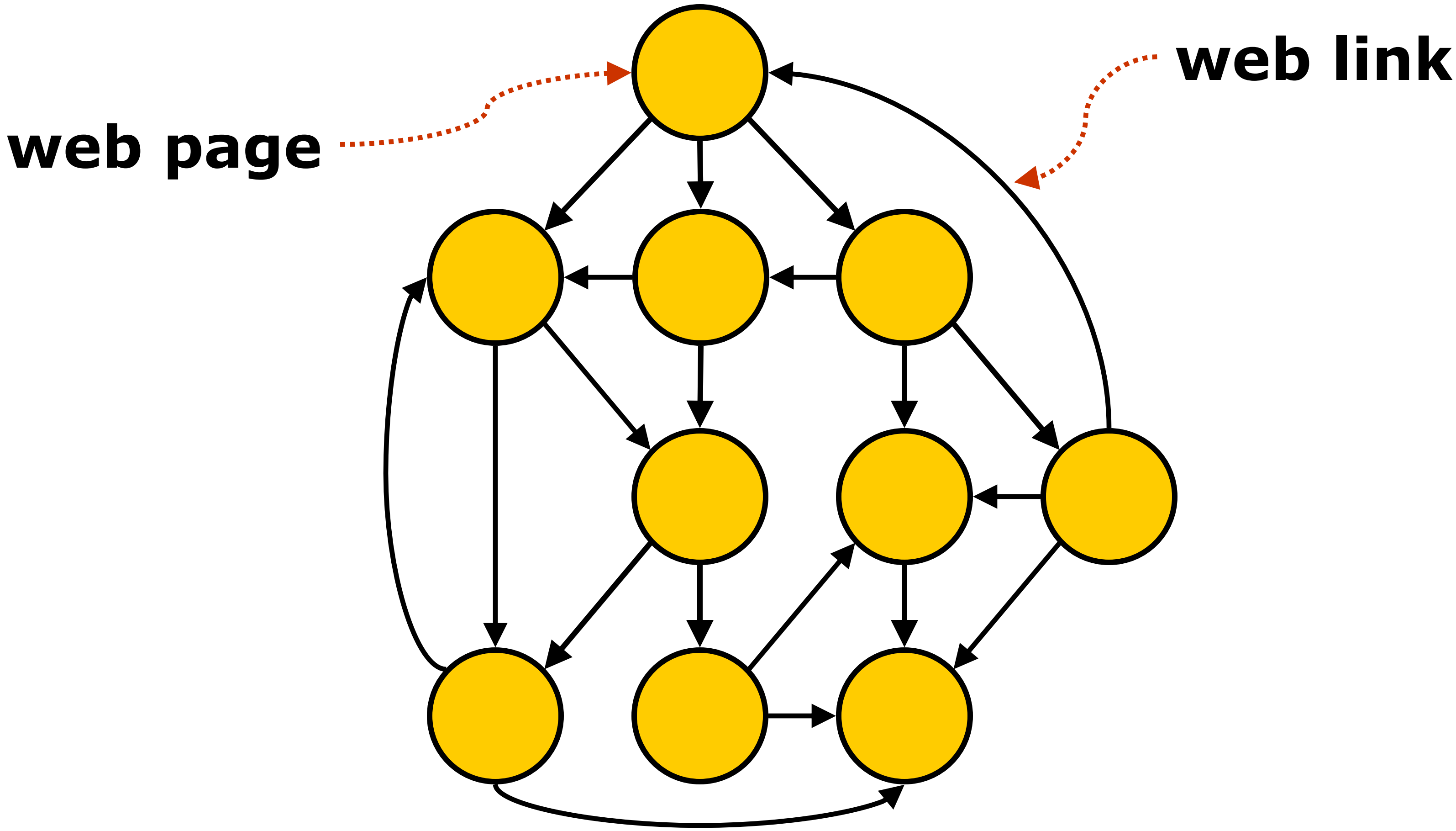
Question

- What is the shortest route to send a packet from A to B?

“**SHORTEST PATH PROBLEM**”

Communication Network

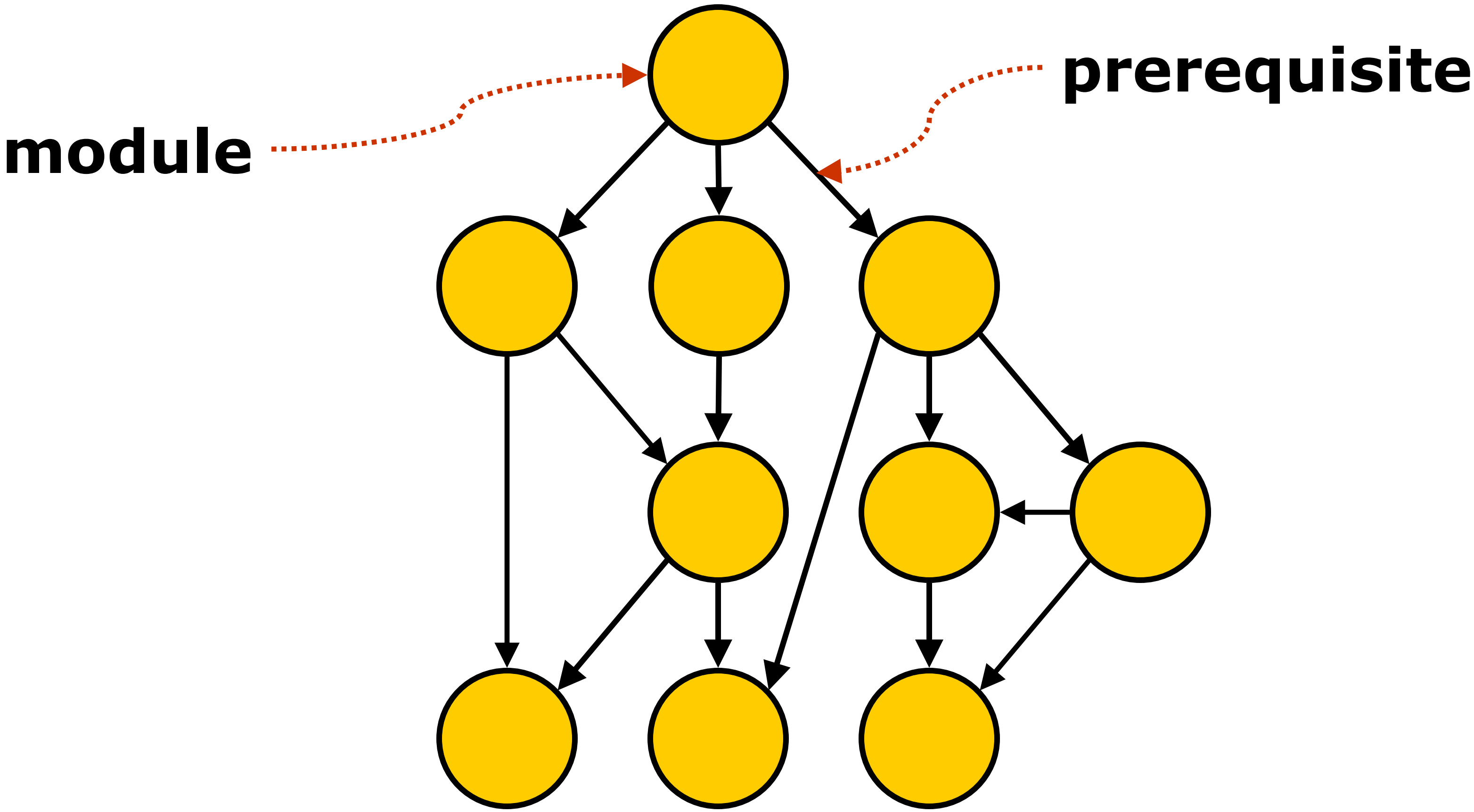
The Web



Question

- Which web pages are important?
- Which set of web pages are likely to be of the same topic?

Module Selection

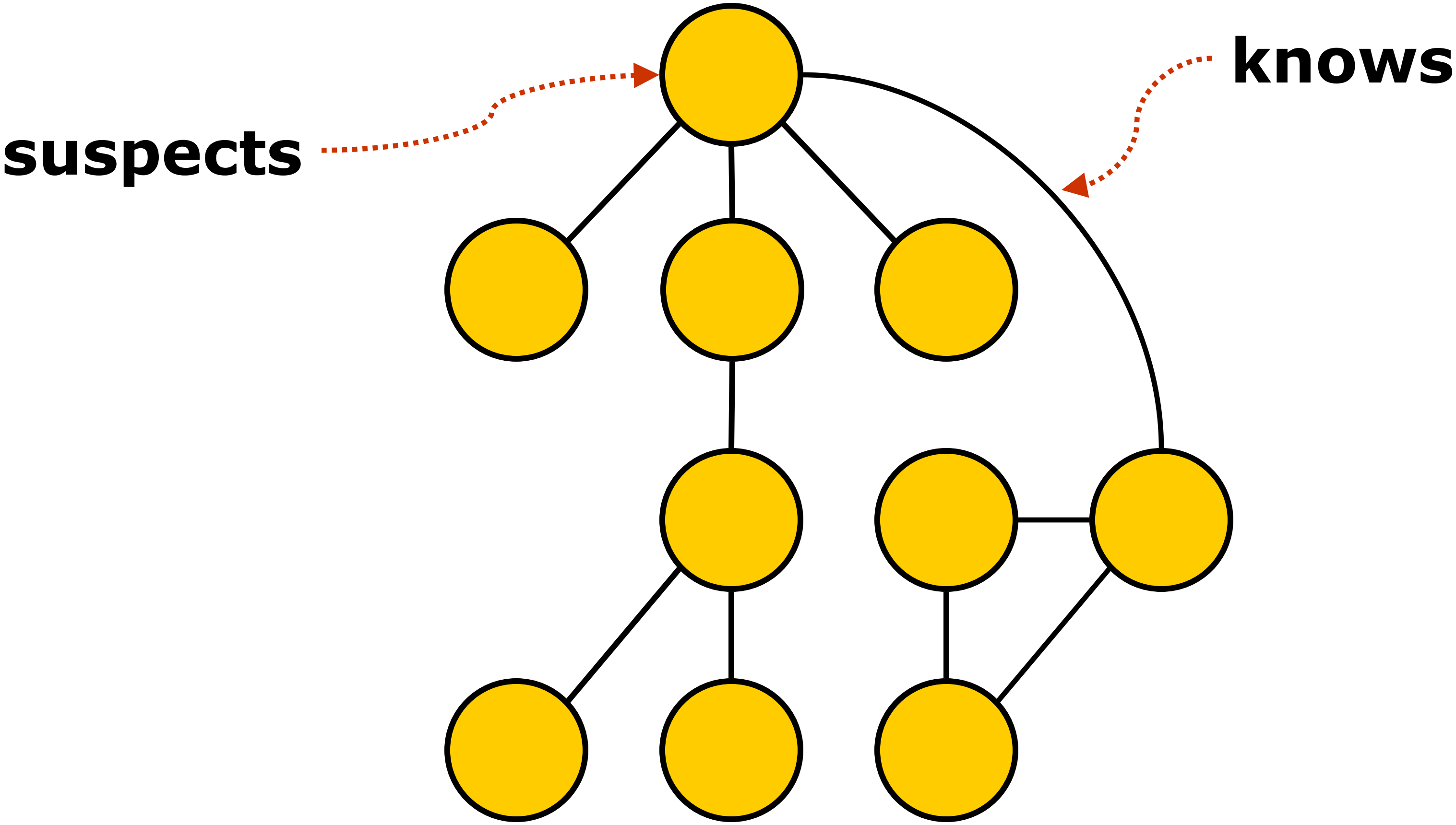


Question

- Find a sequence of modules to take that satisfy the prerequisite requirements.

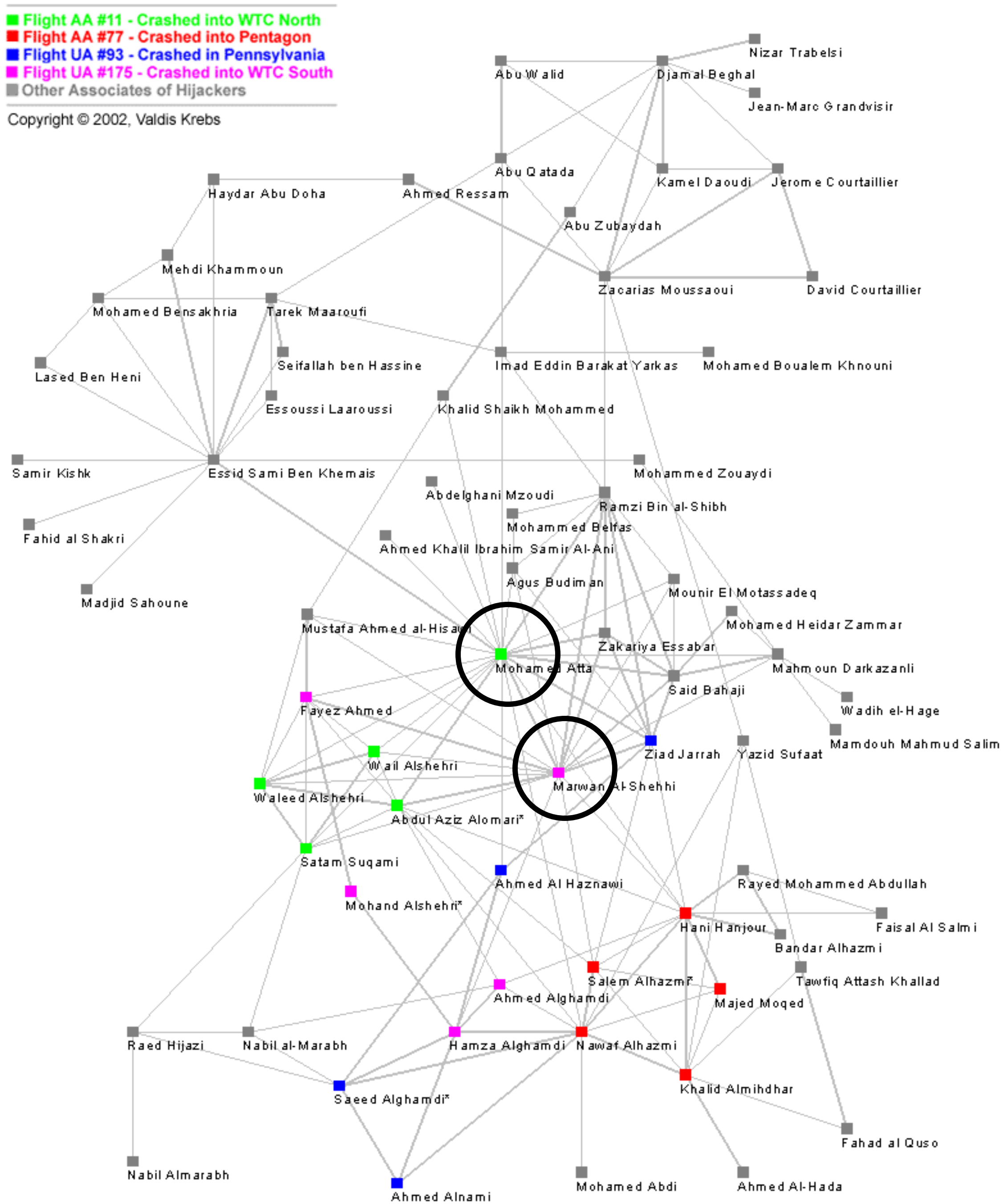
“TOPOLOGICAL SORT”

Terrorist

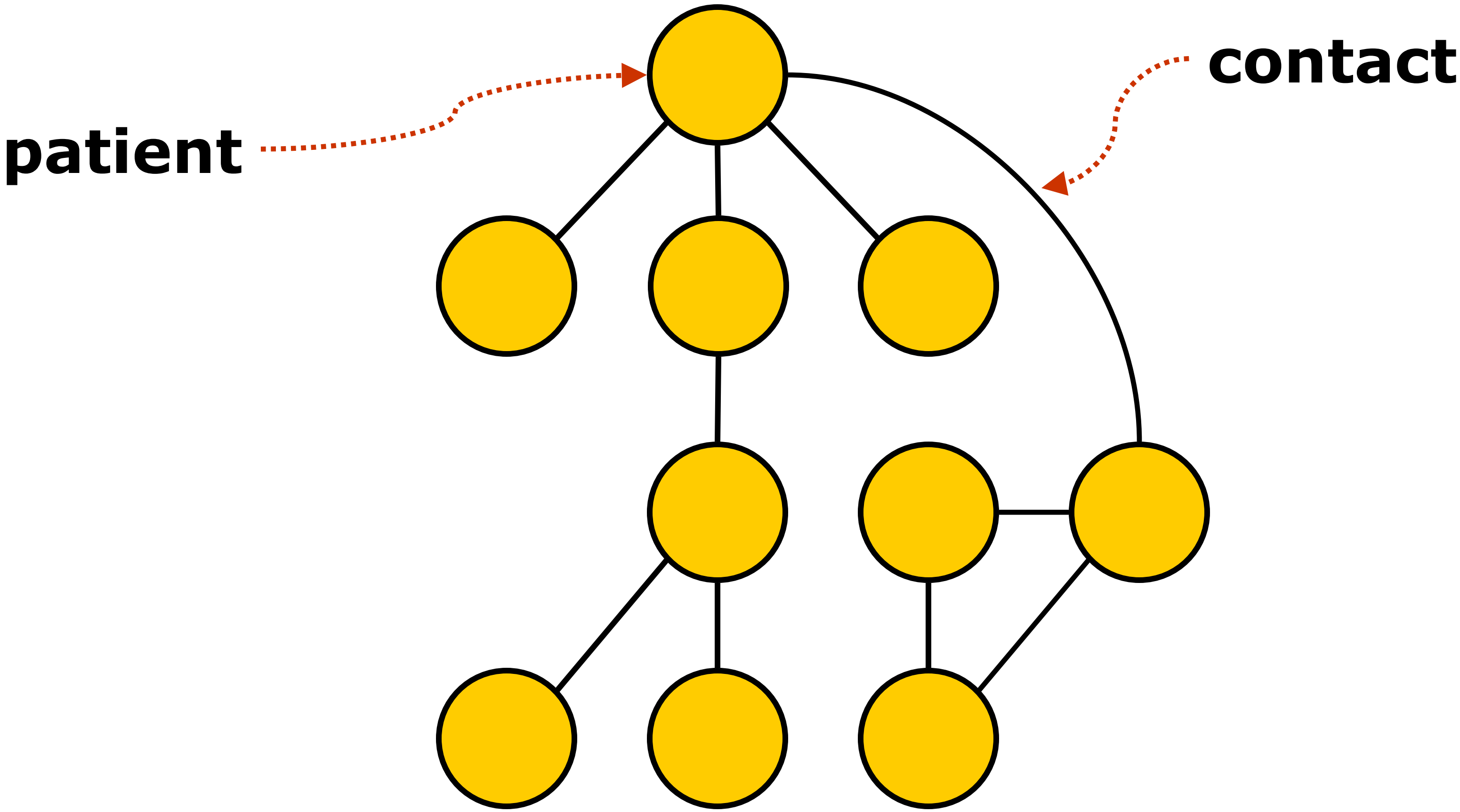


Question

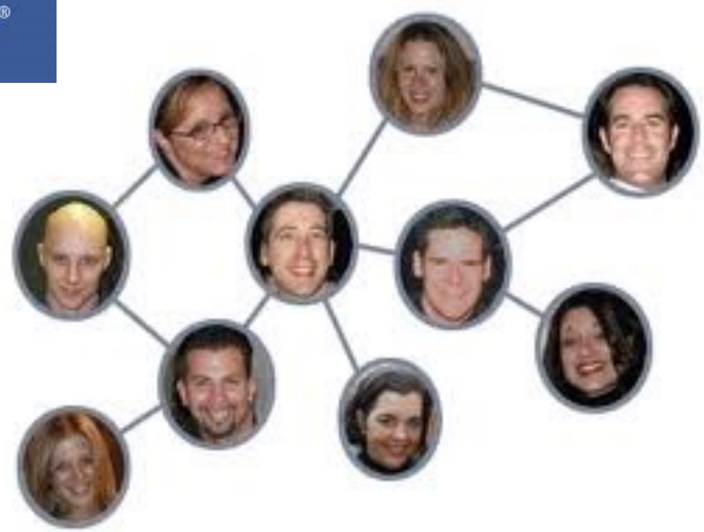
- Who are the important figures in a terrorist network?


Flight AA #11 - Crashed into WTC North
Flight AA #77 - Crashed into Pentagon
Flight UA #93 - Crashed in Pennsylvania
Flight UA #175 - Crashed into WTC South
Other Associates of Hijackers
Copyright © 2002, Valdis Krebs
Nizar Trabelsi
Abu Walid
Djamal Beghal
Jean-Marc Grandvisir
Abu Qatada
Kamel Daoudi
Jerome Courtaillier
Haydar Abu Doha
Ahmed Ressam
Abu Zubaydah
Mehdi Khammoun
Zacarias Moussaoui
David Courtaillier
Mohamed Bensakhria
Tarek Maaroufi
Lased Ben Heni
Seifallah ben Hassine
Imad Eddin Barakat Yarkas
Mohamed Boualem Khnouni
Essoussi Laaroussi
Khalid Shaikh Mohammed
Samir Kishk
Essid Sami Ben Khemais
Mohammed Zouaydi
Abdelghani Mzoudi
Ramzi Bin al-Shibh
Fahid al Shakri
Mohammed Belfas
Ahmed Khalil Ibrahim Samir Al-Ani
Agus Budiman
Madjid Sahoune
Mounir El Motassadeq
Mustafa Ahmed al-Hisawi
Mohamed Heidar Zammar
Zakariya Essabar
Mohamed Atta
Mahmoun Darkazanli
Said Bahaji
Fayez Ahmed
Wadih el-Hage
Mamdouh Mahmud Salim
Ziad Jarrah
Yazid Sufaat
Wail Alshehri
Marwan Al-Shehhi
Waleed Alshehri
Abdul Aziz Alomari*
Satam Suqami
Ahmed Al Haznawi
Rayed Mohammed Abdullah
Mohand Alshehri*
Hani Hanjour
Faisal Al Salmi
Bandar Alhazmi
Salem Alhazmi*
Tawfiq Attash Khallad
Ahmed Alghamdi
Majed Moqed
Raed Hijazi
Nabil al-Marabh
Hamza Alghamdi
Nawaf Alhazmi
Khalid Almihdhar
Saeed Alghamdi*
Fahad al Quso
Nabil Almarabh
Ahmed Alnami
Mohamed Abdi
Ahmed Al-Hada

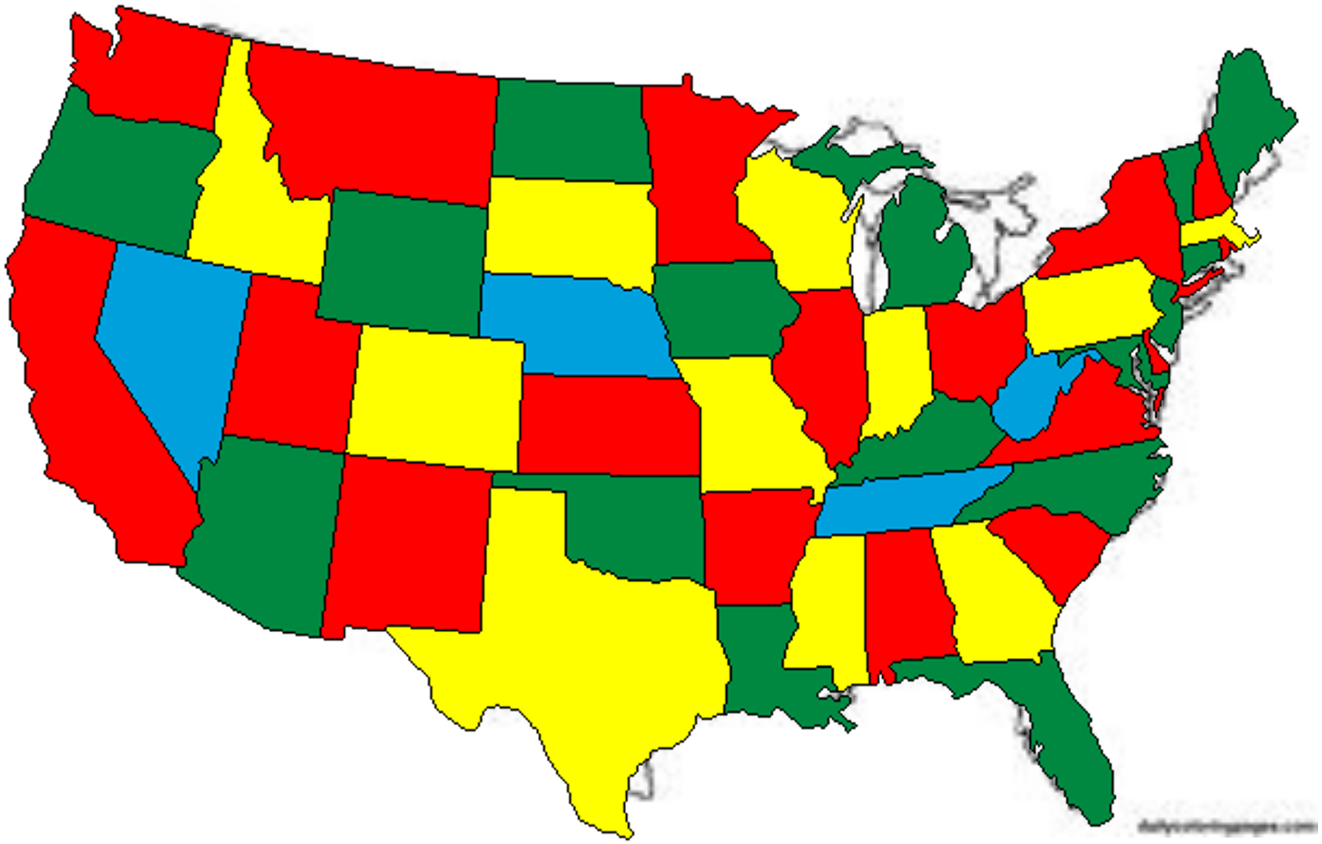
Epidemic Studies



Social Network



Optimization



Other applications

- Biology
- VLSI layout
- Vehicle routing
- Job scheduling
- Facility location

:

:

Implementation

Formally

A graph $G = (V, E, w)$, where

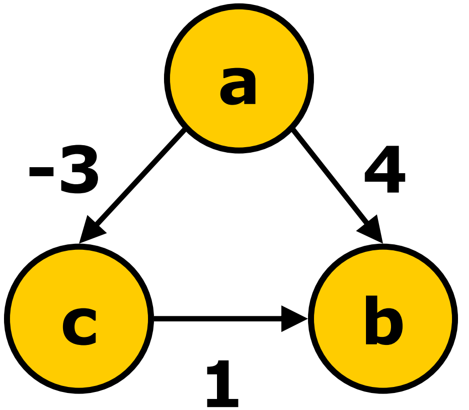
- V is the set of vertices
- E is the set of edges
- w is the weight function

Example

V = { a, b, c }

E = { (a,b), (c,b), (a,c) }

w = { ((a,b), 4), ((c, b), 1), ((a,c),-3) }



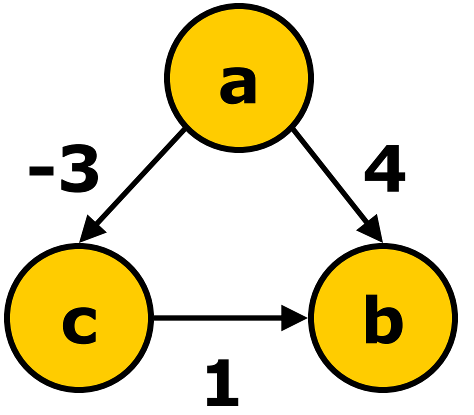
Adjacent vertices

- **adj(v)** = set of vertices adjacent to v

  adj(a) = {b, c}

  adj(b) = {}

  adj(c) = {b}



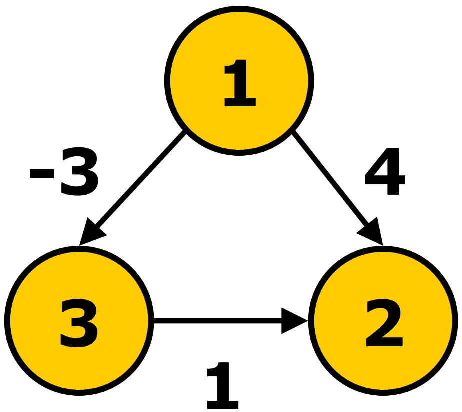
- $\sum_v |adj(v)| = |E|$

- adj(v): Neighbours of v

Adjacency matrix

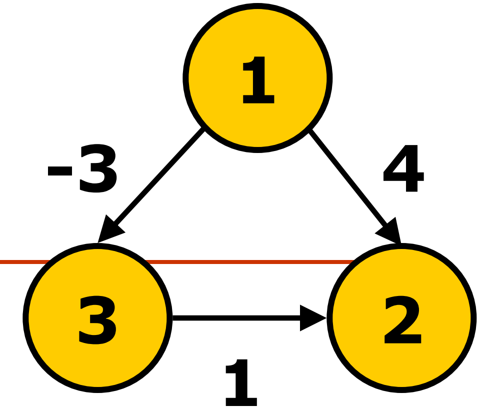
double vertex[][];

	1	2	3
1			
2			
3			

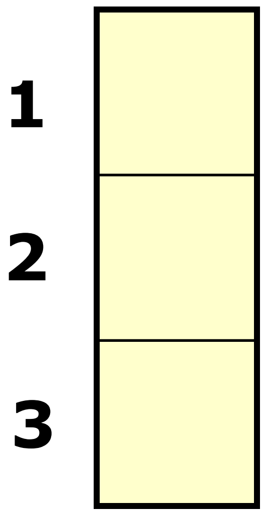


Space Complexity: $O(V^2)$
V is |V| = number of vertices in G

Edge List



EdgeList vertex[];



Format: array **EdgeList** of **E** edges

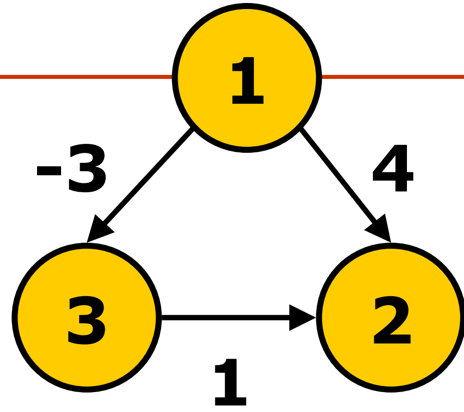
For each edge **i**, **EdgeList[i]** stores an (integer) triple {u, v, w(u, v)}

- For unweighted graph, the weight can be stored as 0 (or 1),
  or simply store an (integer) pair

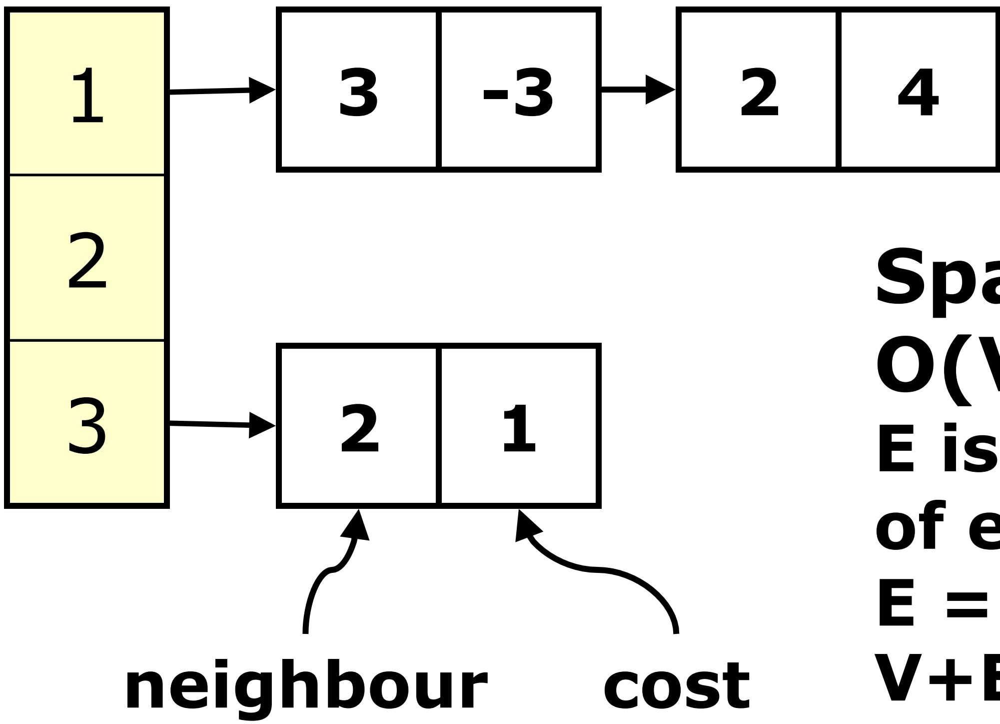
Space Complexity: **O(E)**

- Remember,
  $E = O(V^2)$

Adjacency list



EdgeList vertex[];



Space Complexity:
O(V+E)

E is |E| = number of edges in G,

$E = O(V^2)$

V+E ~= max(V, E)

Vertex map