CE1007/CZ1007 DATA STRUCTURES

Lecture 11: **Graph**

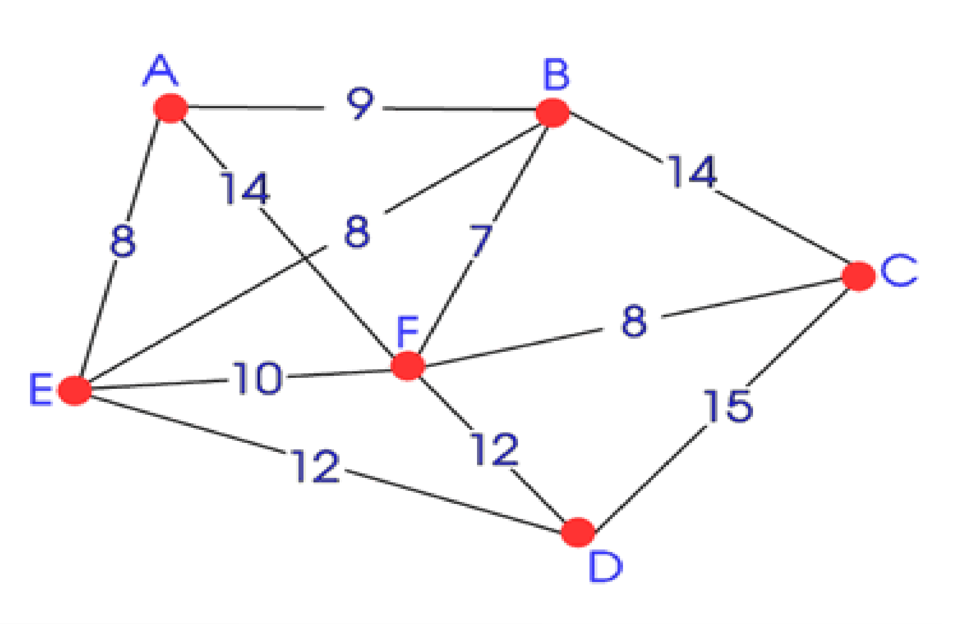
Dr. Owen Noel Newton Fernando

College of Engineering

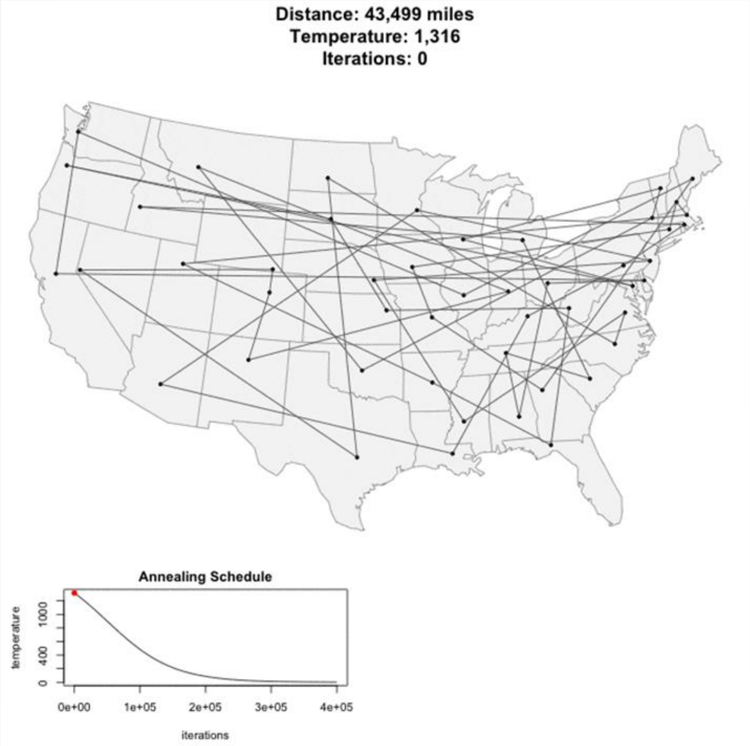
School of Computer Science and Engineering

TRAVELING SALESMAN PROBLEM (TSP)

Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?



Finding the shortest path through the 48 state capitals of the contiguous United States:

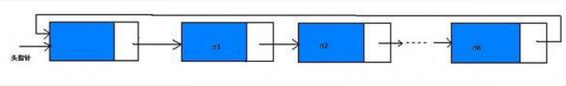


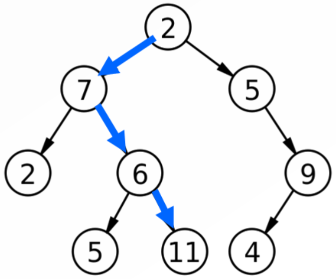
http://toddwschneider.com/posts/traveling-salesman-with-simulated-annealing-r-and-shiny/

OUTLINE

- Graph data structure
- Graph Implementation
- Graph traversal & Path search
- Application background

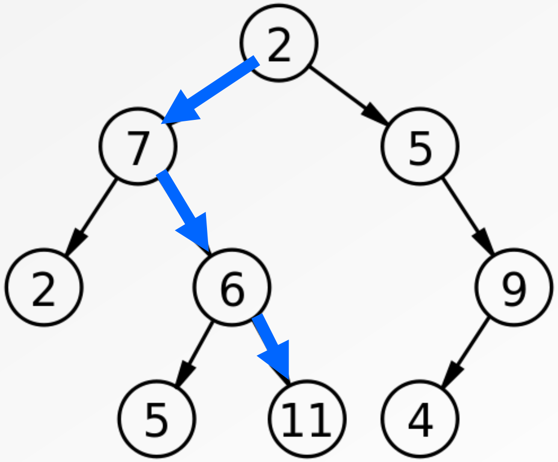
- Linear
  - Items all arranged one after another
  - Random access
    - Arrays
  - Sequential access
    - Linked list
  - Limited-access sequential
    - Stacks
    - Queues
- Non-linear
  - Tree
  - **Graph**



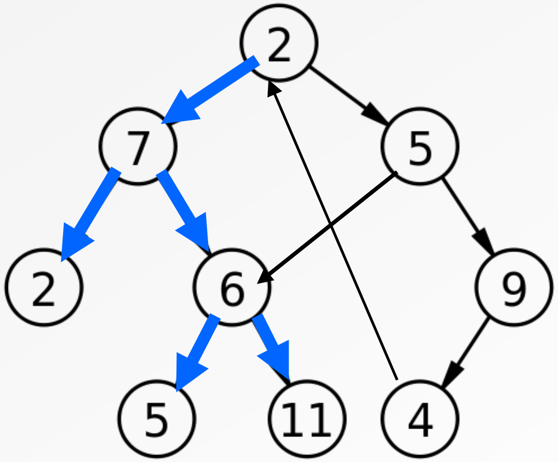


RECALL: TREE DATA STRUCTURE

- Still using nodes + links representation
- New idea:
  - Each node can have links to more than one other node
  - No circle



- Still using nodes + links representation
- New idea:
  - Each node can have links to more than one other node
  - **Circles are allowed, links can exist between any two nodes.**
  - **Links can be directed(one-way) or undirected(two-way)**



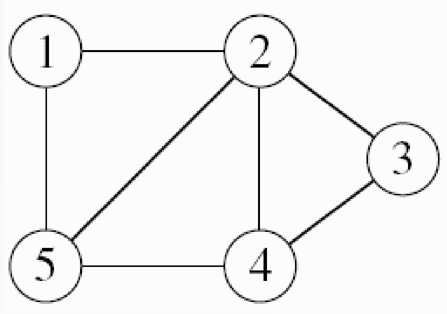
Trees are special cases of graph

WHAT IS A GRAPH?

- A data structure that consists of a set of nodes (vertices) and a set of edges (links) that relate the nodes to each other
  - G = (V, E)
  - where V is a set whose elements are called vertices, and
  - E is a set of ordered pairs of elements of V.
- The set of edges describes relationships among the vertices

GRAPH TERMINOLOGY

- Adjacent nodes: two nodes are adjacent if they are connected by an edge
- Degree of a node i: how many adjacent nodes of i
- Path: a sequence of vertices that connect two nodes in a graph
- Connected graph: there is a path from any node to any other node in the graph
- Complete graph: a graph in which every vertex is directly connected to every other vertex



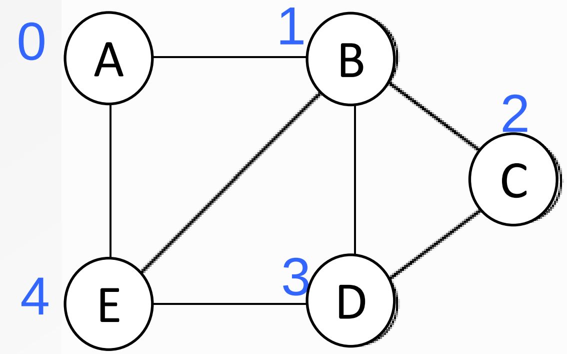
1 is adjacent to 2 and 5;
Degree of node 1 is 2;
A path from 1 to 4: 1->2->3->4;
This is a connected graph;
This is not a complete graph;

OUTLINE

- Graph data structure
- Graph Implementation
- Graph traversal & Path search
- Application background

GRAPH IMPLEMENTATION

- **Array-based implementation**
  - A 1D array is used to represent the vertices
  - A 2D array (adjacency matrix) is used to represent the edges



[0]	A
[1]	B
[2]	C
[3]	D
[4]	E

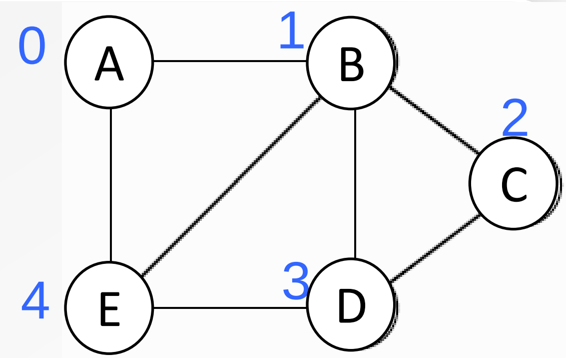
Store vertices' data

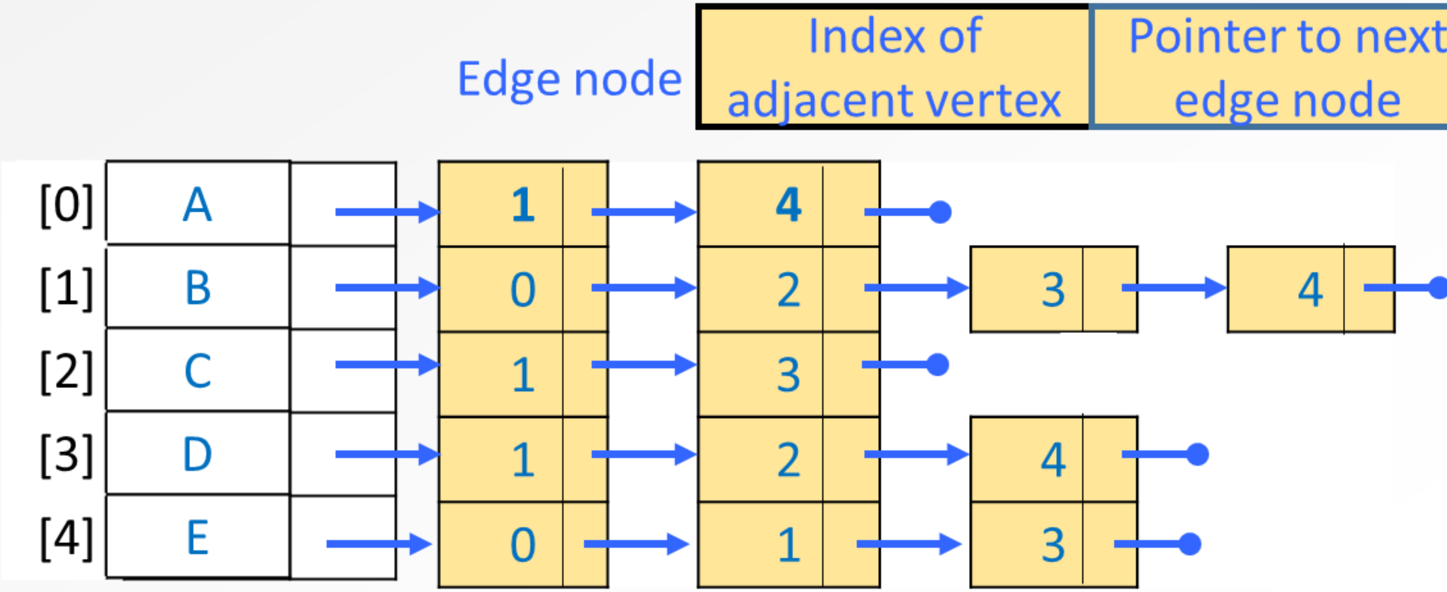
	[0]	[1]	[2]	[3]	[4]
[0]	0	1	0	0	1
[1]	1	0	1	1	1
[2]	0	1	0	1	0
[3]	0	1	1	0	1
[4]	1	1	0	1	0

Adjacency matrix

Linked-list-based implementation

- A 1D array is used to represent the vertices
- A list is used for each vertex v which contains the vertices which are adjacent from v (adjacency list)





Store vertices' data

Adjacency list

ADJACENCY MATRIX VS. ADJACENCY LIST REPRESENTATION

- **Adjacency matrix**
  - Good for dense graphs --|E|~O(|V|2)
  - Memory requirements: O(|V| + |E|) = O(|V|2)
  - Connectivity between two vertices can be tested quickly
- **Adjacency list**
  - Good for sparse graphs -- |E|~O(|V|)
  - Memory requirements: O(|V| + |E|)=O(|V|)
  - Vertices adjacent to another vertex can be found quickly

OUTLINE

- Graph data structure
- Graph Implementation
- Graph traversal & Path search
- Application background

The traversal problem: check all nodes once and only once.

- Graph Traversal
  - List traversal: easy
  - Binary Tree traversal: In-order, Pre-order, Post-order, level-by-level
  - Graph traversal: ?

BREADTH-FIRST SEARCH (BFS)

- Similar to level-by-level Traversal of a Binary Tree
  Use a queue
- What is the idea behind BFS?
  - Look at all possible paths at the same depth before you go at a deeper level
- If the graph is connected:
  - Choose a starting vertex
  - Search all adjacent vertices
  - Return to each adjacent vertex in turn and visit all of its adjacent vertices

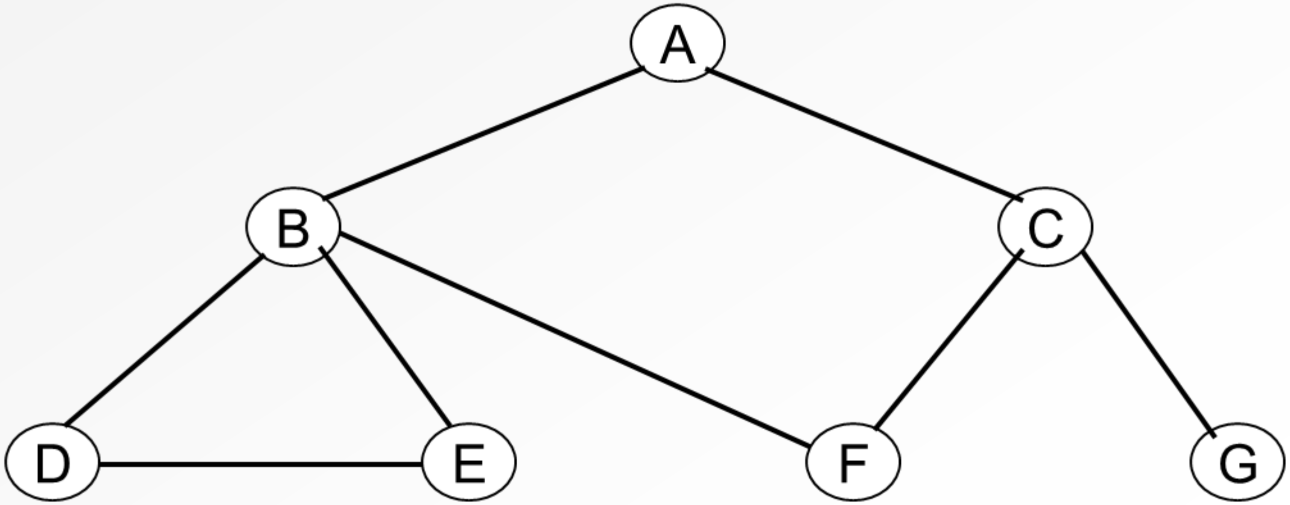
BREADTHFIRSTSEARCH()

```
breadthFirstSearch(gNode cur, graph g)
{
  queue q;
  initialize q;
  mark cur as visited;
  enqueue(&q, cur);
  while (!emptyQueue(&q)){
    cur= dequeue(&q);
    for all unvisited vertices j adjacent to cur{
       mark j as visited;
       enqueue(&q, j);
    }
  }
}
```

ANOTHER TRAVELING PROBLEM

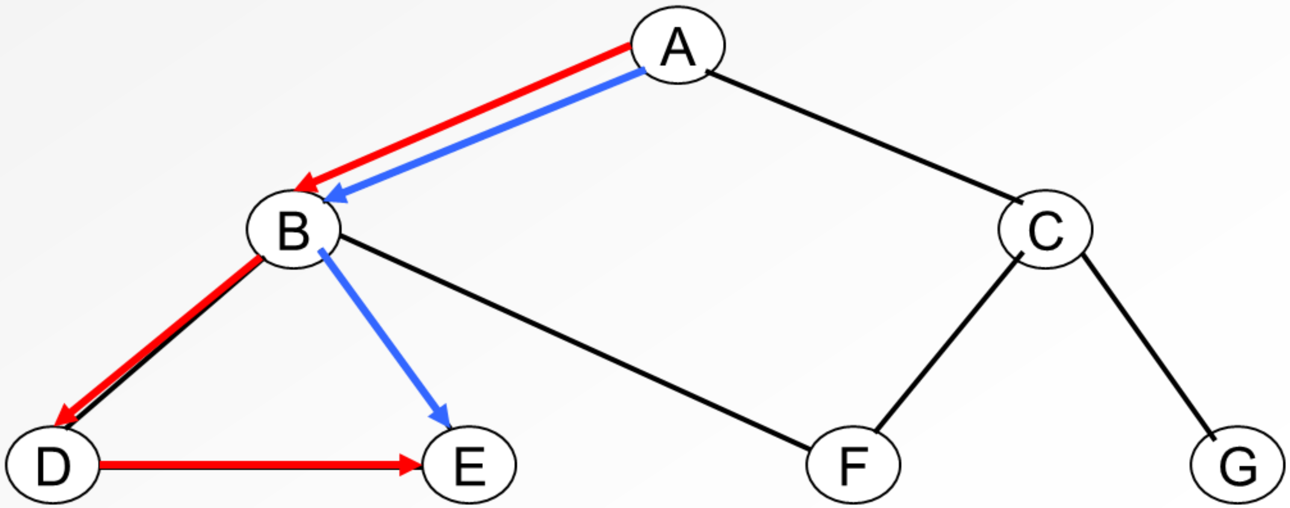
Given a list of cities and highways among them, what is the driving schedule from city A to E?

This is a path searching problem



PATH SEARCHING

- Problem: find a path between two nodes of the graph (e.g., from A to E), nodes in a path are distinct.
- Method: Breadth-First-Search (BFS)
- Use BFS to start search from A, until E is reached.



SHORTEST-PATH

- There are multiple paths from a source vertex to a destination vertex
- *<u>Shortest path</u>*: the path with minimum edges.
- Path found by BFS is the shortest path. (why?)
- Back to the travelling salesman problem:
  - Much difficult than finding a shortest path
  - Find a shortest path from A to A, visiting every nodes once and only once.
  - You need to learn more (local search, algorithm) to solve it

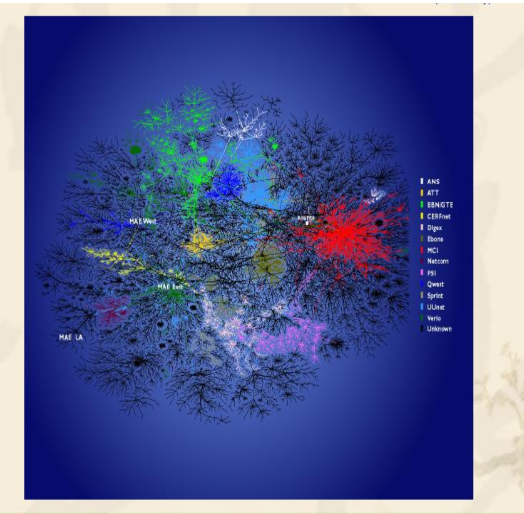
https://www.youtube.com/watch?v=SC5CX8drAtU

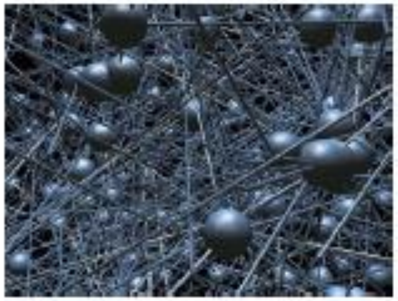
OUTLINE

- Graph data structure
- Graph Implementation
- Graph traversal & Path search
- Application background

GRAPH (NETWORK) IS EVERYWHERE!

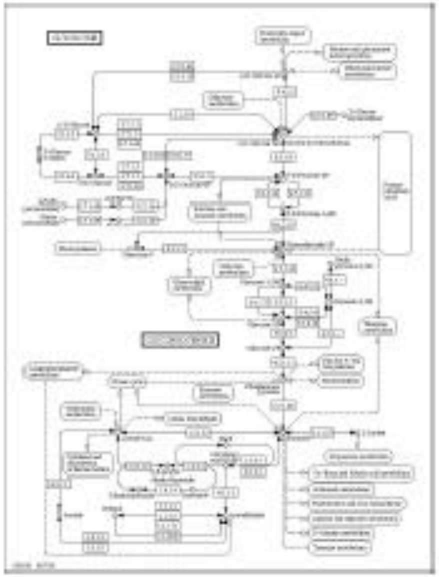
Internet





A social network

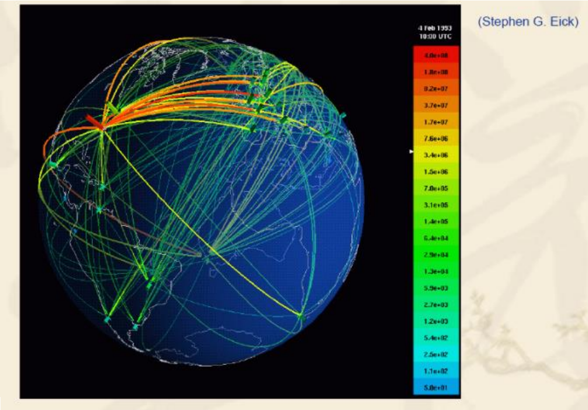
A Neural network



A metabolic network

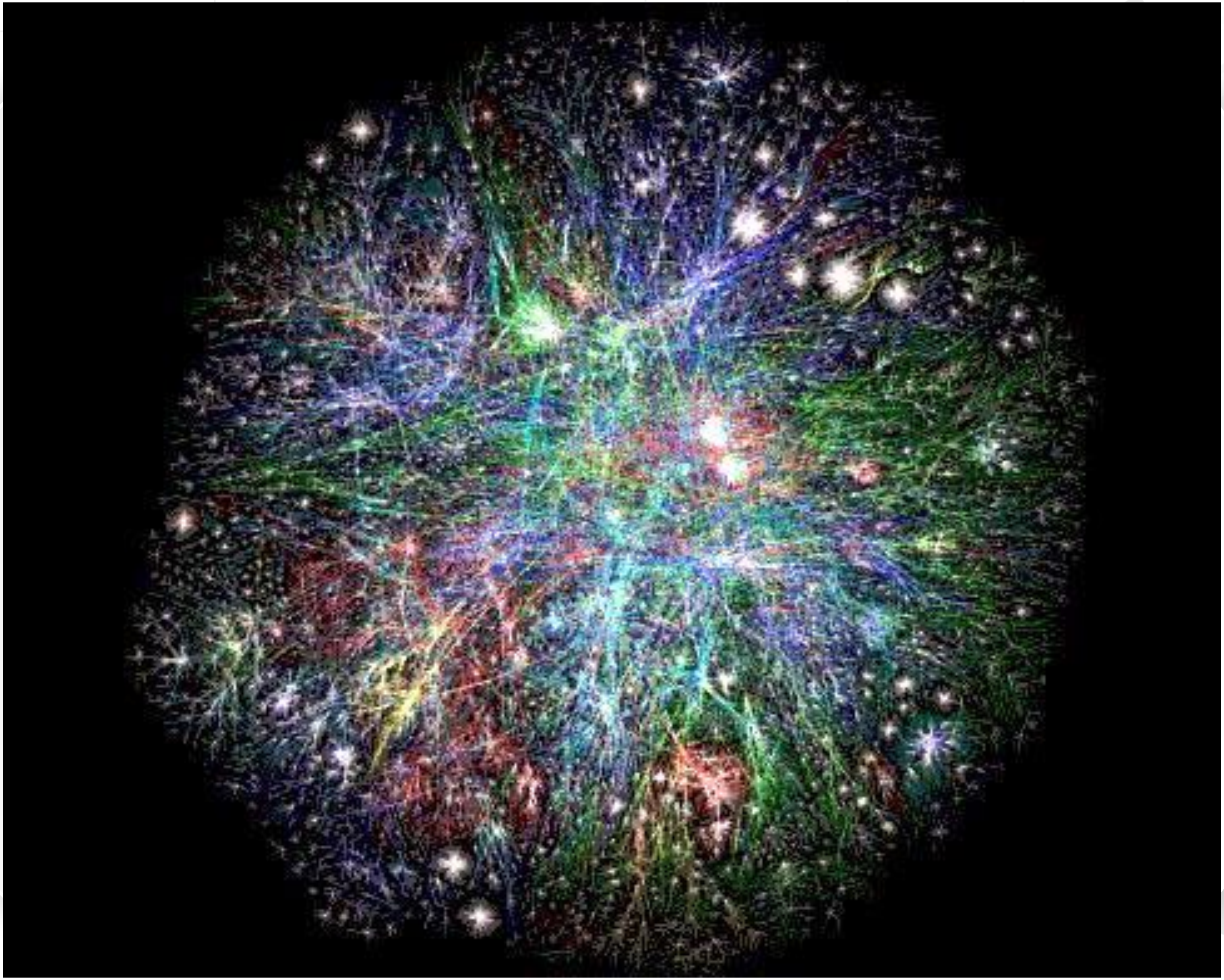
US Airline network

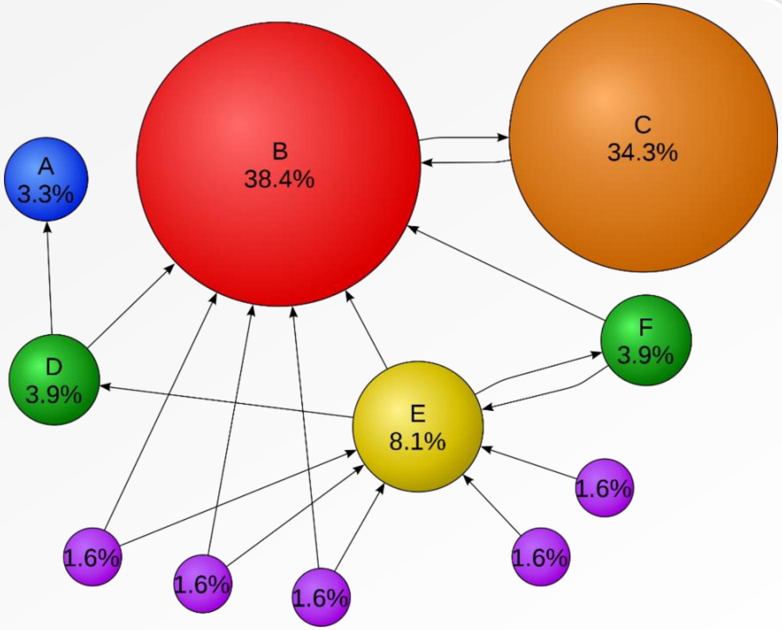
Telecommunication network



Networks are at the heart of some of the most revolutionary technologies of the 21st century, empowering everything from Google to Facebook, CISCO, and Twitter.

GRAPH OF THE WORLD WIDE WEB

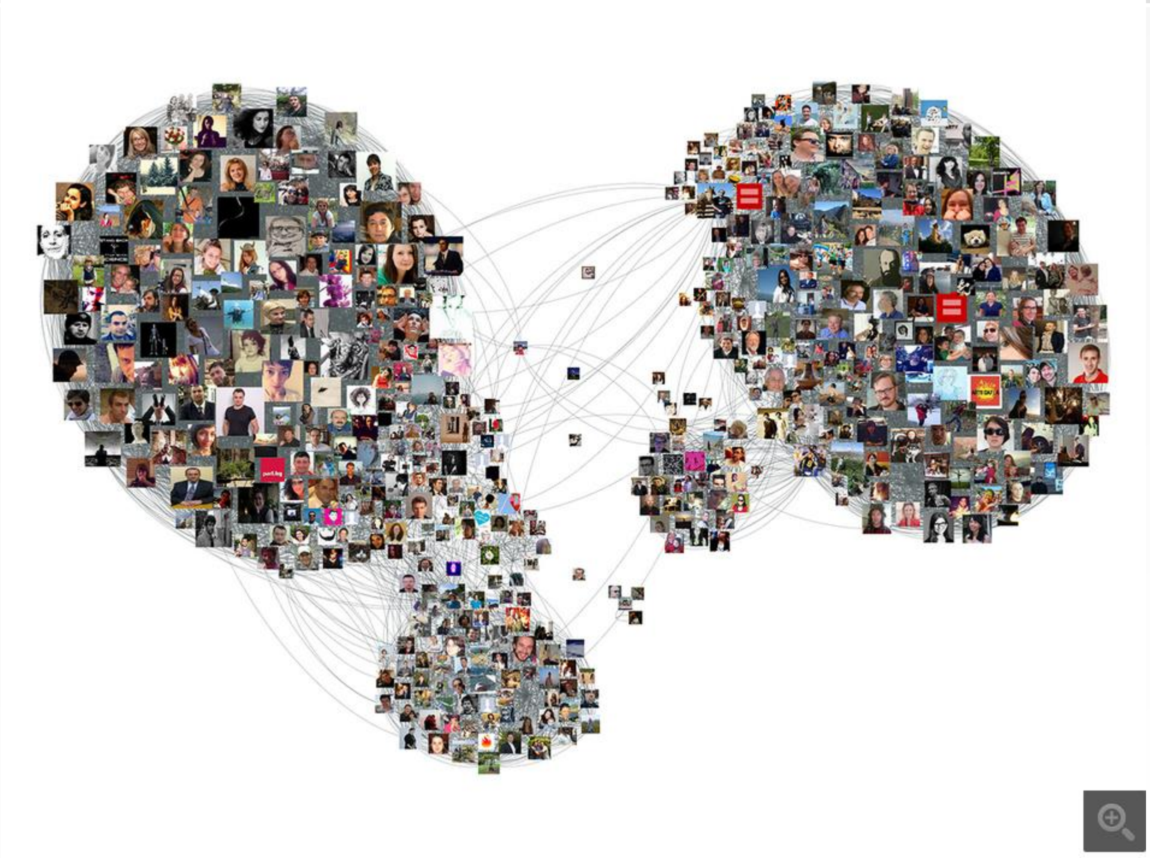




Mathematical PageRanks for a simple network, expressed as percentages. (Google uses a logarithmic scale.) Page C has a higher PageRank than Page E, even though there are fewer links to C; the one link to C comes from an important page and hence is of high value.

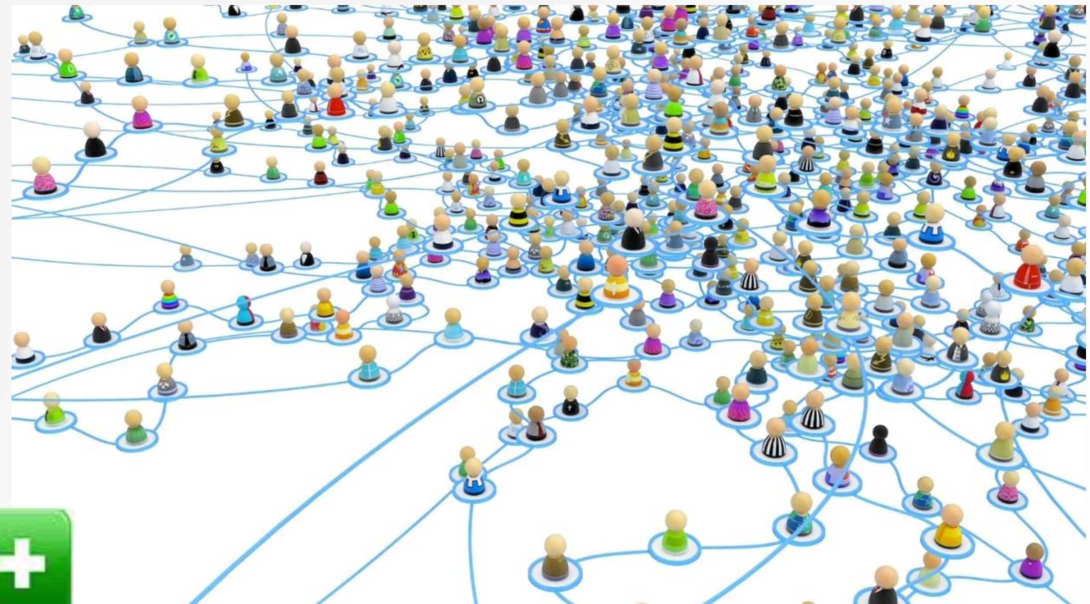
http://en.wikipedia.org/wiki/PageRank

THE “SOCIAL GRAPH” BEHIND FACEBOOK



Collecting and visualizing **Facebook networks** using people's profile photos as node images with **Gephi**

SOCIAL NETWORK



If we build a model of the friendship among people, we get a network (graph)!

SIX-DEGREES OF SEPARATION "STANLEY MILGRAM'S EXPERIMENT

Milgram’s small world experiment:

- Target individual in Boston
- Initial senders in Omaha, Nebraska
- Each sender was asked to forward a packet to a friend who was closer to the target
- Friends asked to do the same

Result: Average of ‘six degrees’ of separation.

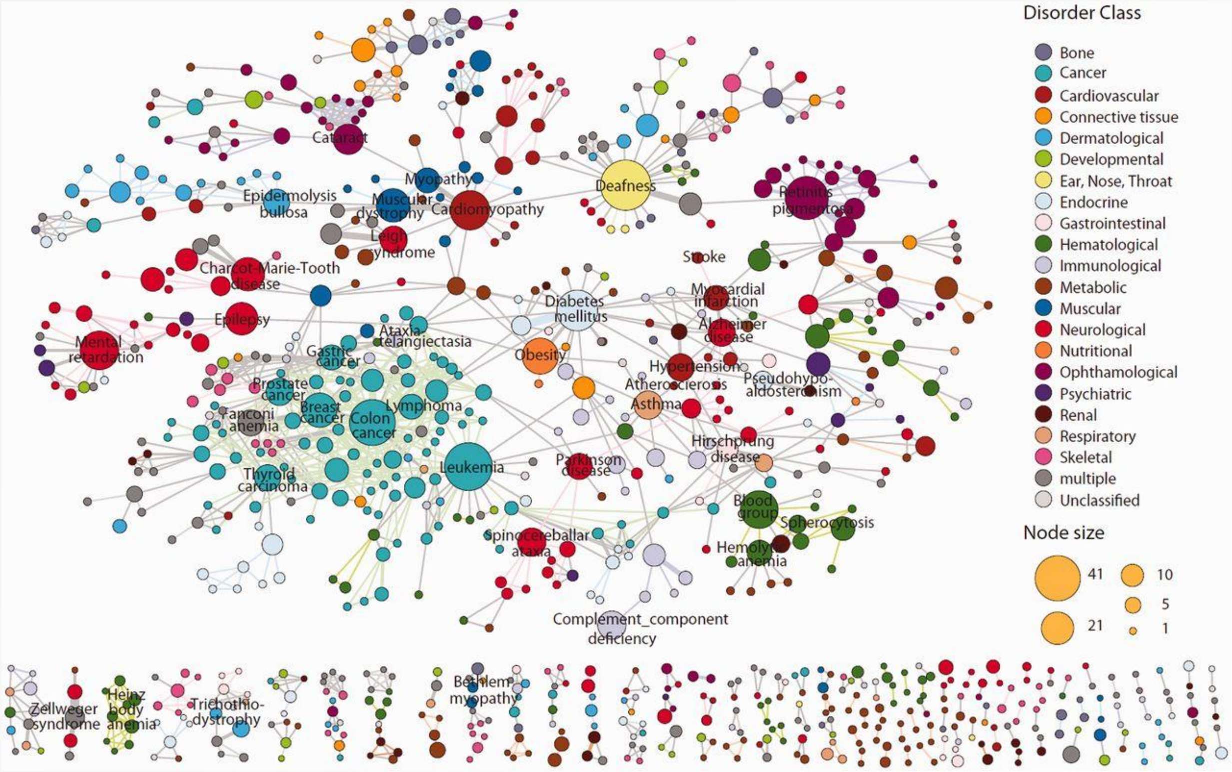
S. Milgram, *The small world problem,* Psych. Today, 2 (1967), pp. 60-67.

WHAT A SMALL WORLD

Six-degree:

- The average path length between any two nodes is short.
- E.g., the average shortest path length of the entire WWW is 19 clicks (hyperlinks).

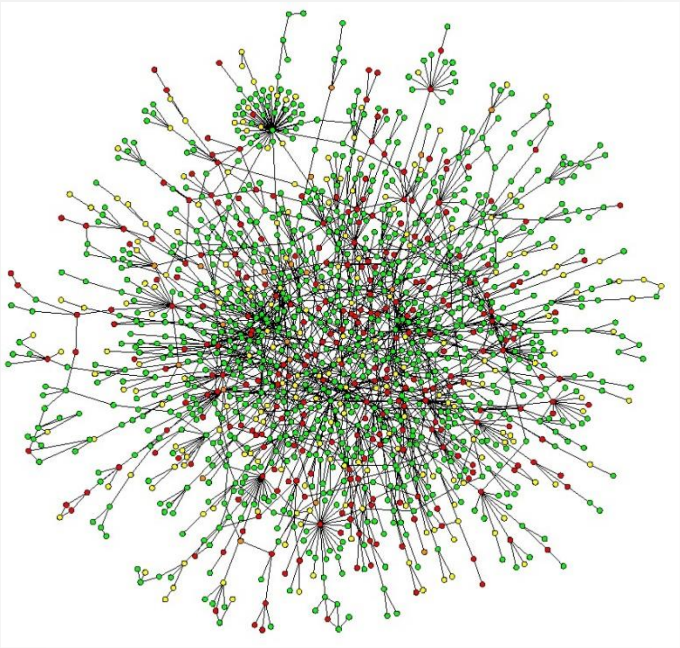
DISEASE NETWORK



Graph representation of the human disease network. Two disease nodes (circles) are connected if they share a common genetic component according to disorder disease-gene associations listed in OMIM as of the year 2005.

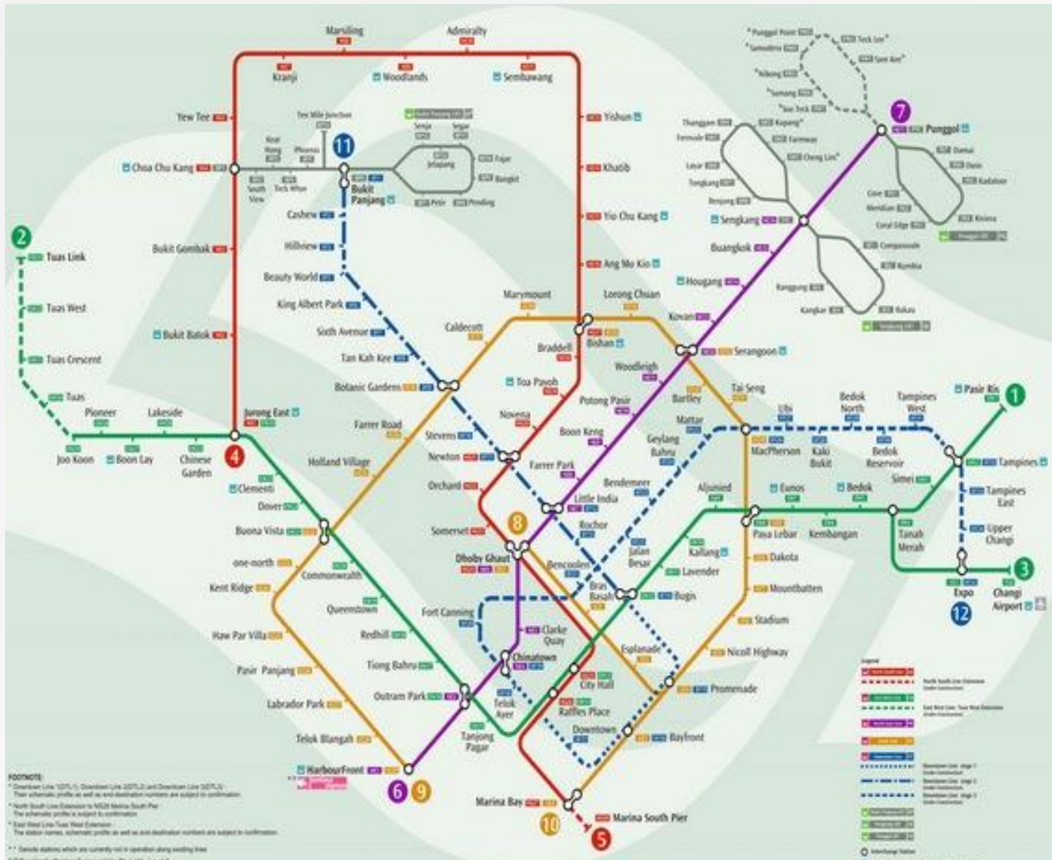
http://bfg.oxfordjournals.org/content/11/6/533.full.pdf

A PROTEIN NETWORK



The image above shows a subset of the PPI network for yeast. Dots represent proteins, and lines connecting the dots (edges) represent interactions between proteins. In general, the further apart two dots are, the less likely it is that their functions are related. Note also that this is a VERY small subset; there are ~6000 yeast proteins and about ~216 unique interactions in the database we used.

https://parasol.tamu.edu/dreu2013/Doroschak/



Singapore's MRT system

NETWORK DYNAMICS: HOT RESEARCH TOPIC

- Social, biological, and technological networks
- Features: power-law degree distribution, clustering coefficient, assortativity, community structure, and hierarchical structure, motif, etc.
- Dynamics: disease spread, synchronization, Google (pageRank), recommendation (big data), network control , etc.
- If you write computer simulations for network approaches, you need to know

  Graph Data Structure!

NOW YOU KNOW.....

- What is a graph?
- Implement a graph in C
- Traversal of a graph (BFS)
- Find the path from one node to another
- Understand how useful and important of graph

<Connected: The Power of Six Degrees>:

https://www.youtube.com/watch?v=2rzxAyY7D7k

Graph is not in the scope of the exam, but you should know graph data structure