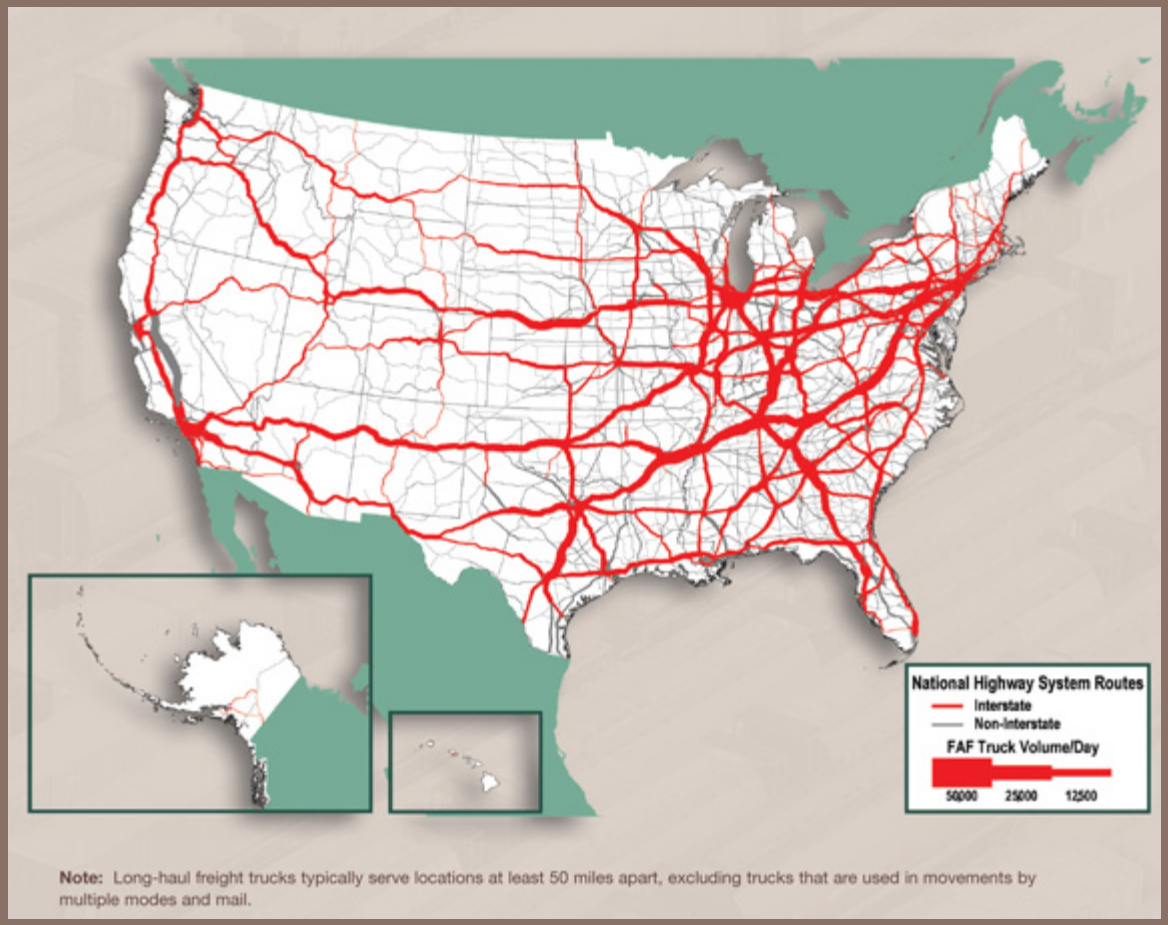
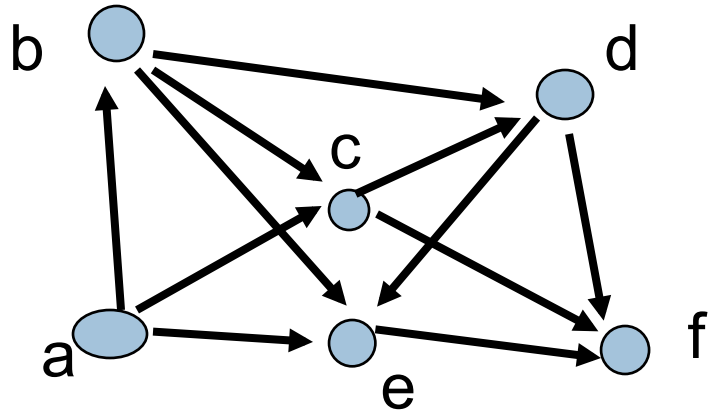


GRAPHS IV: APPLICATIONS

Lecture 22

CS2110 – Spring 2014

Example Directed Graph (Digraph)



V = {a,b,c,d,e,f}

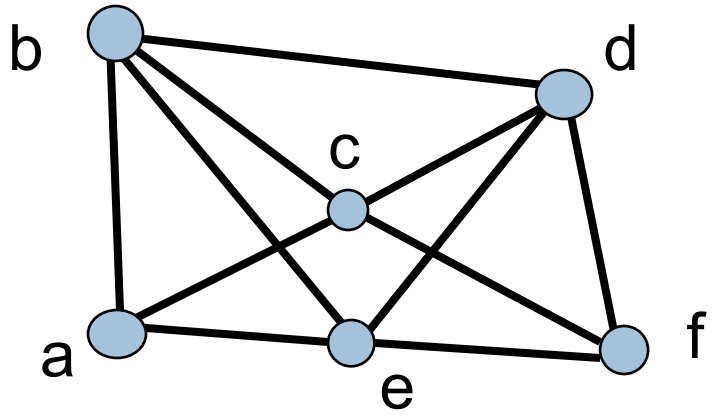
E = {(a,b), (a,c), (a,e), (b,c), (b,d), (b,e), (c,d),
(c,f), (d,e), (d,f), (e,f)}

|V| = 6, |E| = 11

Example *Undirected* Graph

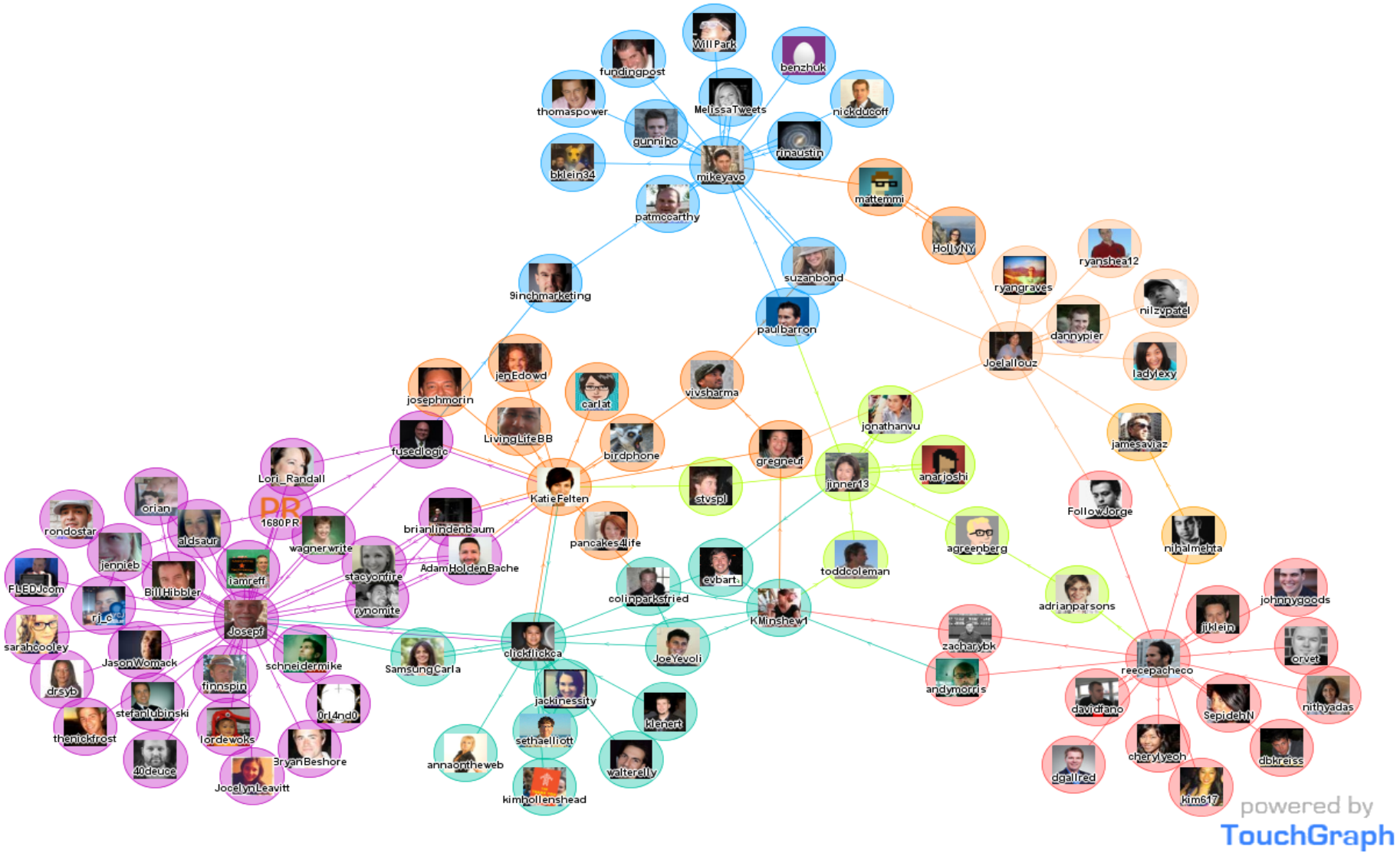
An *undirected graph* is just like a directed graph, except the edges are *unordered pairs* (*sets*) {u,v}

Example:



V = {a,b,c,d,e,f}
E = {{a,b}, {a,c}, {a,e}, {b,c}, {b,d}, {b,e}, {c,d}, {c,f},
{d,e}, {d,f}, {e,f}}

Social Network Graph



powered by TouchGraph

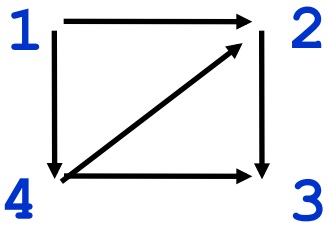
Graph Concepts and Algorithms

- Adjacency Matrix
- Topological Sort
- Search
  - depth-first search
  - breadth-first search
- Shortest paths
  - Dijkstra's algorithm
- Minimum spanning trees
  - Prim's algorithm
  - Kruskal's algorithm

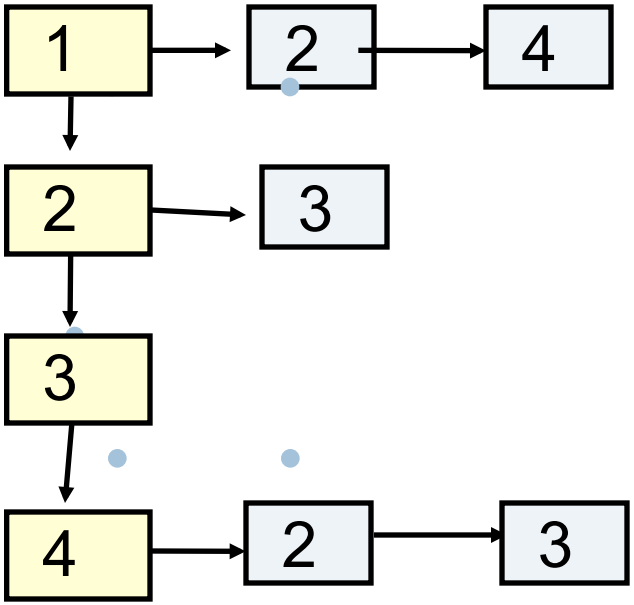
Others:

- Graph Coloring
- Planarity
- Traveling Salesman problem.

Graph Adjacency Matrix



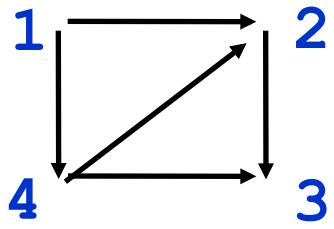
Adjacency List



Adjacency Matrix

	1	2	3	4
1	0	1	0	1
2	0	0	1	0
3	0	0	0	0
4	0	1	1	0

Graph Adjacency Matrix



Adjacency Matrix

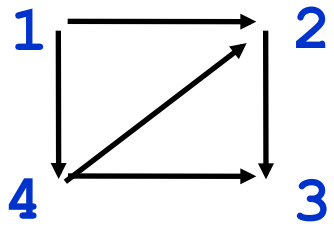
	1	2	3	4
1	0	1	0	1
2	0	0	1	0
3	0	0	0	0
4	0	1	1	0

^{2} =

	1	2	3	4
1	0	1	2	0
2	0	0	0	0
3	0	0	0	0
4	0	0	1	0

Squaring adjacency matrix gives number of possible paths of length 2!

Graph Adjacency Matrix



Adjacency Matrix

Cubing adjacency matrix gives number of possible paths of length 3!

	1	2	3	4
1	0	1	0	1
2	0	0	1	0
3	0	0	0	0
4	0	1	1	0

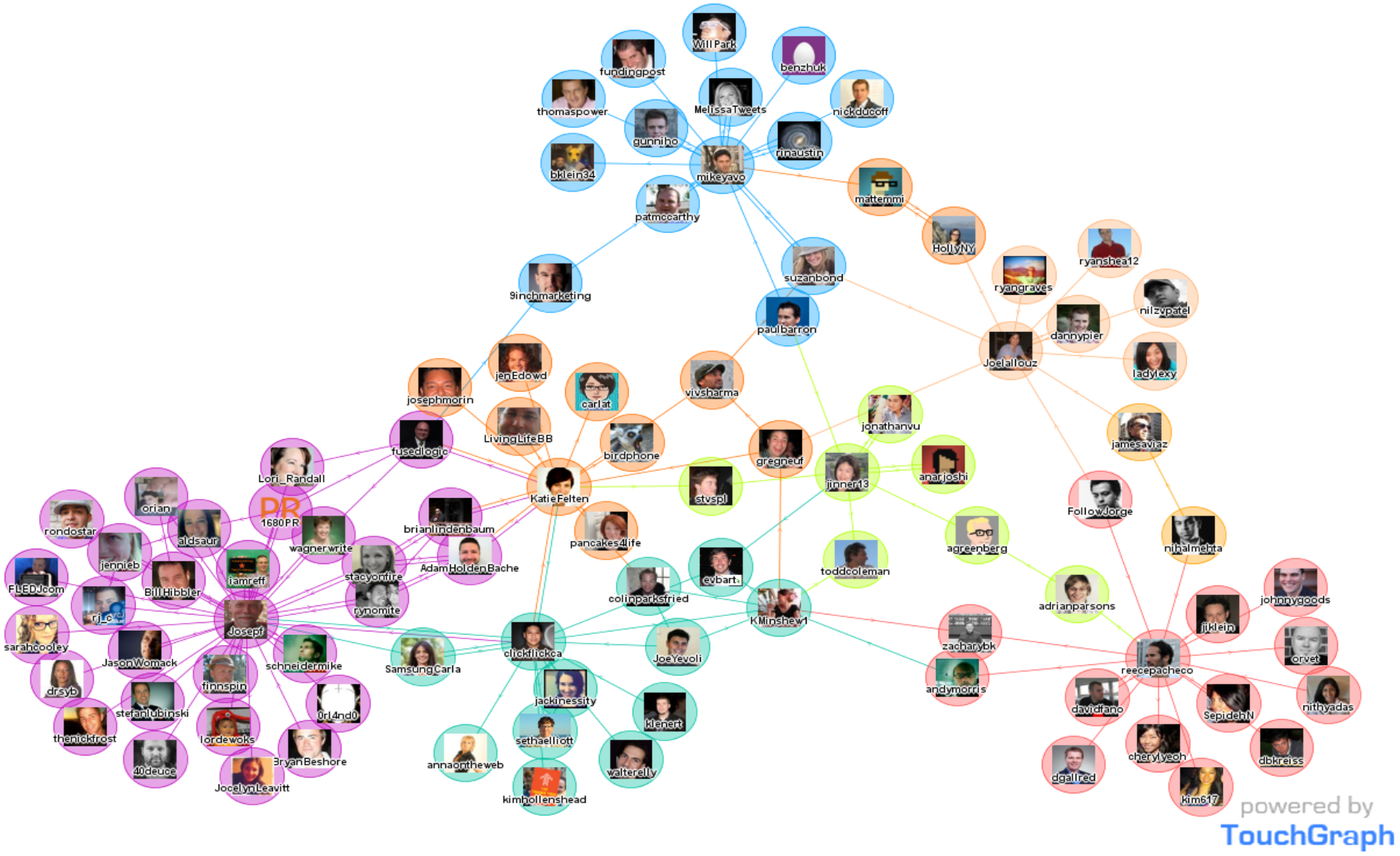
3 =

	1	2	3	4
1	0	0	1	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

Graph Adjacency Matrix

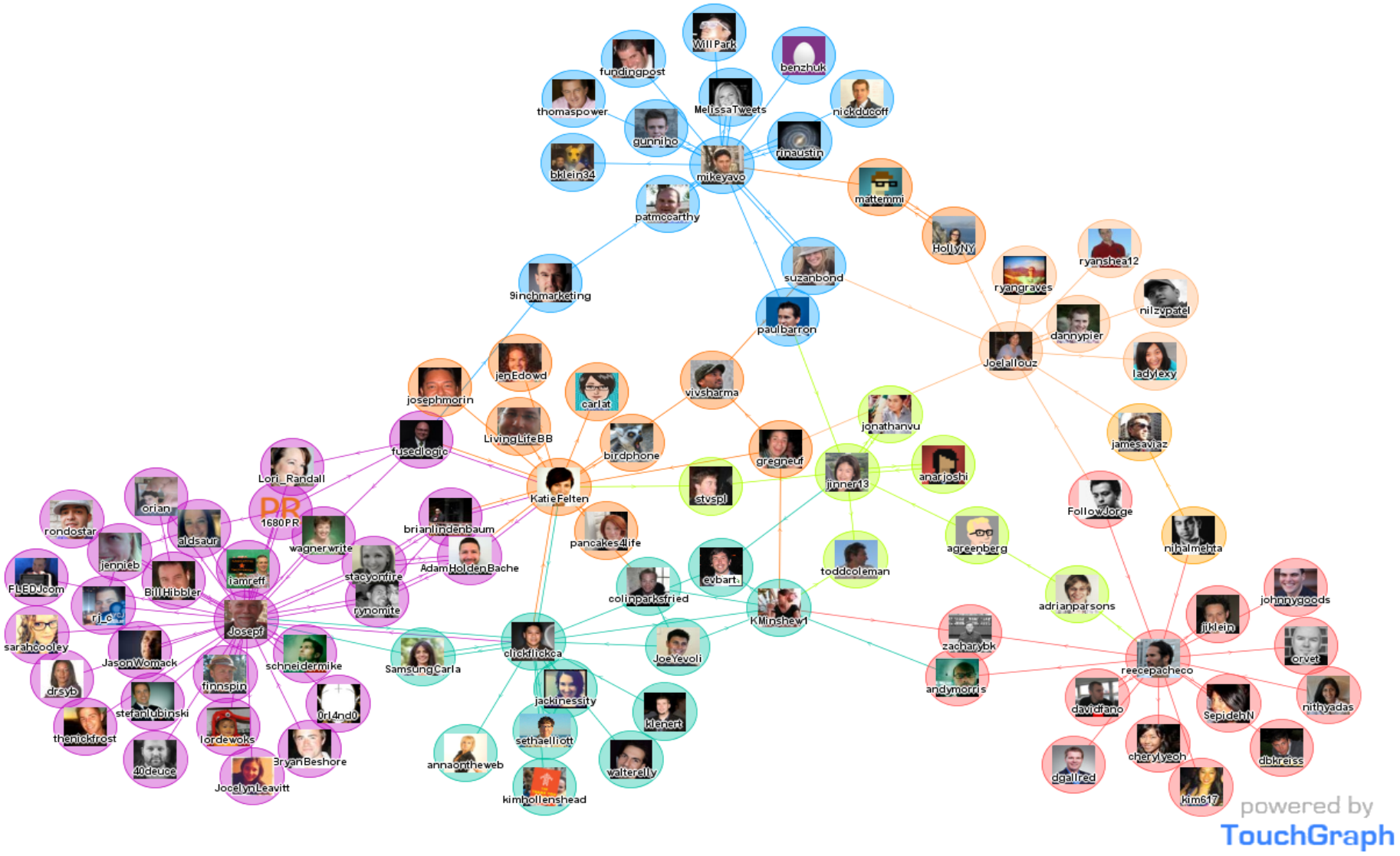
Same idea holds for undirected graphs.

How would you find friends of friends?



powered by TouchGraph

Find the number of common friends?



powered by TouchGraph

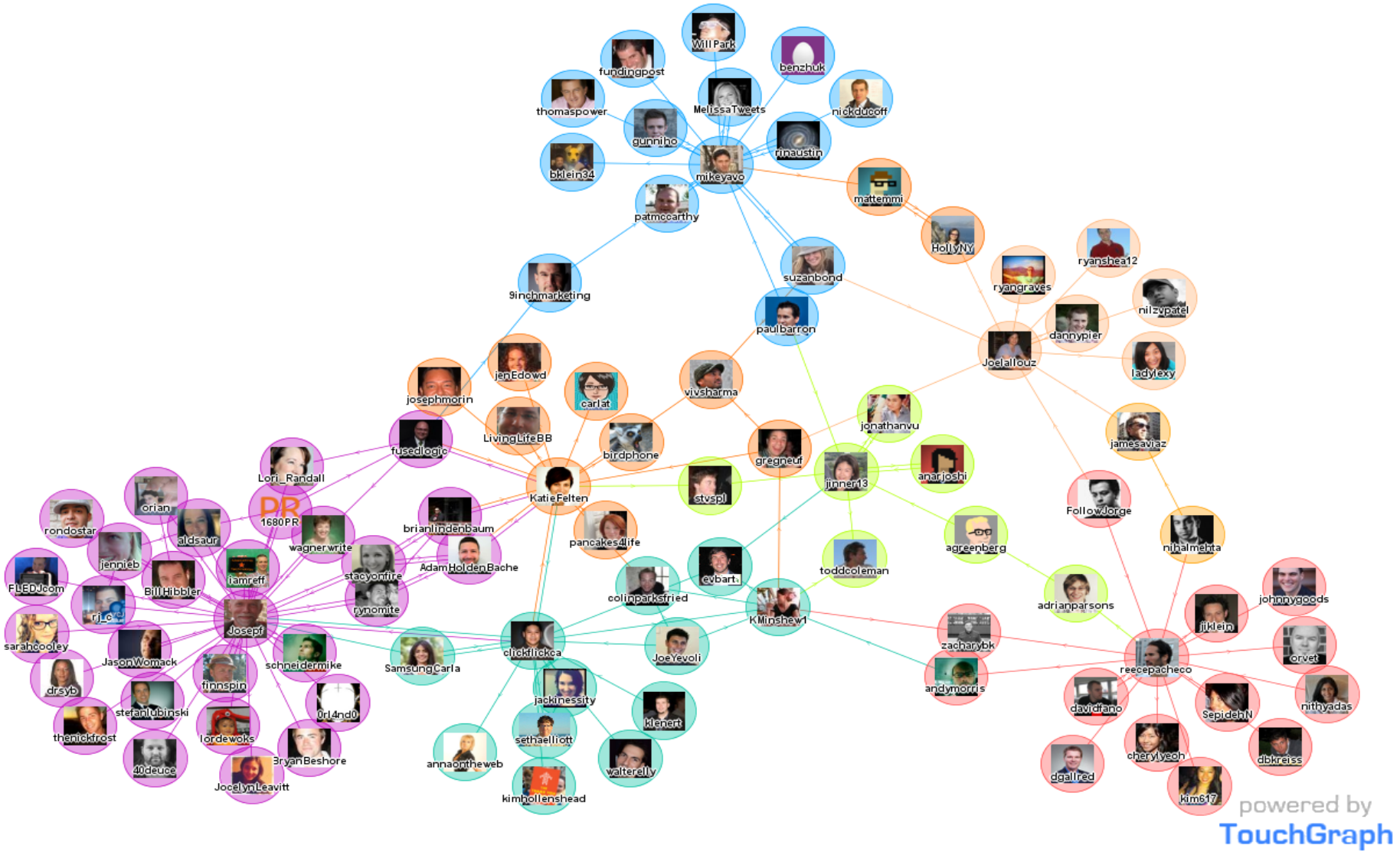
Graph Concepts and Algorithms

- ✓ Adjacency Matrix
- Topological Sort
- Search
  - depth-first search
  - breadth-first search
- Shortest paths
  - Dijkstra's algorithm
- Minimum spanning trees
  - Prim's algorithm
  - Kruskal's algorithm

Others:

- Graph Coloring
- Planarity
- Traveling Salesman problem.

Search if a person is someone's Friend?



powered by TouchGraph

Graph Concepts and Algorithms

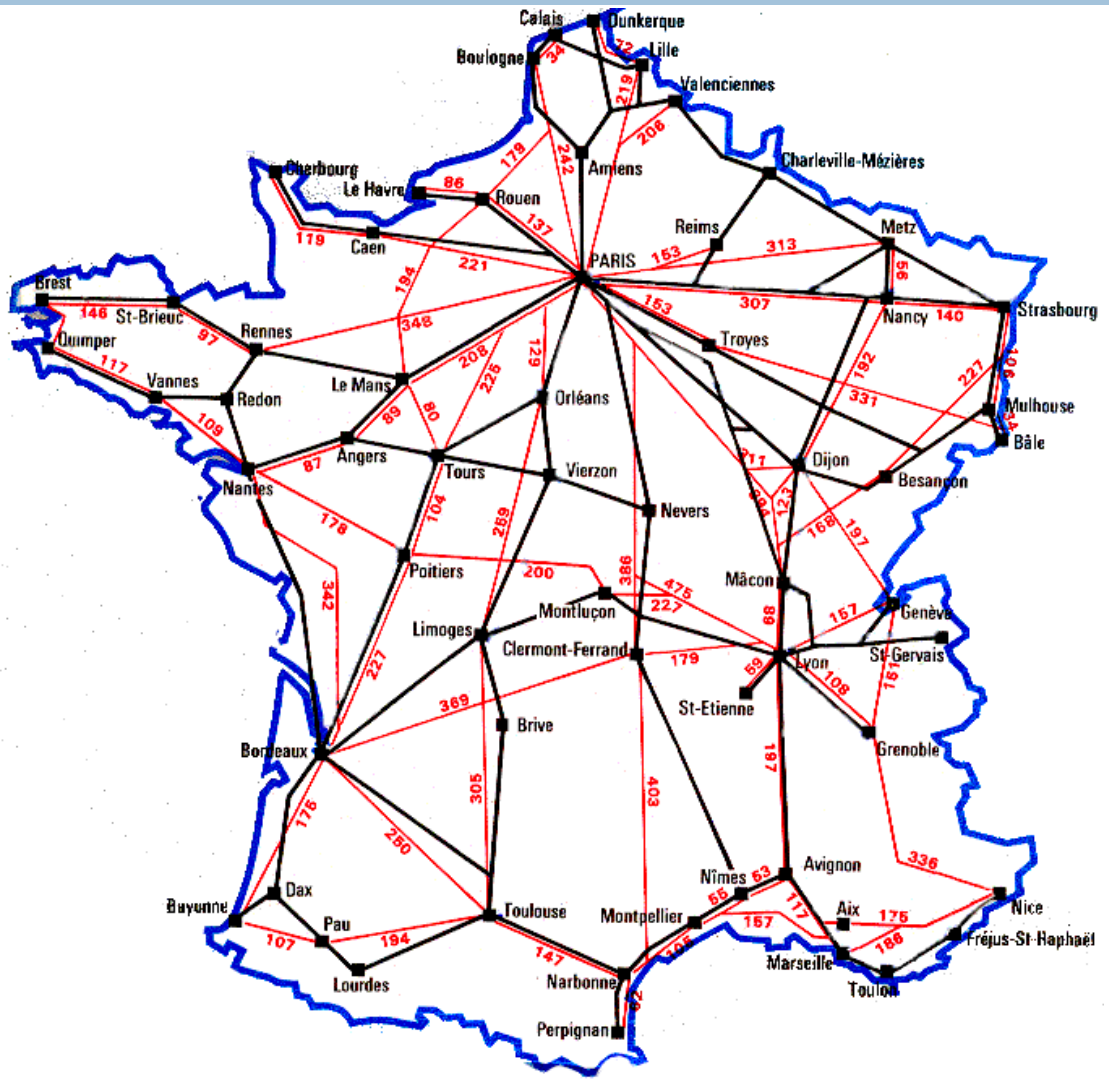
- ✓ Adjacency Matrix
- Topological Sort
- ✓ Search
  - depth-first search
  - breadth-first search
- Shortest paths
  - Dijkstra's algorithm
- Minimum spanning trees
  - Prim's algorithm
  - Kruskal's algorithm

Others:

- Graph Coloring
- Planarity
- Traveling Salesman problem.

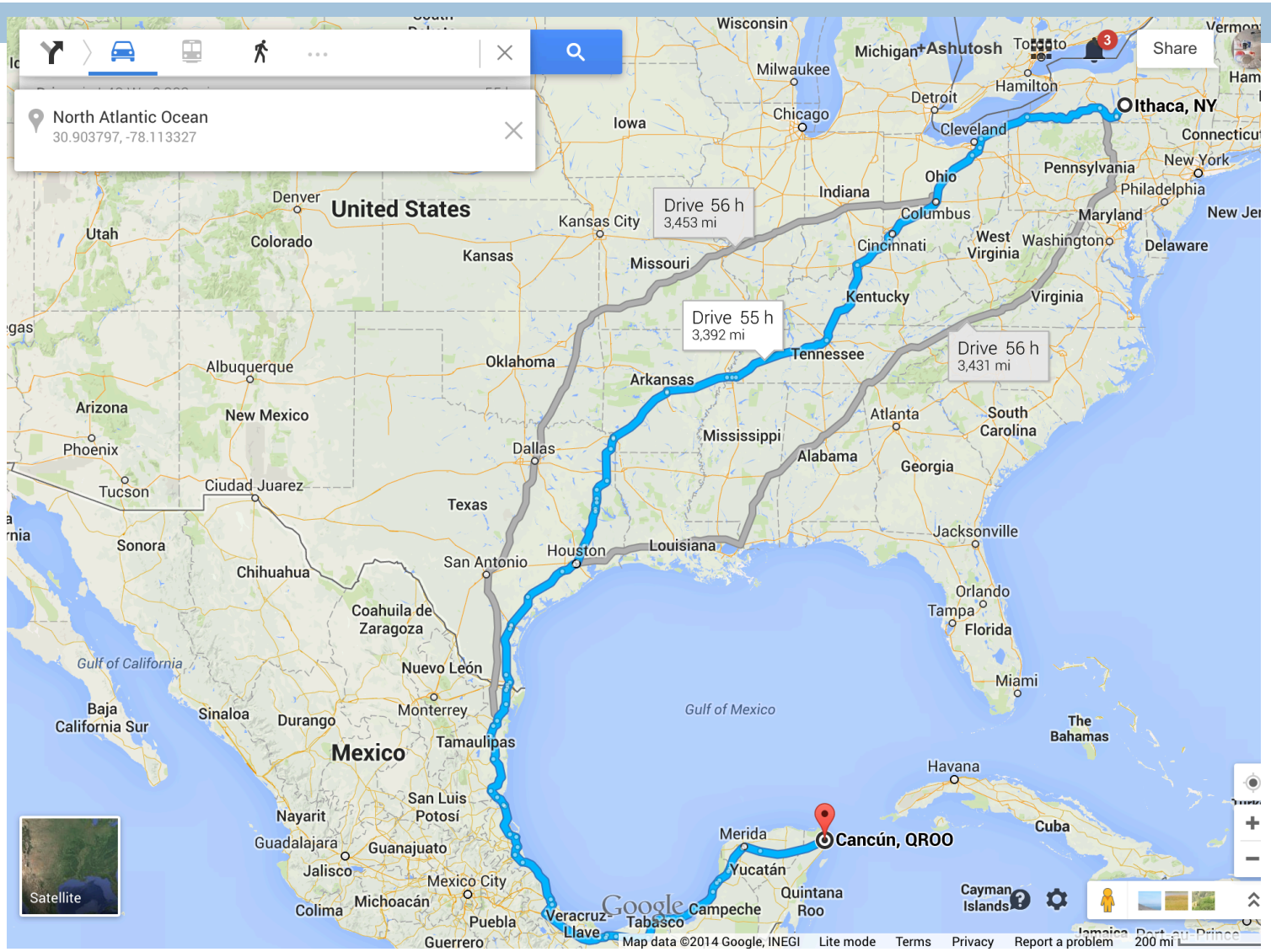
Shortest Path Algorithm: Used Everyday!

Train Network in France.



Shortest Path Algorithm: Used Everyday!

Maps!
GPS
directions!



Graph Concepts and Algorithms

- ✓ Adjacency Matrix
- Topological Sort
- ✓ Search
  - depth-first search
  - breadth-first search
- ✓ Shortest paths
  - Dijkstra's algorithm
- Minimum spanning trees
  - Prim's algorithm
  - Kruskal's algorithm

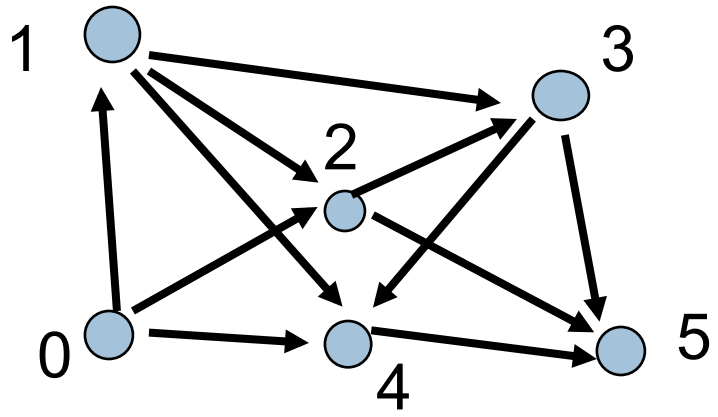
Others:

- Graph Coloring
- Planarity
- Traveling Salesman problem.

Topological Sort

- Topological sort of the dag

  This is a numbering of the vertices such that all edges go from lower- to higher-numbered vertices



- Useful in job scheduling with precedence constraints

Tell Me Dave: Making ‘Avoffato’

Input:

Environment:

NL Instruction:

Place the pot on the tap and turn the tap on. When it is filled, turn the tap off and heat the pot.

Place the pot on the tap

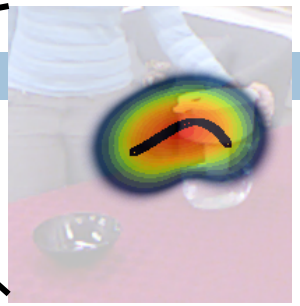
Turn the tap on

Turn the tap off

Heat the pot

Output: Inferred

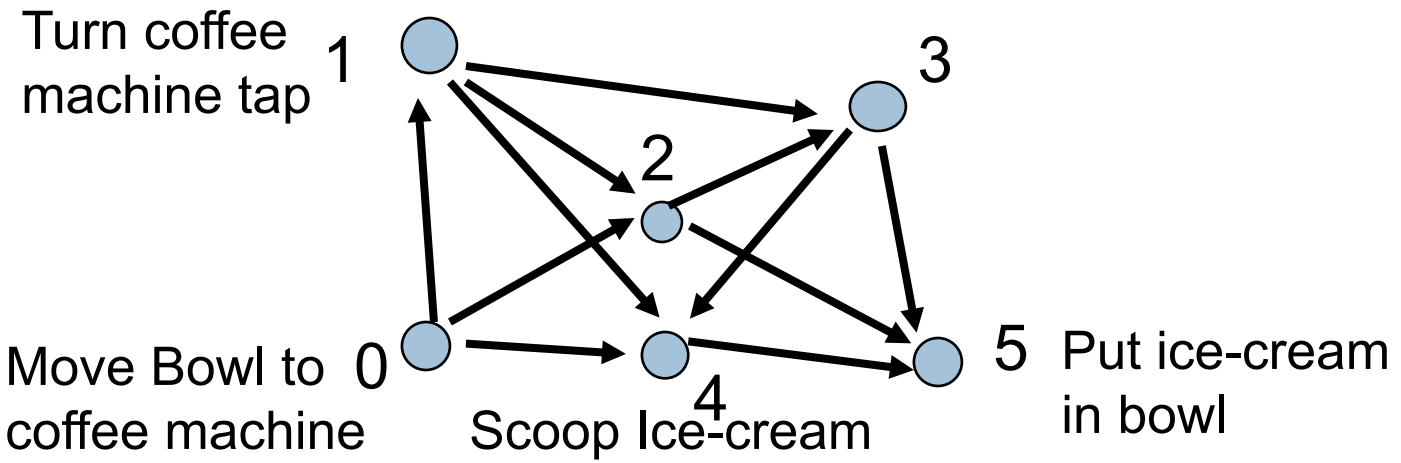
```
moveTo(pot1)
grasp(pot1)

moveTo(sink1)
keep(pot1,sink1,on)

<no-op>

~~keep(pot2,sink2,on)~~
toggle(sink1knob1,on)
wait()

<no-op>

toggle(sink1knob1,off)

moveTo(pot1)
grasp(pot1)
moveTo(stove1)

keep(pot1,burner1,on)
toggle(stove1knob1,on)
wait()
toggle(stove1knob1,off)
```



Topological Sort

- Topological sort of the dag

  This is a numbering of the vertices such that all edges go from lower- to higher-numbered vertices



- Useful in job scheduling with precedence constraints

Graph Concepts and Algorithms

- ✓ Adjacency Matrix
- ✓ Topological Sort
- ✓ Search
  - depth-first search
  - breadth-first search
- ✓ Shortest paths
  - Dijkstra's algorithm
- Minimum spanning trees
  - Prim's algorithm
  - Kruskal's algorithm

Others:

- Graph Coloring
- Planarity
- Traveling Salesman problem.

AI: Vision and Robotics

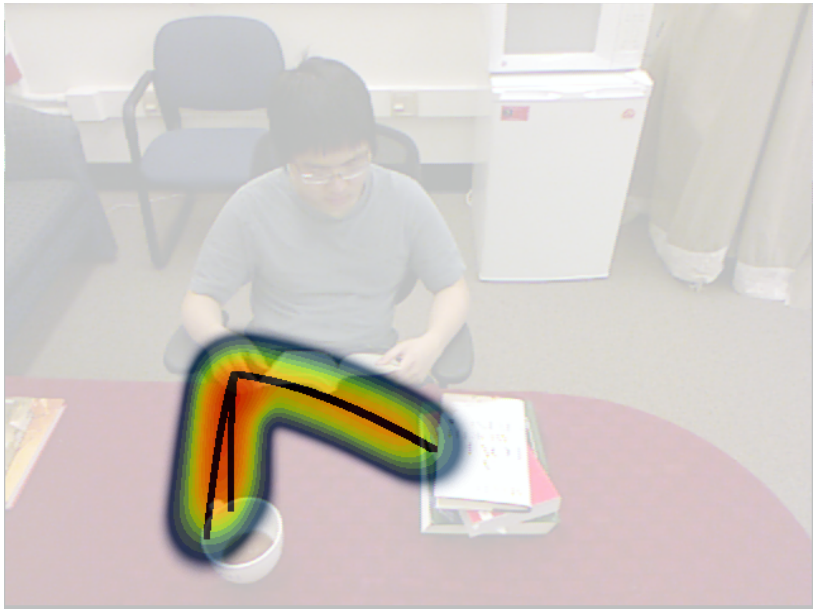
- Number of cell-phones in 2014: 7 billion!
- So many images everywhere.
- But do computers understand the images?

Without Graph Algorithms

Anticipating Future Actions

Anticipating Future Actions

Robot's view

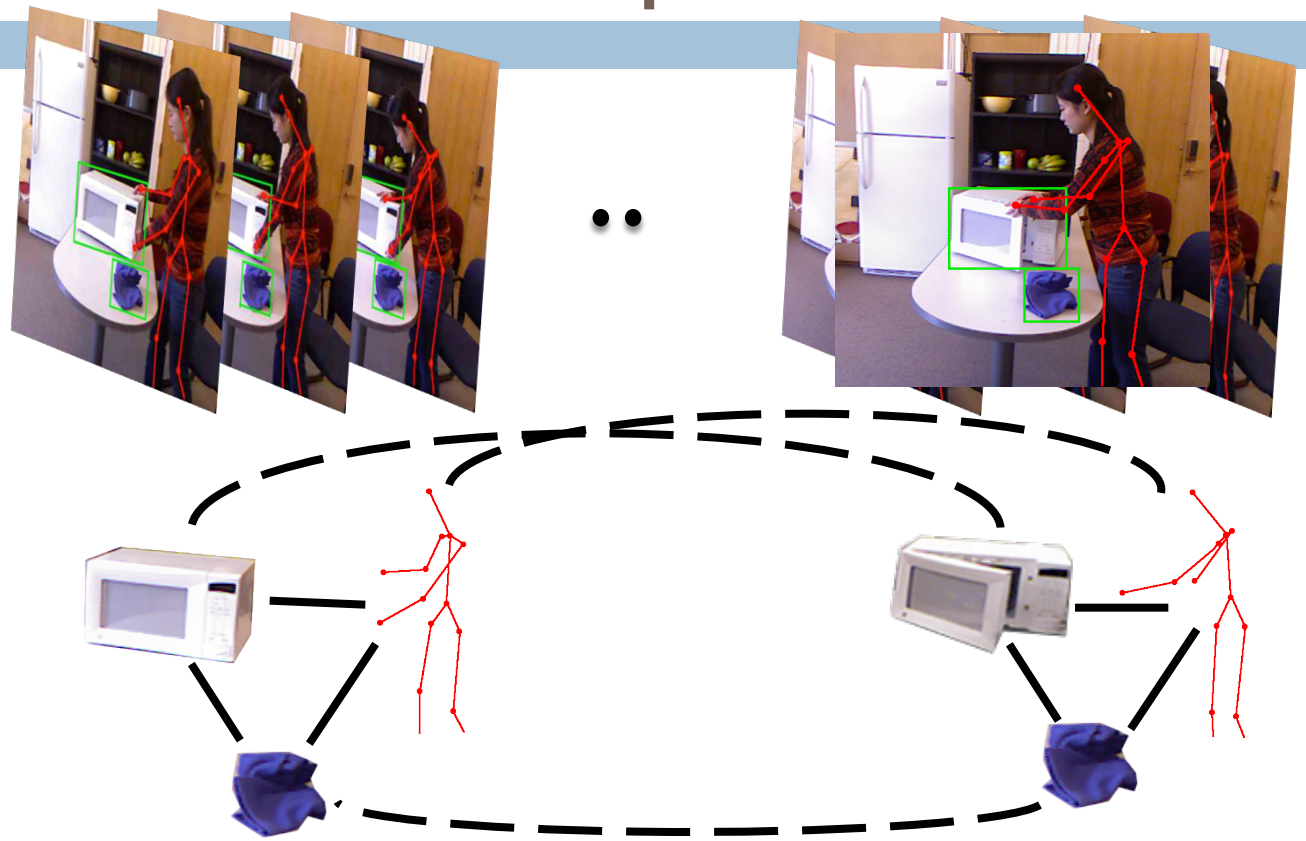


Future anticipations

- Humans use anticipation all the time
  - e.g., interacting with other people, playing sports, driving, etc.

Modeling the Activities with an Undirected Graph

Modeling the Activities with an Undirected Graph



$P(\mathcal{O}, \mathcal{A}|\mathcal{H}, \mathcal{L})$

Undirected Graphs: Probabilities on the Graphs

Robot Sees the Future!

Studio B, with Shepherd Smith

With Lewis Black, Comedy Central

Graph Concepts and Algorithms

- ✓ Adjacency Matrix
- ✓ Topological Sort
- ✓ Search
  - depth-first search
  - breadth-first search
- ✓ Shortest paths
  - Dijkstra's algorithm
- ✓ Minimum spanning trees
  - Prim's algorithm
  - Kruskal's algorithm

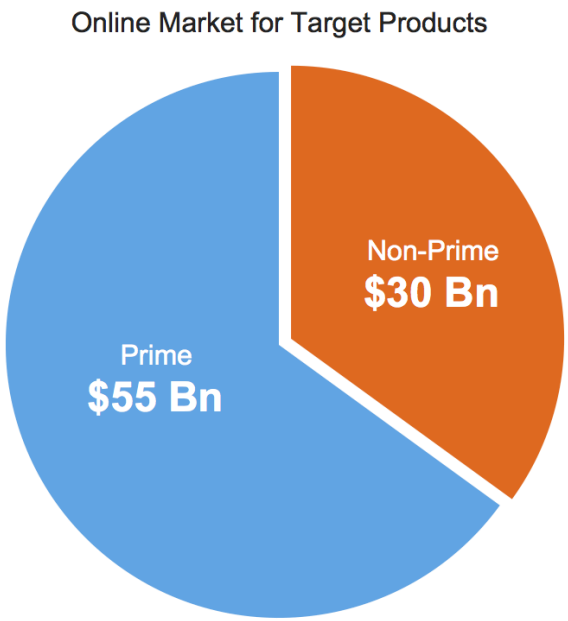
Others:

- Graph Coloring
- Planarity
- Traveling Salesman problem.

Graphs in Practice

- In 2012, we started a company called Cognical.
- Problem:

65 million US consumers cannot finance electronics, furniture, appliances and other durable goods online



1 in 3 US consumers cannot finance online purchases

COGNICAL

65 million US consumers

Have low FICO scores

but are broadly creditworthy

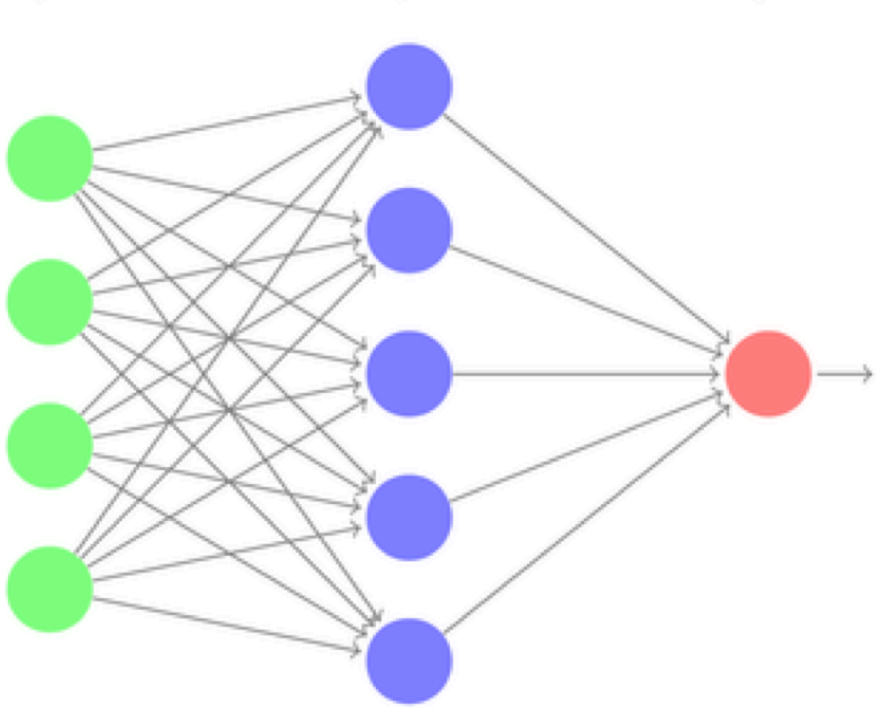
Buy at local stores

with cash or finance via Lease to Own (LTO)

Browse Online

but lack a payment option to buy

- Cognical collects data about customers from various sources.
- Represents over a graph, and predicts how to finance.



COGNICAL

1 **Q1 2014: Launch**


Electronics


Furniture


Appliances


Instruments

2 **2015: Market Penetration, Category Expansion**

$30 Bn

Target market

3 **2016: Brick & Mortar Expansion**

$96 Bn

Graphs!

- Adjacency Matrix
- Topological Sort
- Search
  - depth-first search
  - breadth-first search
- Shortest paths
  - Dijkstra's algorithm
- Minimum spanning trees
  - Prim's algorithm
  - Kruskal's algorithm

Others:

- Graph Coloring
- Planarity
- Traveling Salesman problem.