A Gentle Introduction to Graph Theory

(with a networking sensitivity)

Arnaud Casteigts

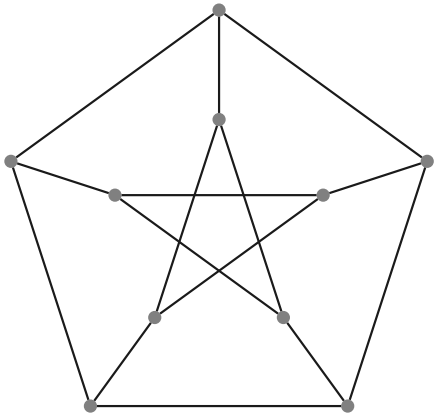
LaBRI, Université de Bordeaux

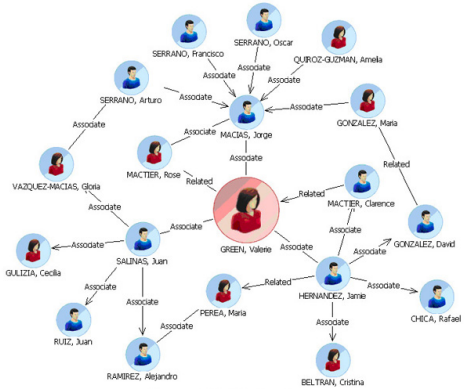
ResCom Summer School 2019
June 25, 2019

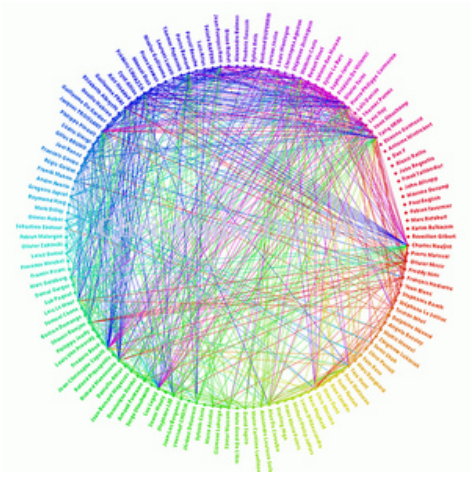
(Part I)

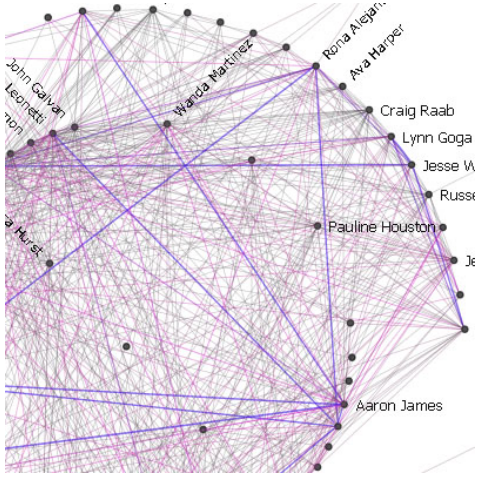
Introduction

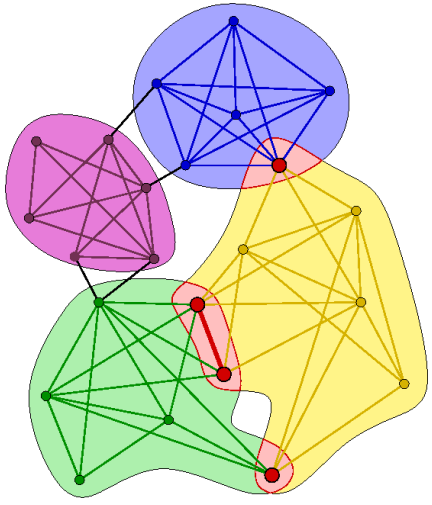
Some graphs

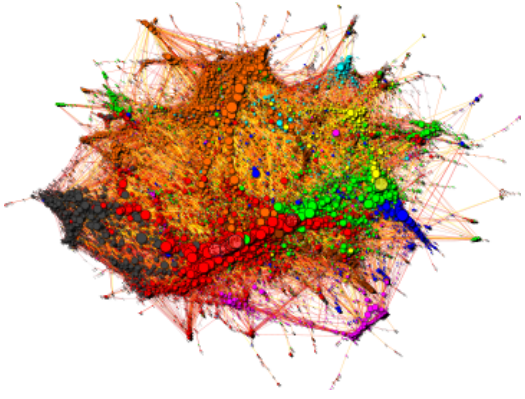



SERRANO, Francisco
SERRANO, Oscar
QUIROZ-GUZMAN, Amelia
SERRANO, Arturo
GONZALEZ, Maria
MACIAS, Jorge
MACTIER, Rose
VAZQUEZ-MACIAS, Gloria
MACTIER, Clarence
GONZALEZ, David
GREEN, Valerie
SALINAS, Juan
GULIZIA, Cecilia
HERNANDEZ, Jamie
CHICA, Rafael
PEREA, Maria
RUIZ, Juan
RAMIREZ, Alejandro
BELTRAN, Cristina
Associate
Related




Wanda Martinez
Craig Raab
Lynn Goga
Pauline Houston
Aaron James





A bit of history...

In 1735, in Königsberg (now Kaliningrad, Russia)

"Can we devise a walk that crosses every bridge exactly once?"

Resolved by Leonhard Euler, beginning of graph theory.

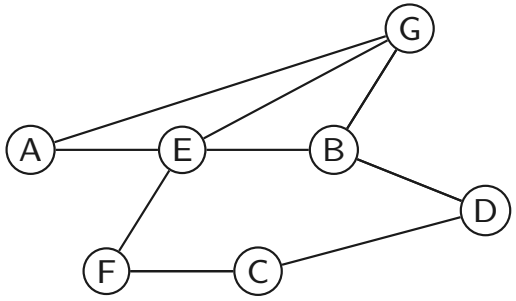
Some concepts:

- Graph $G = (V, E)$
- Set of vertices V and edges $E \subseteq V \times V$ (binary relations among vertices).
- Degree of a vertex: number of edges incident to it (e.g. above, max degree $= 5$)
- Path: sequence of edges $e_1, e_2, \ldots, e_k$ with $e_i = \{u_i, v_i\}$ such that $v_i = u_{i+1}$.
- Cycle: path that terminates where it starts
- Eulerian Cycle: cycle that uses every edge exactly once (the above problem)

Theorem: $G = (V, E)$ admits an Eulerian cycle if and only if the degree of every vertex V is even

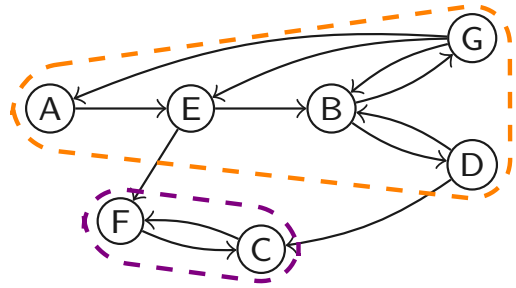
Types of graphs

Undirected graphs



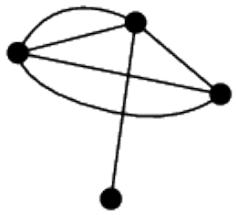
Symmetrical relations
(e.g. "have met", "are within distance x", ...)

Directed graphs

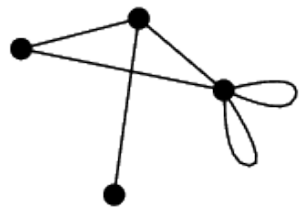


Asymmetrical relations
(e.g. "is older than", "is following", ...)

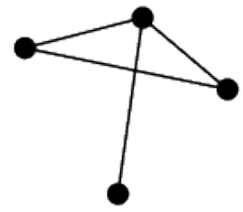
Multiple edges, loops, simple graphs...



graph with multiple edges



graph with loops



simple graph

(courtesy Wolfram)

Mathematically $G = (V, E)$ with...

- Directed: E is a set of *couples*
- Undirected: E is a set of *pairs*
- Multiedges: E is a multiset
- Loops : E allows reflexion (e.g. $\{u, u\}$)

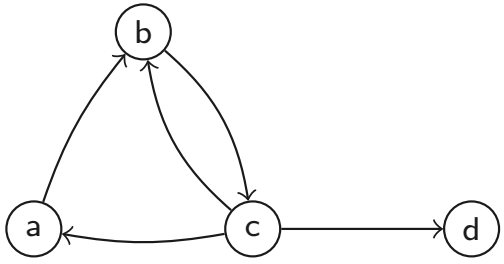
Basic concepts (directed/undirected)

- Paths, Connectivity
- Distance, Diameter, etc.
- Connected components

[+ weights]

Basic data structures for graphs

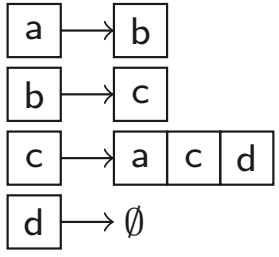
Directed graphs



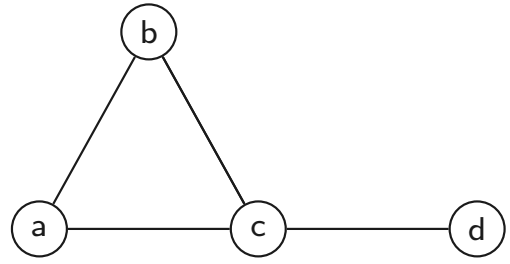
Adjacency matrix:

0	1	0	0
0	0	1	0
1	1	0	1
0	0	0	0

Adjacency lists:



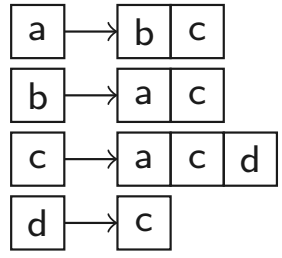
Undirected graphs



Adjacency matrix:

0	1	1	0
1	0	1	0
1	1	0	1
0	0	1	0

Adjacency lists:

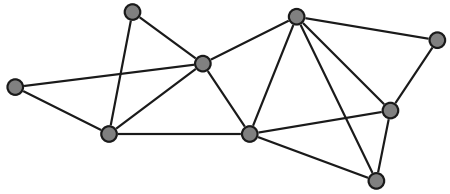


Other options (object-oriented structures, list of edges, *etc.*)

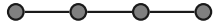
Some facts

- Matrices use $\Theta(n^2)$ memory even if graph is sparse
- Adjacency lists use only $\Theta(m)$ memory (where m is the number of edges)
- Matrices have algebraic features
  (e.g. M^i counts paths of length i; also G connected iff $\sum_i M^i$ has no zero)

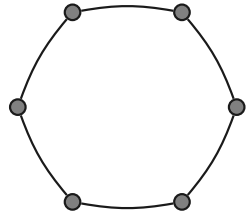
Some classes of graphs



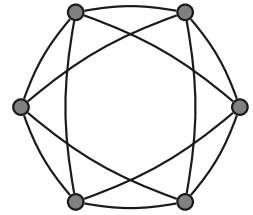
Arbitrary



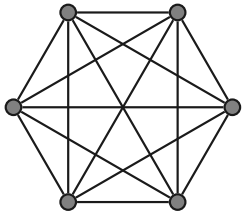
Path/Line



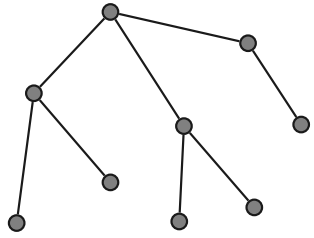
Cycle/Ring



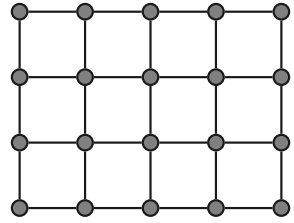
Cycle with chords



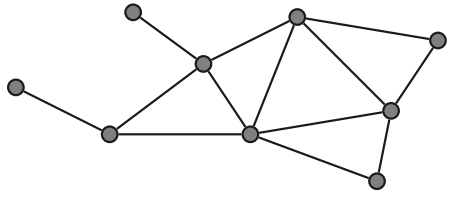
Complete



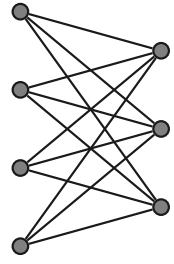
Tree



Grid



Planar



Bipartite

Simple exercises on simple graphs...

Degree sequences (ex 1.1)

Find a graph that realizes the given degree sequence. For example, $(2, 2, 2)$ could be realized by a triangle graph.

- $(3, 3, 2, 1, 1)$
- $(5, 3, 2, 1, 1, 1)$ odd sum of degrees is impossible! (Handshaking lemma)
- $(3, 3, 2, 2)$
- $(3, 3, 1, 1)$ sum of degrees is even, but still impossible, why?

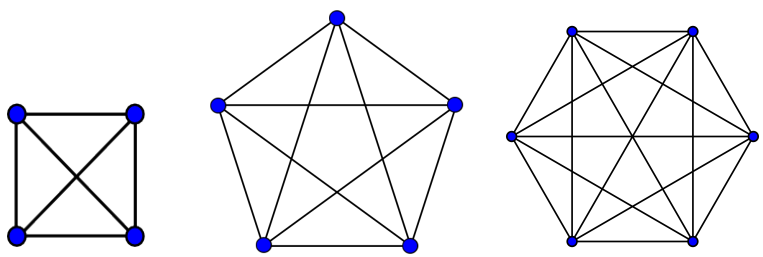
$\rightarrow$ Havel–Hakimi algorithm

Different degrees (ex 1.2)

Find a graph in which all the degrees are different

Planar graphs (ex 1.3)

Which of these graphs are planar? (= can be drawn without crossing).

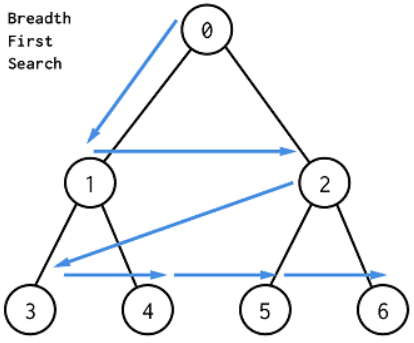


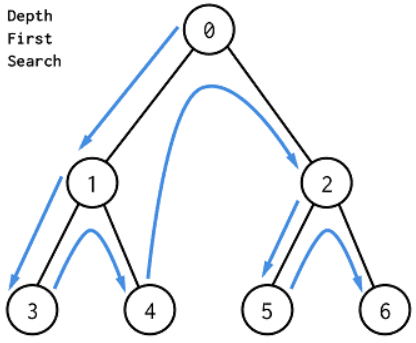
(Part II)

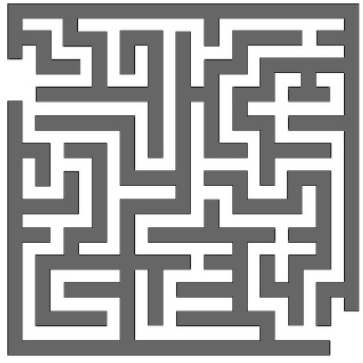
Overview of some topics
(with a Rescom sensitivity)

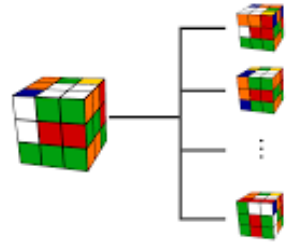
Graph search and shortest paths

Breadth-first search and Depth-first search



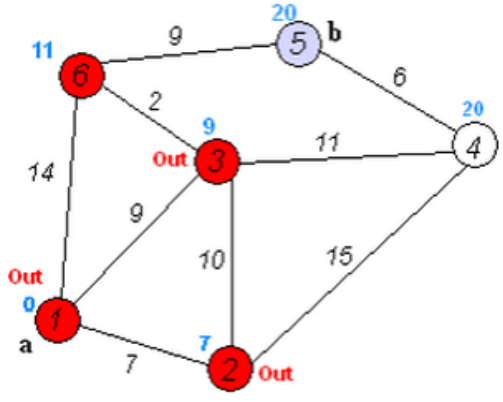






Shortest path, distance, ...

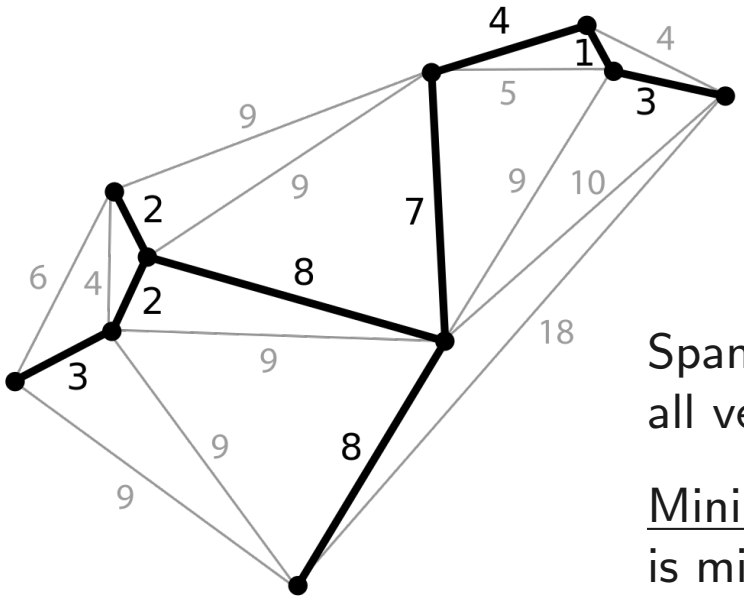
- Unweighted: BFS
- Weighted: Dijkstra
  $\rightarrow$ shortest path from a given vertex to all others



Some applications

- Command `find` on linux (DFS)
- Routing in networks (distributed versions)
- ...

Minimum Spanning Trees



Spanning tree: Set of edges that connects all vertices without cycle

Minimum spanning tree: Sum of weights is minimized over all possible spanning trees

Kruskal's algorithm

- Sort the edges by increasing weights
- For each edge e in sorted order:

  Add e to MST unless it creates a cycle

Prim's algorithm

- Start a tree at an arbitrary vertex
- While the tree is not spanning:

  Add the smallest edge between a node in the tree and a node outside the tree

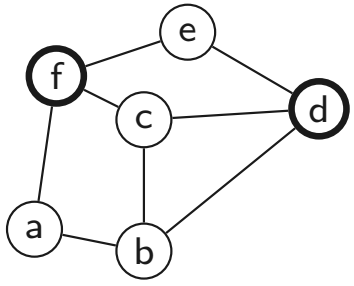
$\rightarrow$ This works (magically) because the solutions have a *matroid* structure.

Some Classical Covering Problems

Vertex cover (min), Dominating set (min), Independent set (max), matching (max)

(Minimum) Dominating set

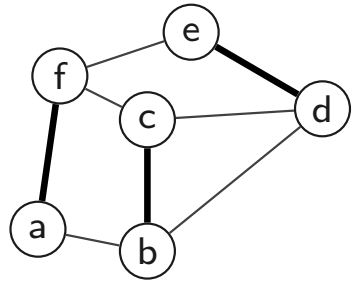
Set of vertices that "sees" every vertices



→ Relay nodes in wireless network
→ Location of fire stations in city

(Maximum) Matching

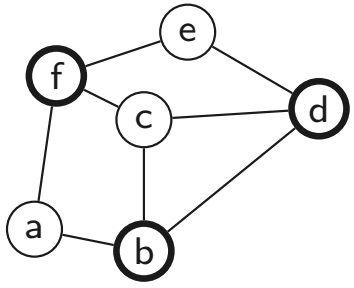
Set of non-adjacent edges



→ Pairwise communications
→ If bipartite: tasks assignment

(Minimum) Vertex cover

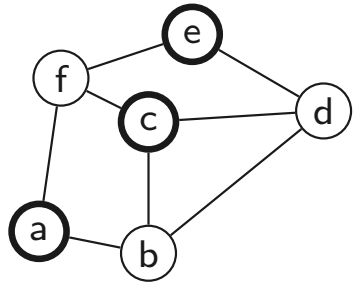
Set of vertices that "sees" every edge.



→ Interception of traffic in network

(Maximum) Independent Set

Set of non-neighbor vertices

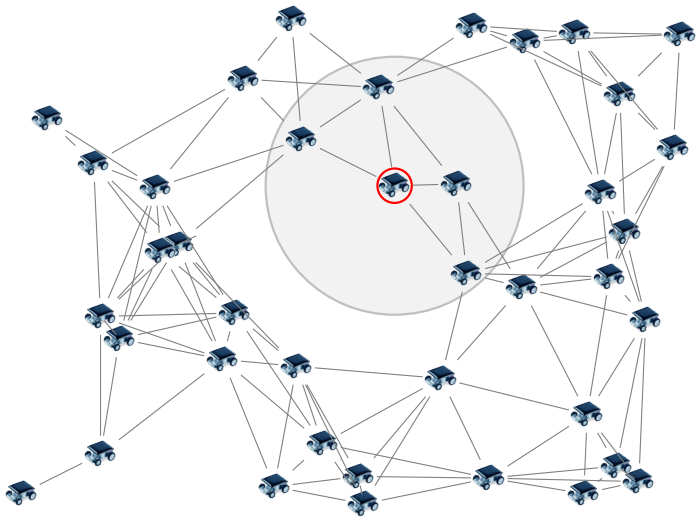


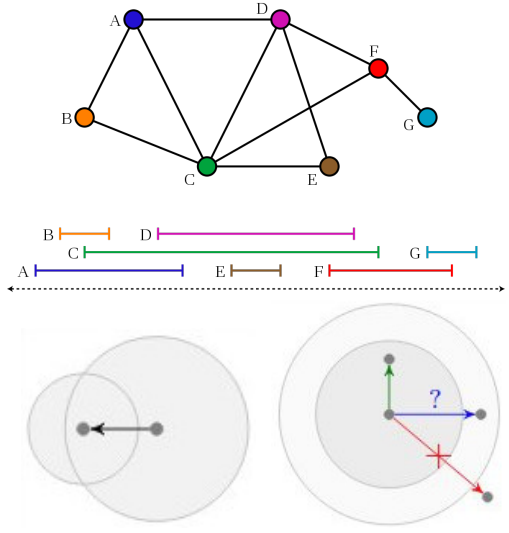
→ Frequency assignment in wireless networks

Unit disk graphs

Definition: Vertices have coordinates, there is an edge $\{u, v\}$ iff $dist(u, v) \leq x$

Particular case of *intersection graph*, analogue in 2D of *unit interval graphs* in 1D.



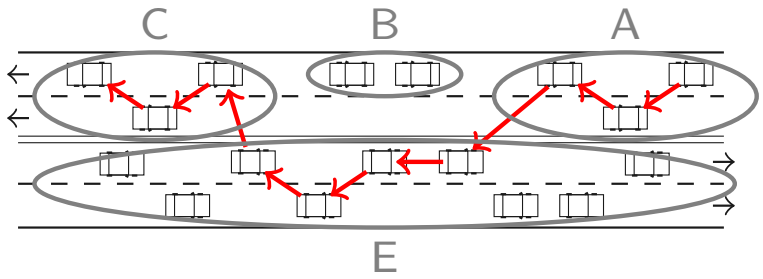


Used frequently for simulations in wireless networks.

Some facts about random UDGs (as $n \to \infty$)

- Penrose'97 (also Gupta & Kumar'98):
  Connectivity threshold $r \to \sqrt{\frac{\ln n}{\pi n}}$
- Penrose'03: $\Theta(n)$-size component threshold $\to c/\sqrt{n}$ for some c (≈ 2.07)
- Díaz et al.'09: At connectivity threshold, $P(\exists$ component of size $i)$

Vehicular networks



Important literature, using both static or dynamic graphs.

Led to theoretical results of independent interest

Ex: Percolation in grids

(Shioda et al.'08)

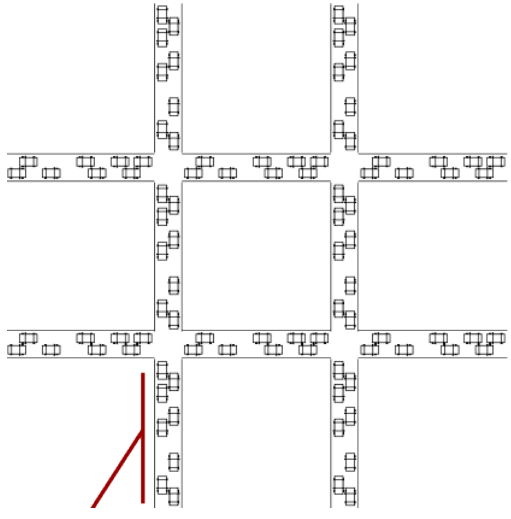
Model:

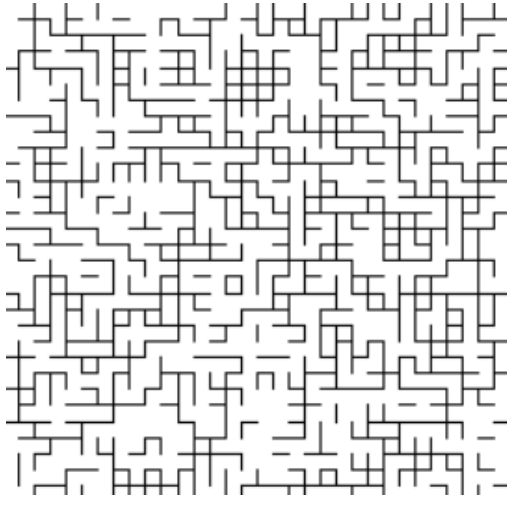
Infinite Manhattan grid

Theorem:

$P(\text{segment is connected}) \geq 1/2$

$\Rightarrow \exists$ component of size ∞

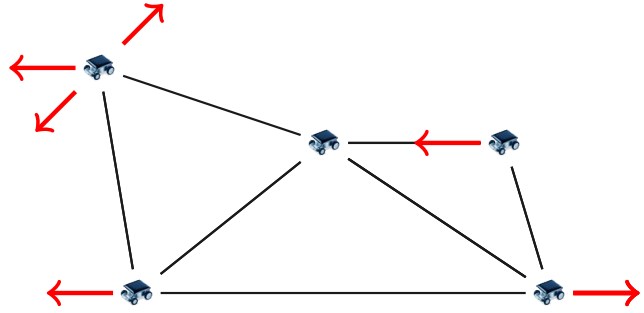




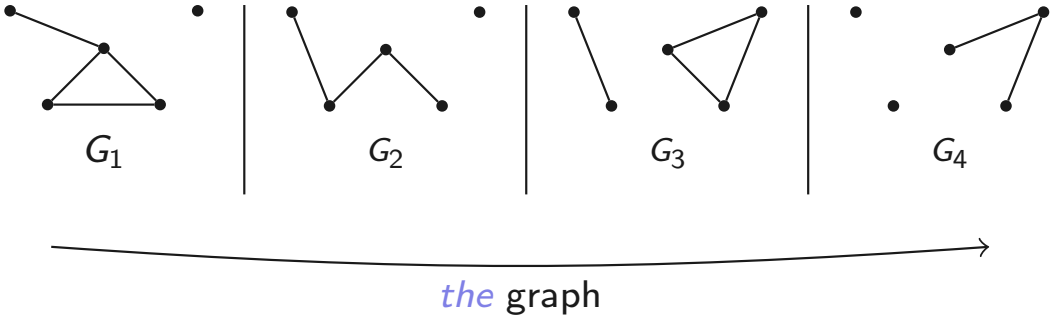
Dynamic Graphs (*a.k.a.* time-varying graphs, temporal graphs, evolving graphs, ...)

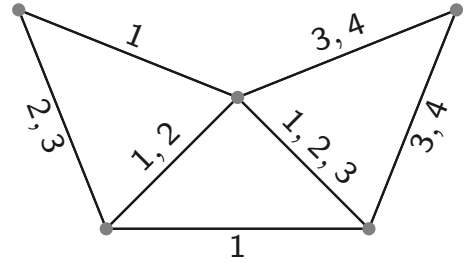
Ex:      

Exemple of scenario



Graph representation (for example)





More later...

(Part III)

Topics of independent interest

Random walks

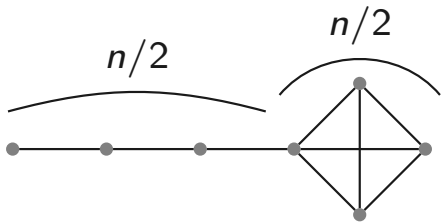
Start at a given vertex. Move to a randomly chosen neighbor. Repeat.

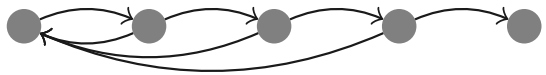
Some common measures

- Cover time: how long before all vertices are visited (at least once)
- Return time: how long before the walks returns to its initial position
- Hitting time: how long before a given vertex is visited
- Mixing time: how long before the position no longer depends on initial position

Some bounds on cover time (worst case)

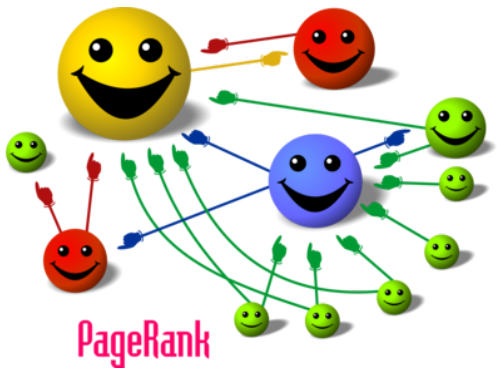
- Undirected graphs: $\Theta(n^3)$ steps
- Directed graphs: 2^n steps!





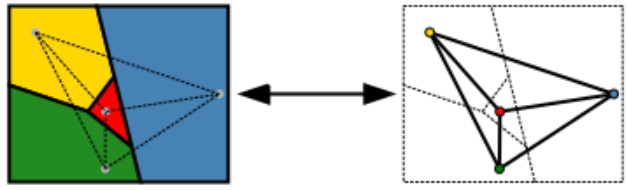
Applications

- Economics, physics, genetics, algorithms, ...
- Ex: Page rank algorithm (initial version)



Graph coloring (the four color theorem)

- 1852: Francis Guthrie (then a botanist) notices that **four** colors are enough to color the map of England's counties.



- 1879: A. Kempe proves this is true whatever the map... bug found by P. Heawood 11 y. later
- 1880: P. Tait proves it differently... bug found by J. Petersen 11 y. later
  $\rightarrow$ Formulated using graphs: "2D map = planar graphs"
- 1890: P. Heawood proves that **five** colors are enough (using Kempe's arguments)
- 1960s: H. Heesch starts using computers to search for a proof
- 1976: K. Appel and W. Haken succeed: **Every planar graph is four colorable!**
  $\rightarrow$ reduction from ∞ to 1476 possible cases, all tested using a computer.
- 1996: N. Robertson, D. Sanders, P. Seymour reduce it to 633 cases
- 2005: G. Gonthier certifies the proof using Coq $\square$

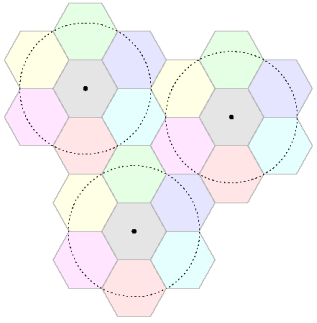
Graph coloring (besides the four color theorem)

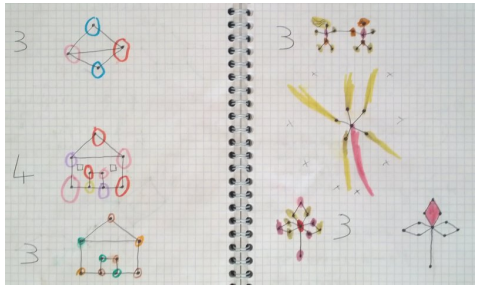
Several variants

- Vertex coloring: neighbors must have different colors
- Edge coloring: every two adjacent edges must have different colors
- List coloring: choose among a given list of possible colors (vertex or edge)
- 2-hop coloring: 2-hop neighbors must have different colors

Many applications (e.g. telecom)

- Wireless communications
- Mutual exclusion in general, scheduling, ...
- Unplugged computer science



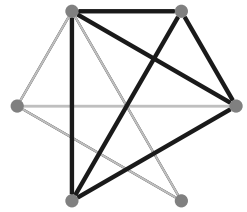


Theoretical depth

- Connects with a number of problems and structural properties (e.g. clique, minors, cycles, symmetries)
- Many open questions and ongoing projects
- MacArthur price (Maria Chudnovsky) in 2012

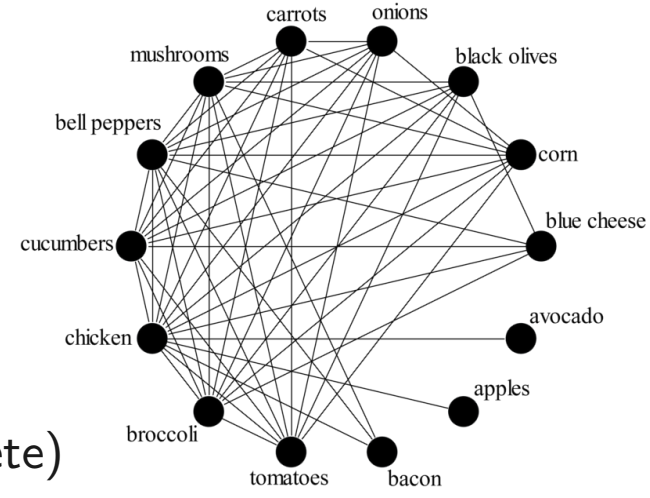
Cliques (and Ramsey theory)

Clique = Complete graph within another graph
(complete subgraph)



Applications / Connexions

- Community detection
- Graph Coloring
- Cooking
- ...



MaxClique problem

Ex: Is there a clique of size k here? (NP-complete)

Best exact algorithm by Mike Robson $O^*(1.1888^n)$

Classical relaxations in social network analysis

- k-plexes
- s-clubs
- ...

Ramsey theory

- See exercise...

Ramsey theory (exercises)...

Any two persons either know each other or they don't (the relation is symmetrical). We say that three persons know each other if they all know each other pairwise. We say that they don't known each other if none of them knows any of the others. Consider the following declaration: "In a group of four persons, there is always three persons who know each other or three persons who don't know each other."

- Is this true in a group of 5 persons?
- Is this true in a group of 6 persons?

Hint: You may represent the group of people as a complete graph whose *edges* are colored with two colors. The question becomes whether you can color the edges without creating a monochromatic triangle.

Proving that a problem is NP-hard / NP-complete

Definitions (classes of problems)

1. NP: Easy to verify if answer is YES
   (e.g. easy to verify that a given coloring is valid (proper) and uses 3 colors)
2. NP-hard: At least as hard as any other problem in NP
3. NP-complete: Both NP-hard (bad news) and in NP (good news)

How to prove that a problem is NP-complete?

Example: MaxIndependentSet

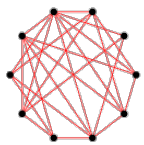
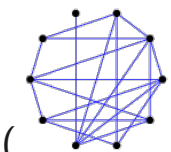
Input: A graph G, an integer k

Question: Does G admit an independent set of size k?

1. Easy to check that a given set is independent and of size k (thus the problem is in NP ✓)
2. How to prove that the problem is NP-hard?

   $\rightarrow$ Take another NP-hard problem, and reduce it to this one (in polynomial time)

Ex: MaxClique reduces to MaxIndependentSet as follows:

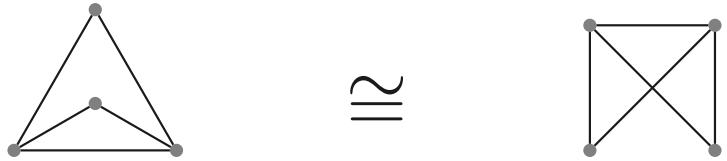
MaxClique ( , k) $\overset{\text{complement}}{\rightarrow}$ MaxIndependentSet ( , k) □

Asymptotic notations (O, Ω, Θ)

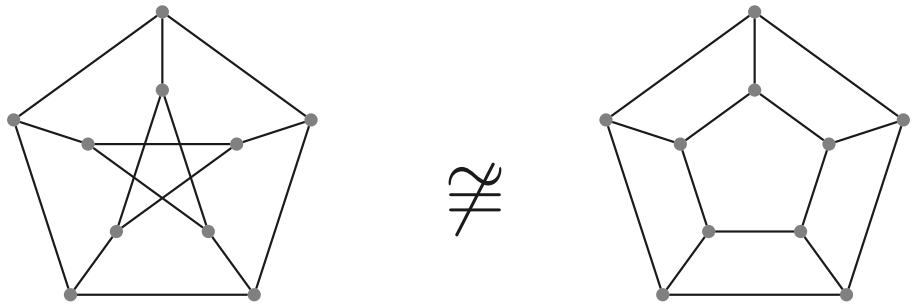
	True	False
$4n^2 - 5n + 1 = O(n^2)$	✓	
$4n^2 - 5n + 1 = \Theta(n^2)$	✓	
$n \log n = \Omega(n)$	✓	
$n \log n = O(n^2)$	✓	
$n \log n = \Theta($ _____ $)$?	N/A	N/A
$n \log n^2 = \Theta($ _____ $)$?	N/A	N/A
$500 = \Theta(1)$	✓	
$17n^2 + 3 = O(n^{\Theta(1)})$	✓	
$\sqrt{n} = O(n)$	✓	
$n! = \Omega(2^n)$	✓	

The graph isomorphism problem

Given two graphs G_1 and G_2, are they "identical"?



Usually defined in term of the existence of a bijection between the vertices of G_1 and the vertices of G_2 that preserves adjacency (and non-adjacency)



- Candidate for NP-intermediate
- Strong connexions with another problem: Graph automorphism (given a graph G, is there non trivial symmetries in G)
- Generalizes as subgraph isomorphism, but NP-complete (why?)
- Applications in topology embedding (e.g. for cloud computing)