Introduction to Graph Mining Algorithms

Basic Definitions from Graph Theory

Andrea Marino

PhD Course on Graph Mining Algorithms,
Università di Pisa

Pisa, February 2018

Today

1. Basic definitions of graph theory

Degree of Separation and Small World

2. Computing the **diameter** in huge graphs
3. Computing the **distance distribution** in huge graphs easily.
4. **Sketches and probabilistic counting**: distance distribution and other applications.

Node Properties

5. **Clustering Coefficient**: counting the number of triangles.
6. **Centrality**: computing closeness and betwennees.

Grouping Nodes

7. **Clustering**: overview, algorithms and spectral clustering.
8. **Finding Patterns** in graphs with applications to community detection: listing cliques.

Social Networks

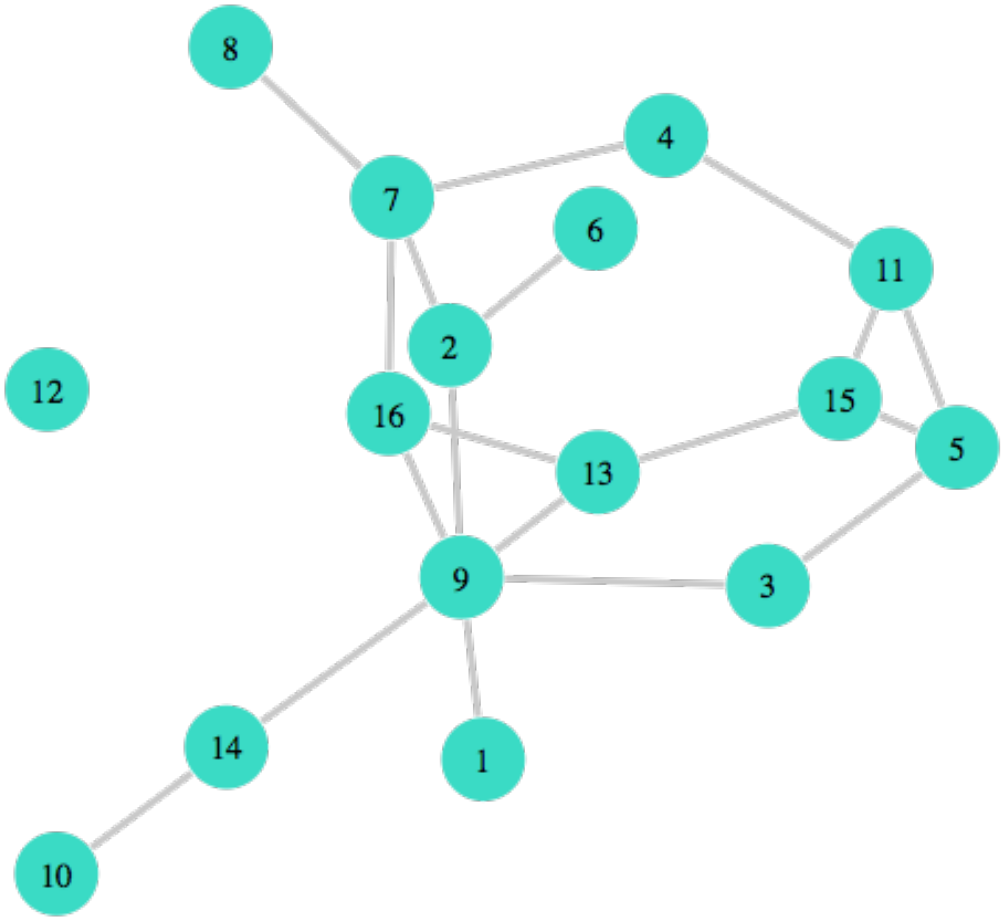
- Our life is full of binary relations of very different types.
- Some examples are:
  - *communications* (like talking at the phone or sending a message on WhatsApp),
  - *collaborations* (like co-authoring a paper or co-playing in a movie),
  - *memberships* (like being the member of the same university department),
  - *evaluations* (like "liking" a picture on Instagram),
  - *dependencies* (like citing a paper or being a child), and
  - *transfers* (like lending or borrowing money).
- These relationship induce connections between the two involved agents.
- These connections are the *edges* of a graph whose *nodes* are the agents themselves.

From Social Relations to Graphs

- We can make use of the many mathematical and computer science tools that have been developed in the field of graph theory.
- We can analyze mathematical properties of a graph.
- We can design, analyze, and implement efficient algorithms computing these properties.

The ultimate goal

Deriving conclusions relevant from a social science point of view such as, for instance, the identification of important agents or group of agents.



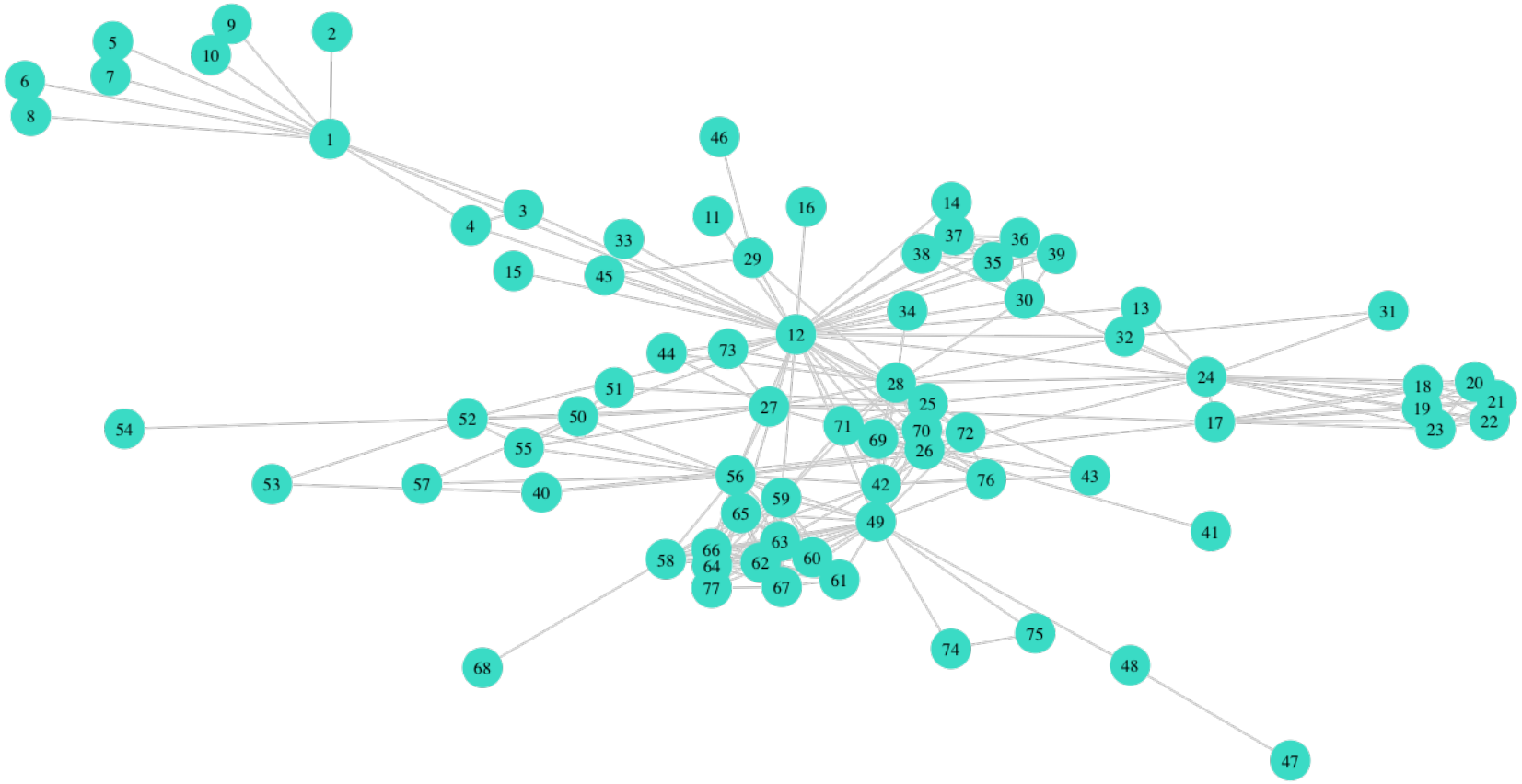
The social network whose agents are 16 Florentine families of the 15th century. Two families are related if one member of one family has been married to one member of the other family.
Can you guess which node corresponds to the Medici family?

Part I

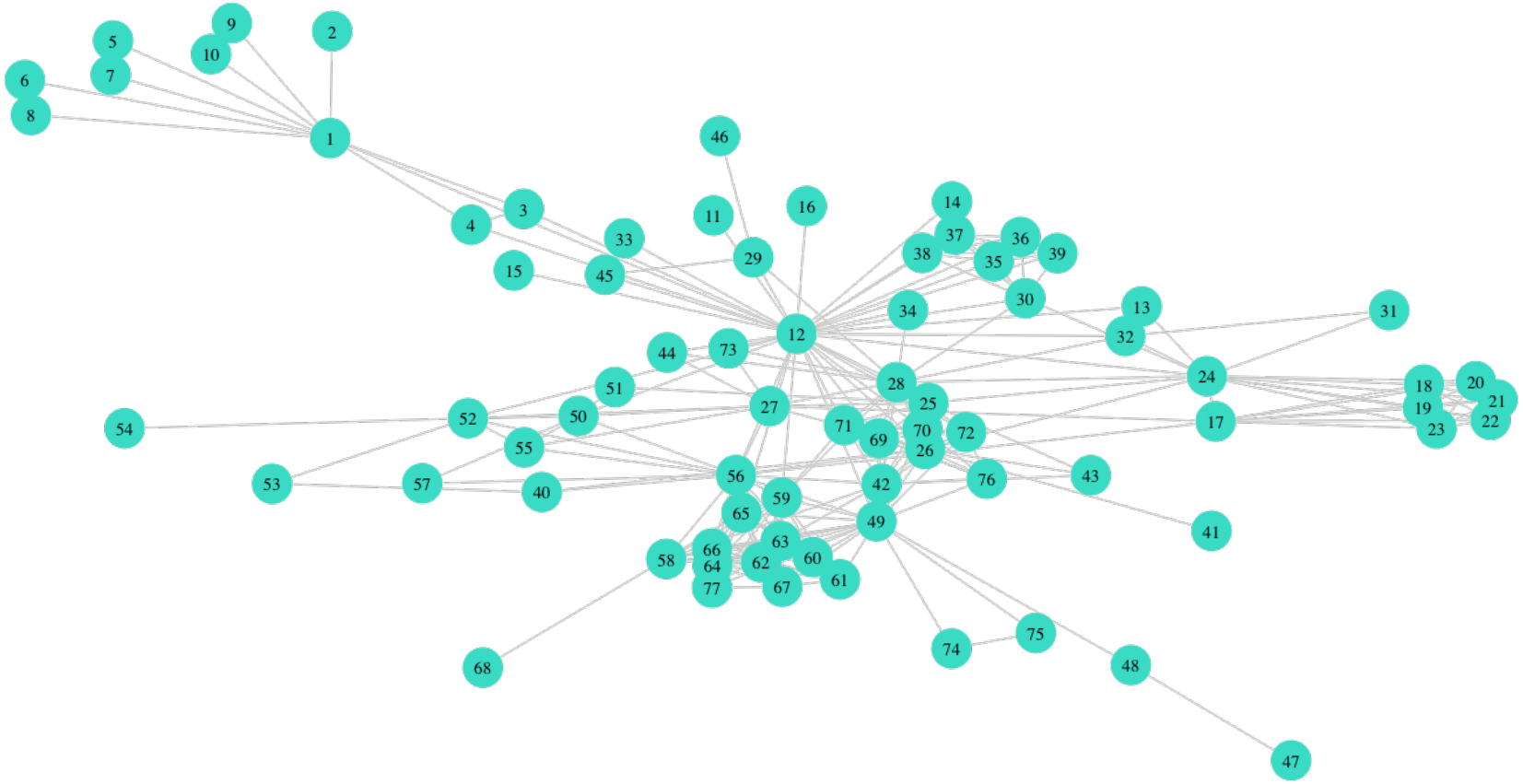
The graph of Les Misérables

The Story

- Les Misérables is one of the greatest French historical novel, written by Victor Hugo and first published in 1862.
- Set in early 19th-century France, it is mostly the story of ex-convict Jean Valjean (and of his quest for redemption), who is relentlessly tracked down by a police inspector named Javert.
- Along the way, Valjean and a slew of characters are swept into a revolutionary period in France, where a group of young idealists make their last stand at a street barricade.
- Many characters which appear and disappear through the more than one thousand pages of the book.

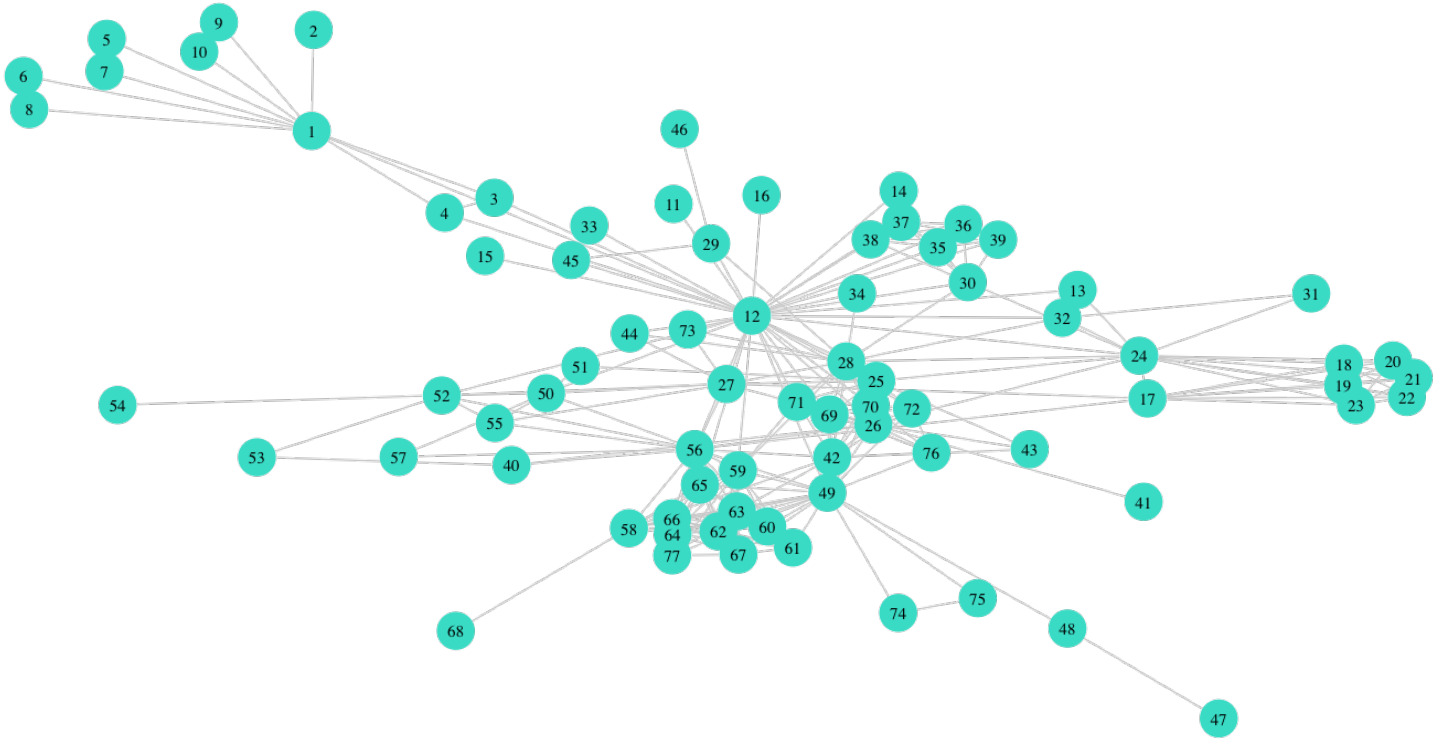


Two characters are related if they both appear in at least one chapter.
The corresponding graph has 77 nodes and 254 edges.



There is a very *central* node, that is, the node labeled 12. Not surprisingly, this node corresponds to Jean Valjean.

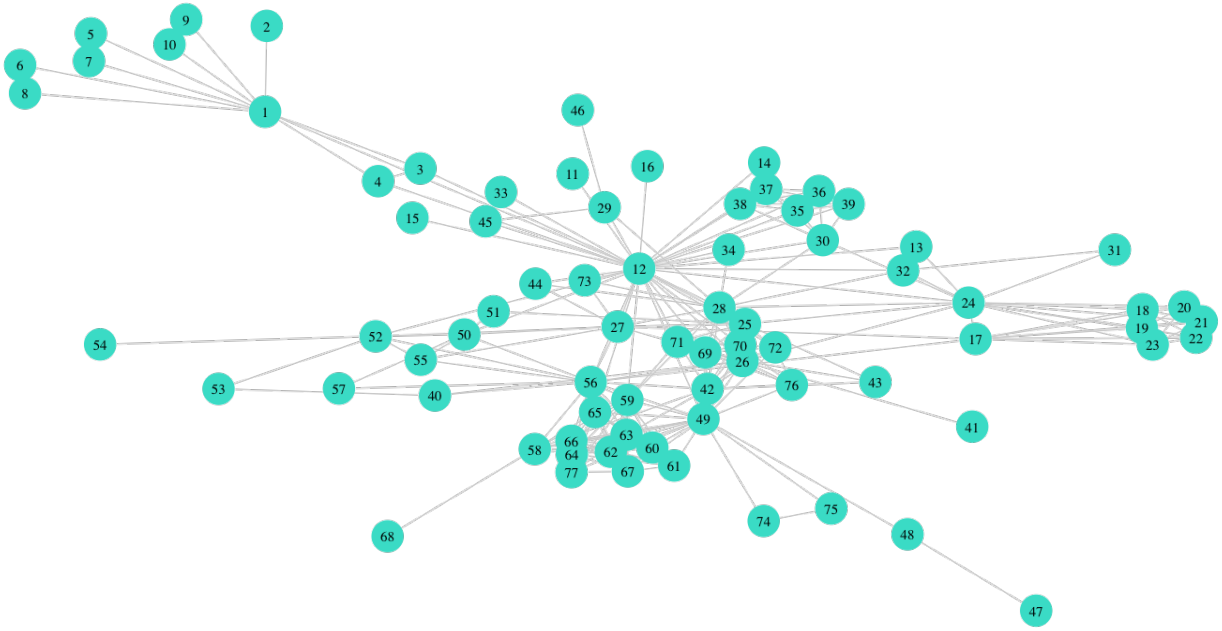
Do not forget, do not ever forget, that you have promised me to use the money to make yourself an honest man.



The node 1 plays some special role: it is a connection point between a subset of nodes and the rest.

This node corresponds to a key character, i.e. the Bishop Myriel[1]

[1] who, instead of punishing Valjean for robbing him, sees an opportunity to change Valjean's life by letting him go.



- Similarly, the two nodes labeled 24 and 17 connects a subset of nodes to the rest. These two are Fantine and Tholomyes, the parents of Cosette[2]
- Fantine is a key character of the book: a "miserable" human being who is punished all out of proportion for having a love affair without being married.

[2] who will be adopted by Jean Valjean when Fantine dies.

What else can we get from the graph representation of the book of Victor Hugo?

- The figure does not help us too much, since the center of the figure itself seems just a mess of nodes and edges.
- It is now time to take advantage of the notions and the algorithms developed in the field of graph theory.

Part II

Graph theory basic notions

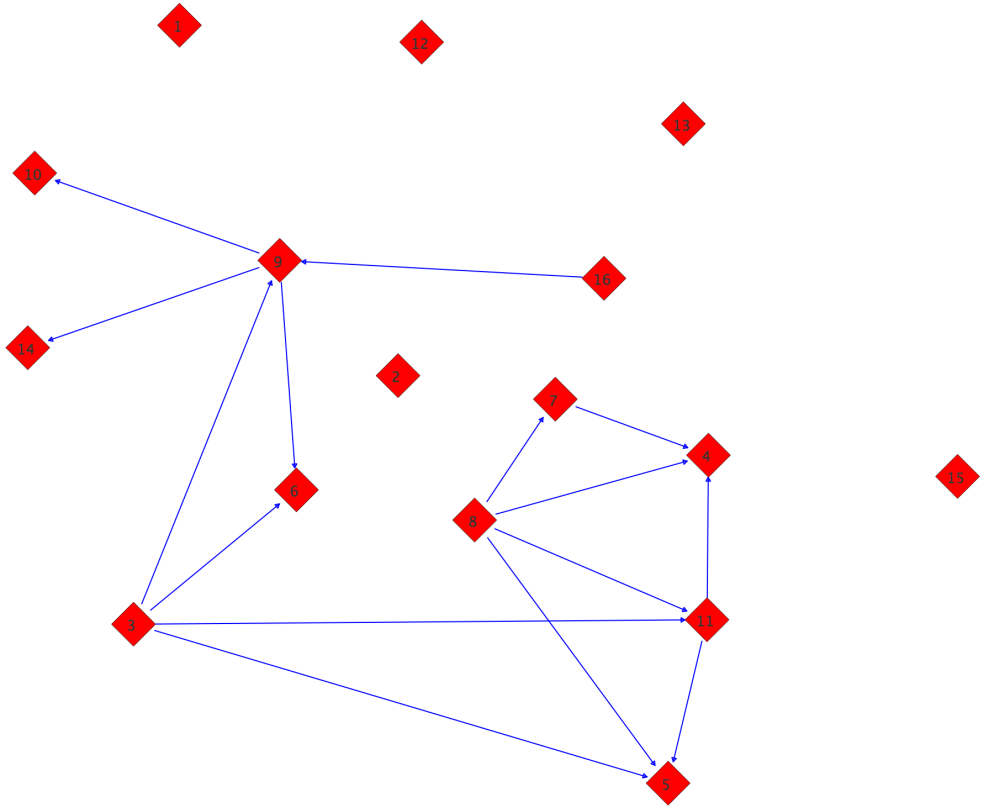
Edge Direction

Undirected Whenever two characters of Les Misérables appear in a chapter a symmetric relation is created between them:

- if character x appears in the same chapter of y, then y also appears in the same chapter of x.
- We denote such edge as the set $\{x, y\}$.

Directed In other cases, it might be that the relation represented by the graph is not symmetric:

- for instance, if a paper x cites another paper y, most likely the paper y does not cite the paper x since it has been published before.
- We denote such edge as the pair (x, y).



The business social network whose agents are 16 Florentine families of the 15th century.
The directed relations are business ties (specifically, recorded financial ties such as loans, credits and joint partnerships).

Density

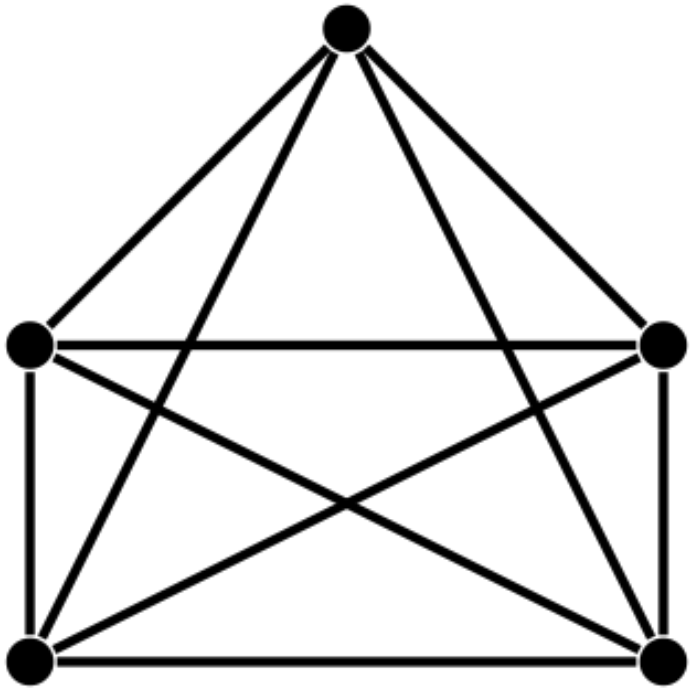
The graph corresponding to the social network of the characters of Les Misérables has 77 nodes and 254 edges.

Is this a "dense" graph? That is, are the nodes connected (almost) as much as they could be?

Let's calculate what is the maximum number of edges a graph of n nodes can have.

- If the graph is directed, then any node can have at most $n-1$ edges towards the other nodes of the graph: hence, the maximum number of directed edges is $n(n-1)$.
- If the graph is undirected, then this number has to be divided by two.

When a graph has all the possible edges, it is called a *complete graph* or *clique*.



An undirected complete graph or clique with 5 nodes. The number of edges is equal to $\frac{5 \cdot 4}{2} = 10$.

Definition

The *density* of a graph with n nodes is the number of edges in the graph divided by the number of edges in the clique of n nodes.

- In the case of the graph of Les Misérables, the density is

$$\frac{254}{(77 \cdot 76)/2} = \frac{254}{5852/2} = \frac{254}{2926} \approx 0.09.$$

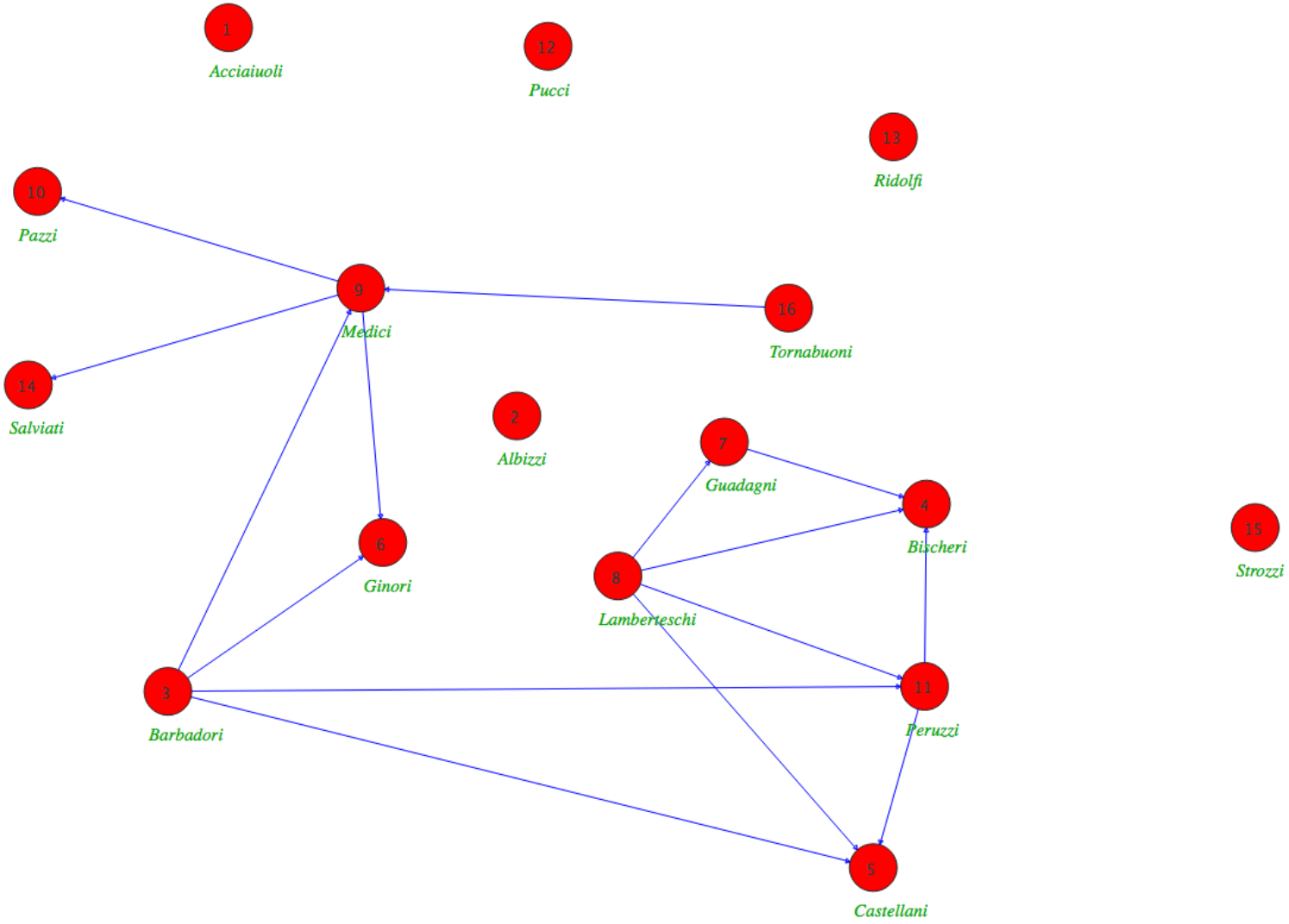
- This is close to 0, so that we can conclude that this graph is not dense, that is, it is *sparse*.
- This is the most frequent situation arising when dealing with real-world social networks.

Node labels

Labeled In the graph corresponding to Les Misérables, the names of the characters are associated to the corresponding nodes. This information is called the *label* of the node.

Unlabeled If we forget about these names and no additional information is associated with the nodes.

Business Social Network for 16 Florentine families of the 15th century

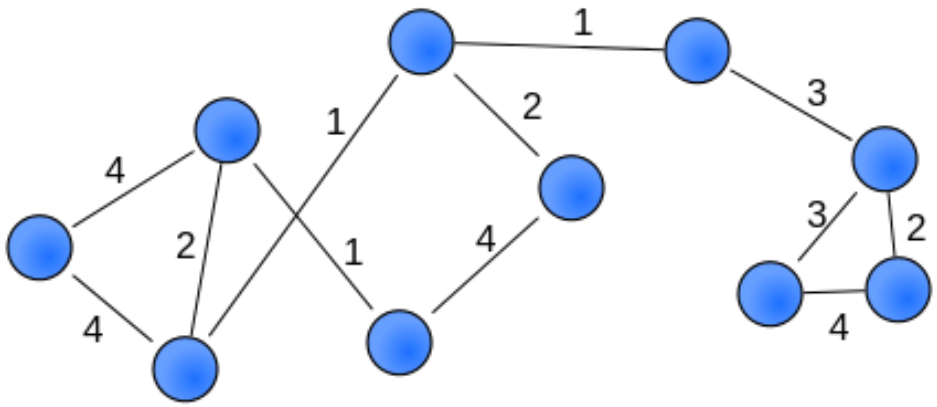


At each node is associated a label, i.e. the name of the family.

Edge weights

Unweighted In the graph corresponding to Les Misérables no additional information is associated with the edges.

Weighted However, we could have considered also the number of times two characters appear in the same chapter and associate such integer number to the corresponding edge. This information is called the *weight* of the edge.



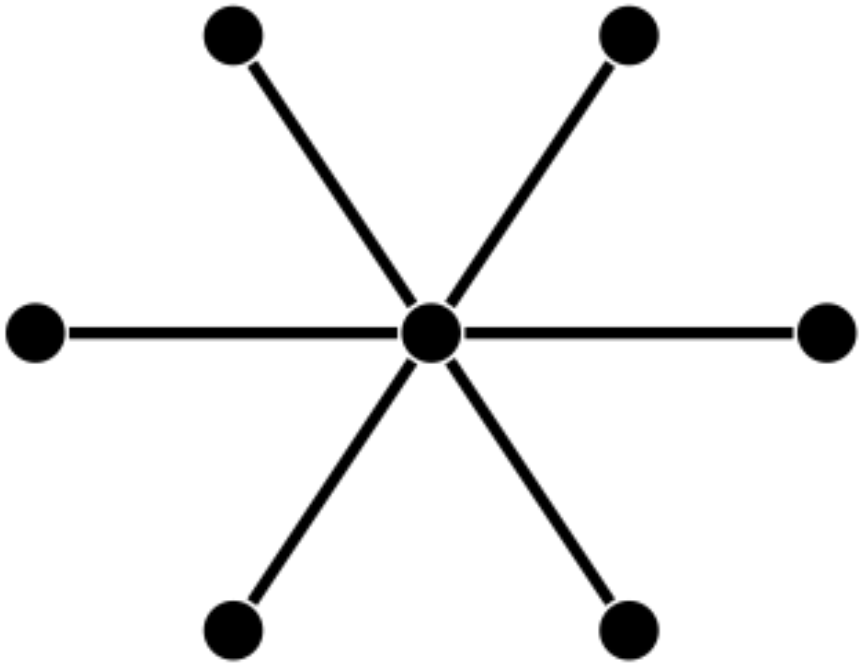
Degree

Definition

The *degree* of a node is the number of edges adjacent to it, that is, the number of its *neighbors*.

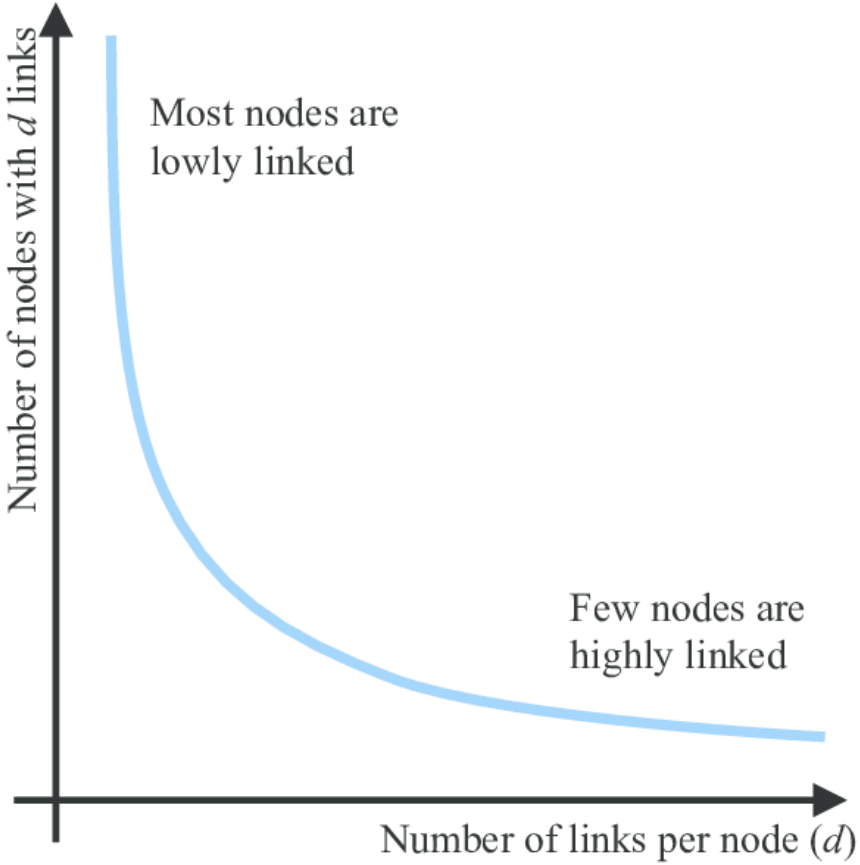
- In the case of the graph corresponding to Les Misérables, we can assume that the most important character corresponds to the node of highest degree.
  - This node is the one associated to Jean Valjan, who is the main character of the book: the degree of this node (the one labeled 12) is 32.
  - The second most popular character is Gavroche (node labeled 49), a boy who lives on the streets of Paris and plays a short yet significant role.

- In a clique, all nodes have degree $n - 1$
- In a *star*, that is a node x connected to all the other nodes and with no other edges, x has degree $n - 1$ and all the other edges have degree 1.



- In the case of directed graphs, we distinguish between
  - the *indegree* of a node (that is, the number of edges "entering" the node) and
  - the *outdegree* of a node (that is, the number of edges exiting the node).
- Sometimes, it might be convenient to "symmetrize" the edges by making them undirected and to consider the degree in the resulting undirected graph.

Power Law



In social network, it seems that the degree distribution $f(d)$, i.e. the frequency of the degree d, follows a power law:

$$f(d) \propto \frac{1}{d^{\gamma}},$$

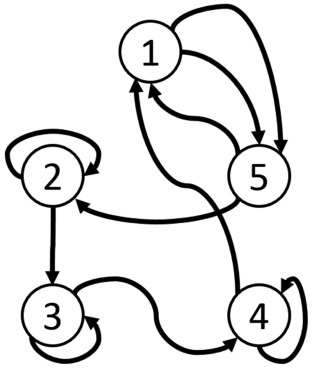
for some constant γ, typically $1 \leq \gamma \leq 3$.

Simple graphs

- In the graph of Les Misèrables we have not included an edge between one node and itself (the same character appears in the same chapters).
  - These kind of edges (of the form $\{x, x\}$ or (x, x)) are called *loops*.
- A graph can have several edges between the same two nodes. We call these multiple connections *multiedges*.

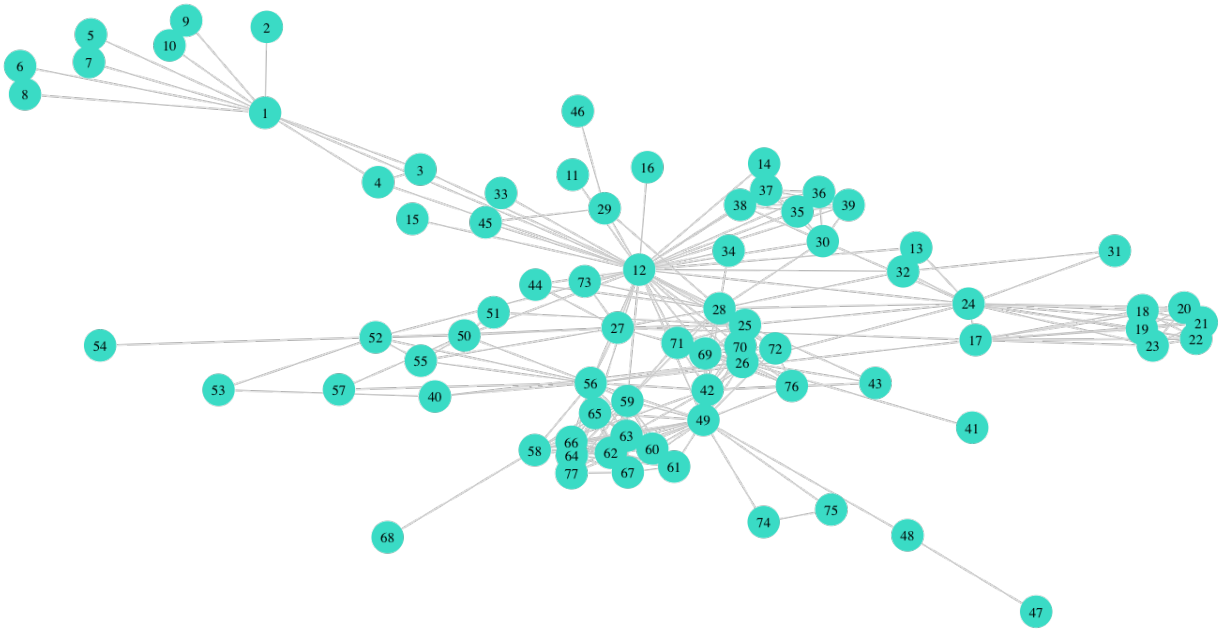
Definition

If a graph has no loop and has no multiedge, it is said to be *simple*.

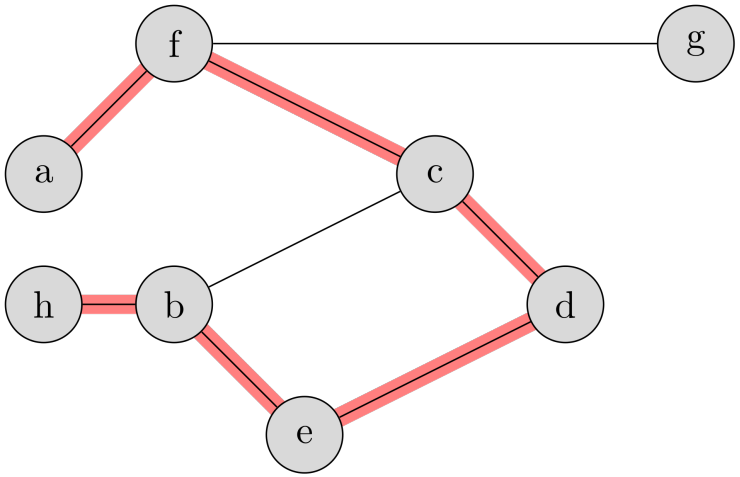


Paths

- If a node x is connected to a node y and y is connected to another node z, then we say that there is a *path* from x to z.
- A path can be formed by more than two edges: for example, in the graph of Les Misérables there is a path from from node 1 to node 23 passing through nodes 12 and 24.



- The number of edges in a path is called the *length* of the path.
- In many cases, we are interested in the path with the minimum length, which is called a *shortest path*. The length of a shortest path from a node x to a node y is called the *distance* from x to y and is denoted by $d(x, y)$
  - If there is no path from x to y, then we set $d(x, y) = \infty$.
  - If the graph is weighted, then the length of the path is equal to the sum of the weights of the edges in the path itself.



Small World

In social networks, the diameter, that is the maximum distance among all the nodes of the graph, and the average distance are low compared to the size of the network.

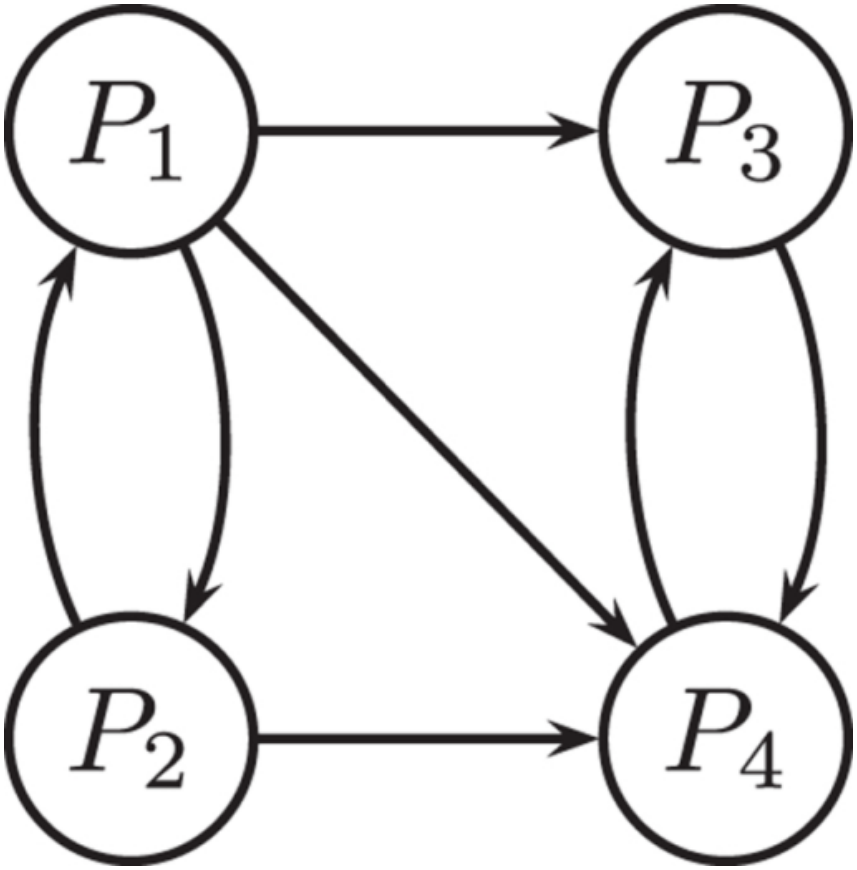
- The intermediate nodes are called *degrees of separations*.
- Milgram experiment, to upper bound degrees of separations in real-life.
- In the case of the graph of Les Misérables the diameter is 5 and the average distance is 2.4.
- In the case of Facebook in 2011 (721.1M nodes and 68.7G edges) the diameter was 41 and the average degrees of separation were 4.74.
- Inspiration for many games:
  - Connecting one given actor to Kevin Bacon in IMDB network
  - Going from one Wikipedia page to another one in few hops.

Connectivity

Definition

- If, for any pair of nodes, there exists a path between them, then the graph is said to be *connected*.
- For directed graphs, a graph is *strongly connected* if, for any pair of nodes x and y, there exists a path from x to y and vice versa.
- If a directed graph is not strongly connected but removing the directions of the edges the resulting graph is connected, then we say that the graph is *weakly connected*.

For example the graph of Les Misérables is connected.



A directed graph which is not strongly connected.

The giant component

Definition

A *giant component* is a (strongly) connected component of a given (directed) graph that contains a constant fraction of the entire graph's nodes.

- It is very common in the case of social networks that there always exists a giant component.
- The connectivity property by itself does not really identify interesting communities of the social network.
- A single bridge make collapsing components.

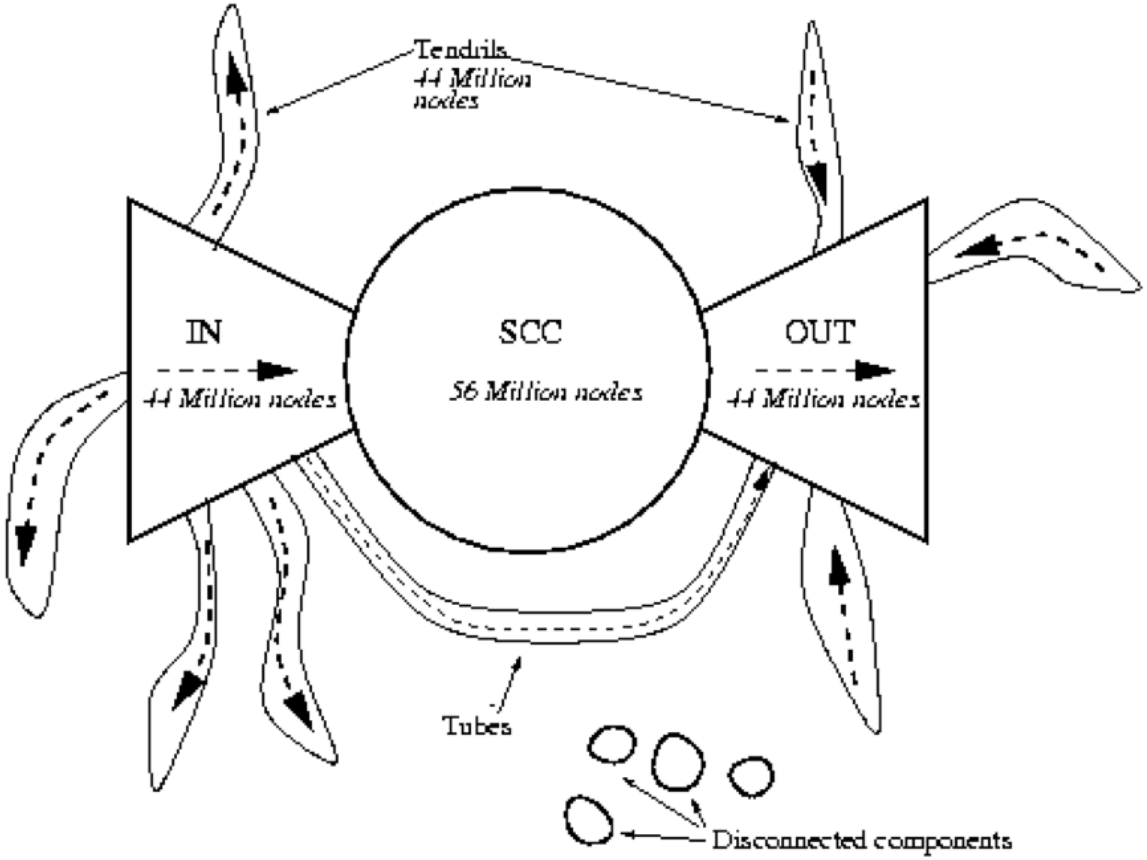
Connected Components in the Web

- In the case of the web directed graphs, in which nodes are web pages and edges are hyperlinks connecting one page to another, it has been observed the *bowtie phenomenon*.
- A detailed analysis in

  Andrei Z. Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, Janet L. Wiener: Graph structure in the Web. Computer Networks 33(1-6): 309-320 (2000)

The bowtie structure of the web

High-level view of the web structure, based on its connectivity properties and how its strongly connected components fit together.



This study has been replicated on other larger snapshots of the web and it still a useful way of thinking about giant directed graphs in the context of the web and more generally.

Part III

Representing a graph

Adjacency matrices

- The *adjacency matrix* A of a graph is a bidimensional table with as many rows as the number of nodes and as many columns as the number of nodes.
- The element of the table at row x and column y (denoted by $A[x][y]$) is 1 if there exists an edge connecting x to y, otherwise it is 0.
- If the graph is not oriented, the adjacency matrix is symmetric, in the sense that $A[x][y] = A[y][x]$.
- The advantage of this representation is that it allows us to determine very quickly whether two nodes are neighbours.

- An adjacency matrix requires space quadratic in the number of nodes, independently from the number of edges in the graph.
  - In the case of the graph of Les Misérables, even if we have only 254 edges, the adjacency matrix requires 5929 elements (or 29126 if we use the symmetry of the graph and we represent only the upper right part of the matrix).

Actor/Actor	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
5	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0
6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
9	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
14	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
15	0	0	0	1	1	0	0	0	0	0	1	0	1	0	0	0
16	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0

The adjacency matrix of the Florentine families social network, where two families are related if one member of one family has been married to one member of the other family.

Adjacency lists

- The *adjacency list* representation associates to any node x a list of its *neighbors*, that is, a list of all nodes y such that there exists an edge from x to y.
- By using an adjacency list representation, the amount of used space is proportional to the total number of edges in the graph.

For example, in the case of the graph of Les Misérables, the used space is approximately 254, which is the number of edges.

- One disadvantage of the adjacency list representation is the fact that deciding whether a node x is neighbour of y costs a time proportional to the degree of the node itself.
- In the case of social networks this is a quite negligible cost, since the average degree of a social network is usually very small compared to the number of nodes.
- For instance, in the case of the graph corresponding to Les Misérables the average degree is equal to $\frac{2 \cdot 254}{77} \approx 6.6$.
- We will always use adjacency lists.

$1,1$	9	$9,6$	1,2,10,13,14,16
$2,3$	6,7,9	$10,1$	14
$3,2$	5,9	$11,3$	4,5,15
$4,3$	7,11,15	$12,0$	
$5,3$	3,11,15	$13,3$	9,15,16
$6,1$	2	$14,2$	9,10
$7,4$	2,4,8,16	$15,4$	4,5,11,13
$8,1$	7	$16,3$	7,9,13

- The adjacency lists of the Florentine families social network, where two families are related if one member of one family has been married to one member of the other family.
- The blocks on the left of the arrows contain a node and the size of its adjacency list, while the blocks on the right of the arrows contain the list of neighbors of the node.

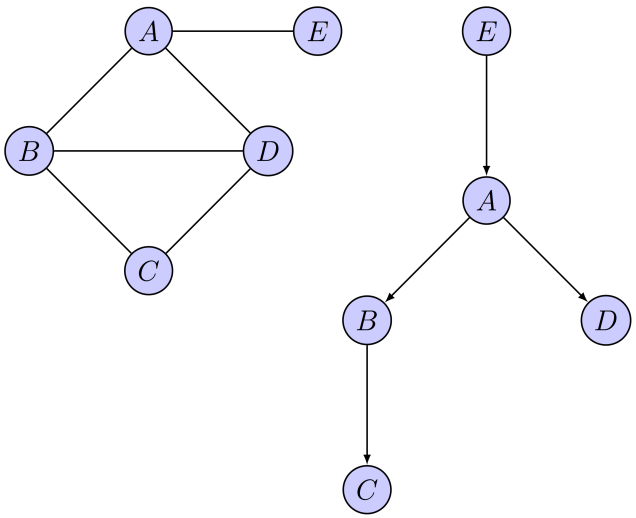
Part IV

Traversing a Graph

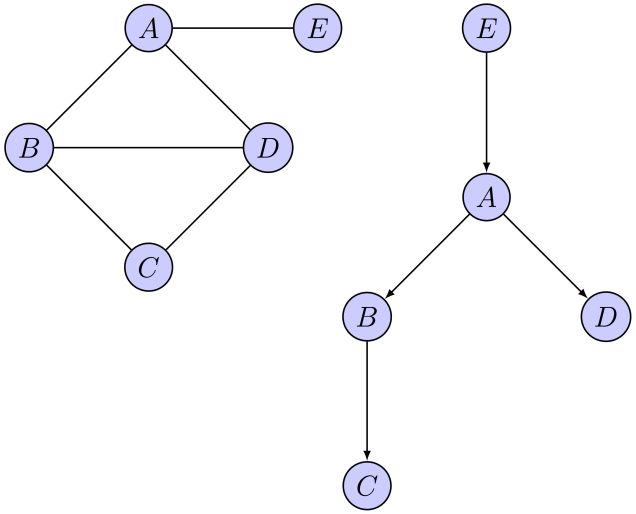
Breadth-first search from a source node

The key idea of the *breadth-first search* is to mark each node which has already been visited. Nodes that are visited for the first time are put into a *queue*, which is a first-in first-out data structure.

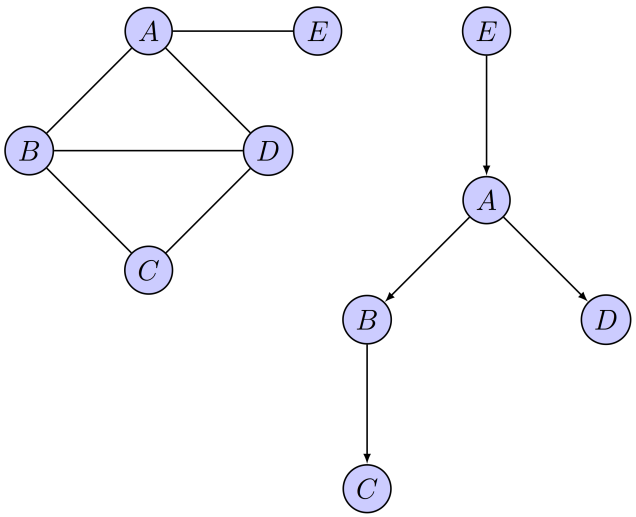
- At the beginning the source node is marked, and inserted into the queue.
- Until this queue is not empty, we "serve" the first node in the queue, that is:
  - we examine all its neighbors and,
  - for each of them, if it is not marked, we set its state to marked and we insert it into the queue.




```
queue = [s]
marked[s]=true
while queue.notempty():
    v = queue.pop()
    for u in v.neighbours:
     if not marked[u]:
         marked[u]=true
         queue.append(u)
```



- If each time a node is inserted into the queue, we also associate to this node its *parent*, which is the node whose neighborhood exploration led us to insert the node in the queue, we obtain the *breadth-first search tree*.
- One property of this tree is that the nodes at level i are all the nodes which are at distance i from the initial node.




```
queue = [s]
marked[s]=true
dist[s]=0
pred[s]=s
while queue.notempty():
    v = queue.pop()
    for u in v.neighbours:
     if not marked[u]:
          marked[u]=true
          pred[u]=v
          dist[u]=dist[v]+1
          queue.append(u)
```

The BFS requires time proportional to the number of edges, since each edge is taken into consideration exactly two times.

Applications:

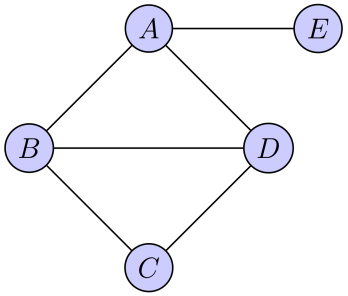
- By executing a BFS from any node of the graph, we can both compute the *diameter* of the graph, that is, the maximum distance between two nodes of the graph, and the average distance of the graph.

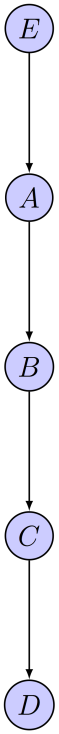
Applications to Connected Components in Undirected Graphs

- The BFS also allows us to determine whether the graph is connected.
  - It is connected if and only if all nodes of the graph are included in the BFS tree.
  - This is the case, for example, of the graph of Les Misérables.
- If the graph is not connected, the breadth-first search tree contains a *connected component* of the graph.
- In order to find the other connected components we can simply execute again the BFS starting from any of the nodes which has not been reached until all nodes have been included in at least one connected component.

Depth-first search from a source node

- The *depth-first search* is similar to the breadth-first search.
- Whenever a node x is reached for the first time and it is marked as visited, the depth-first search recursively invokes itself with source node x.
- Same procedure of before, using a stack instead of a queue.
- Same time complexity.
- Also the depth-first search generates a tree, which is called a *depth-first search tree*.






```
stack = [s]
marked[s]=true
while stack.notempty():
    v = stack.pop()
    for u in v.neighbours:
     if not marked[u]:
          marked[u]=true
          stack.push(u)
```

Thanks

Part of these slides are based on a chapter written by Pierluigi Crescenzi for his course "Algorithms for Graph Mining".