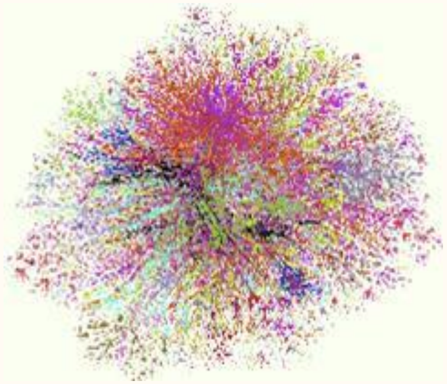
Advanced Topic - Graph Data Analytics

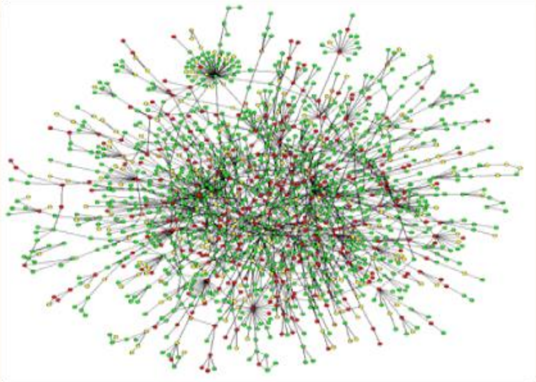
Why Graphs ?

Common model across different fields, they use **graph databases**.



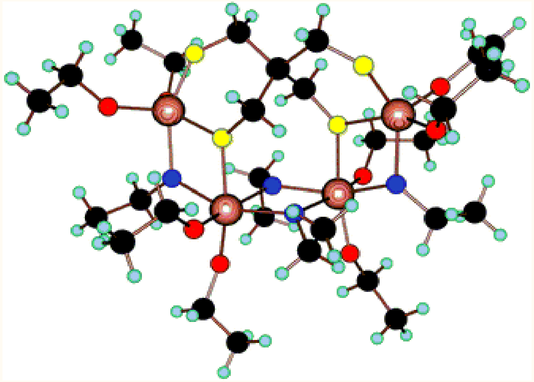
Web Graph

Social Network

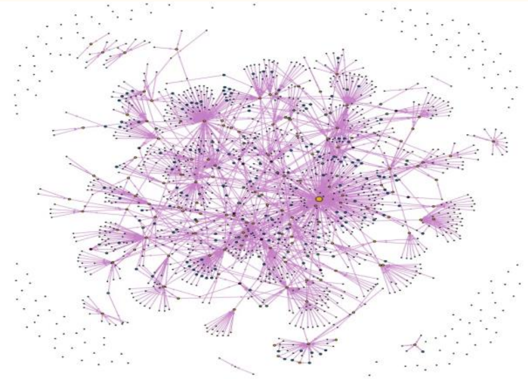


Protein Interaction Network

Road Network

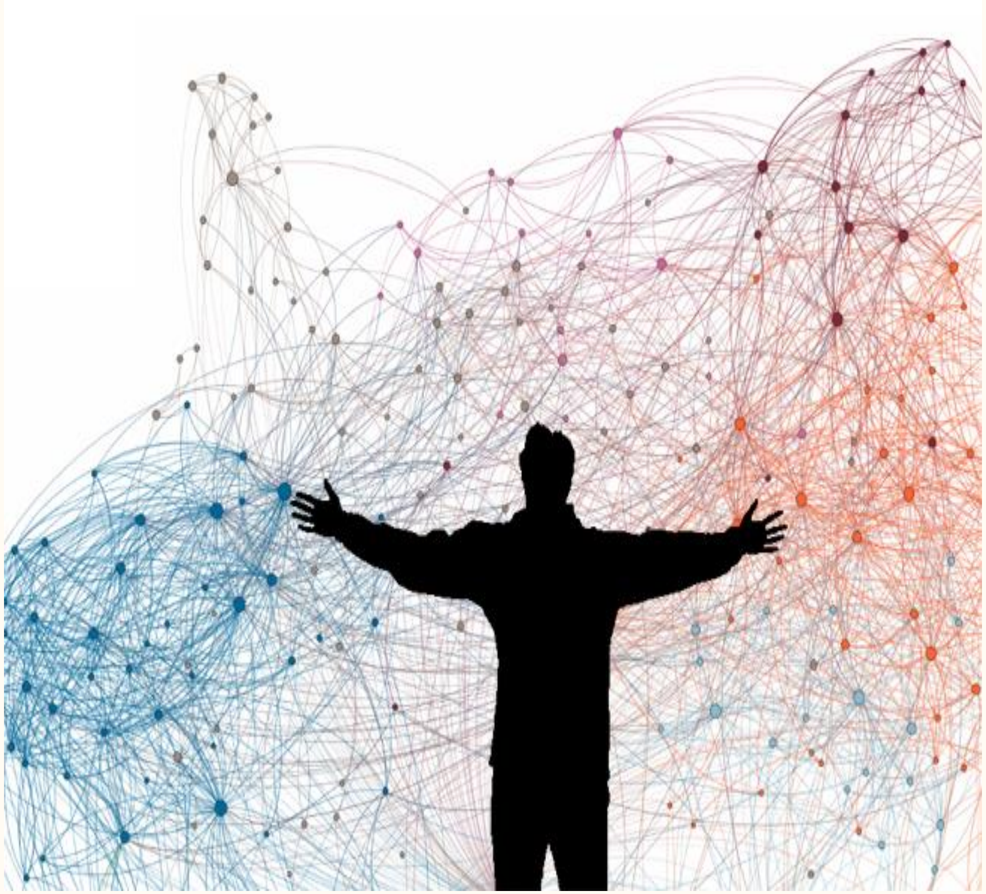


Chemical Compound



Ontology Graph

Social Networks



The Scale/Growth of Social Networks

Facebook statistics

- 829 million daily active users on average in June 2014
- 1.32 billion monthly active users as of June 30, 2014
- 22% increase in Facebook users from 2012 to 2013

Facebook activities (every 20 minutes on Facebook)

- 1 million links shared
- 2 million friends requested
- 3 million messages sent

http://newsroom.fb.com/company-info/

http://www.statisticbrain.com/facebook-statistics/

facebook

The Scale/Growth of Social Networks

Facebook statistics

- 1.04 billion daily active users on average in Dec 2015
- 1.59 billion monthly active users as of Dec 31, 2015
- 12% increase in Facebook users from 2014 to 2015

Facebook activities (every 20 minutes on Facebook)

- 1 million links shared
- 2 million friends requested
- 3 million messages sent

http://newsroom.fb.com/company-info/

http://www.statisticbrain.com/facebook-statistics/

facebook

The Scale/Growth of Social Networks

Facebook statistics

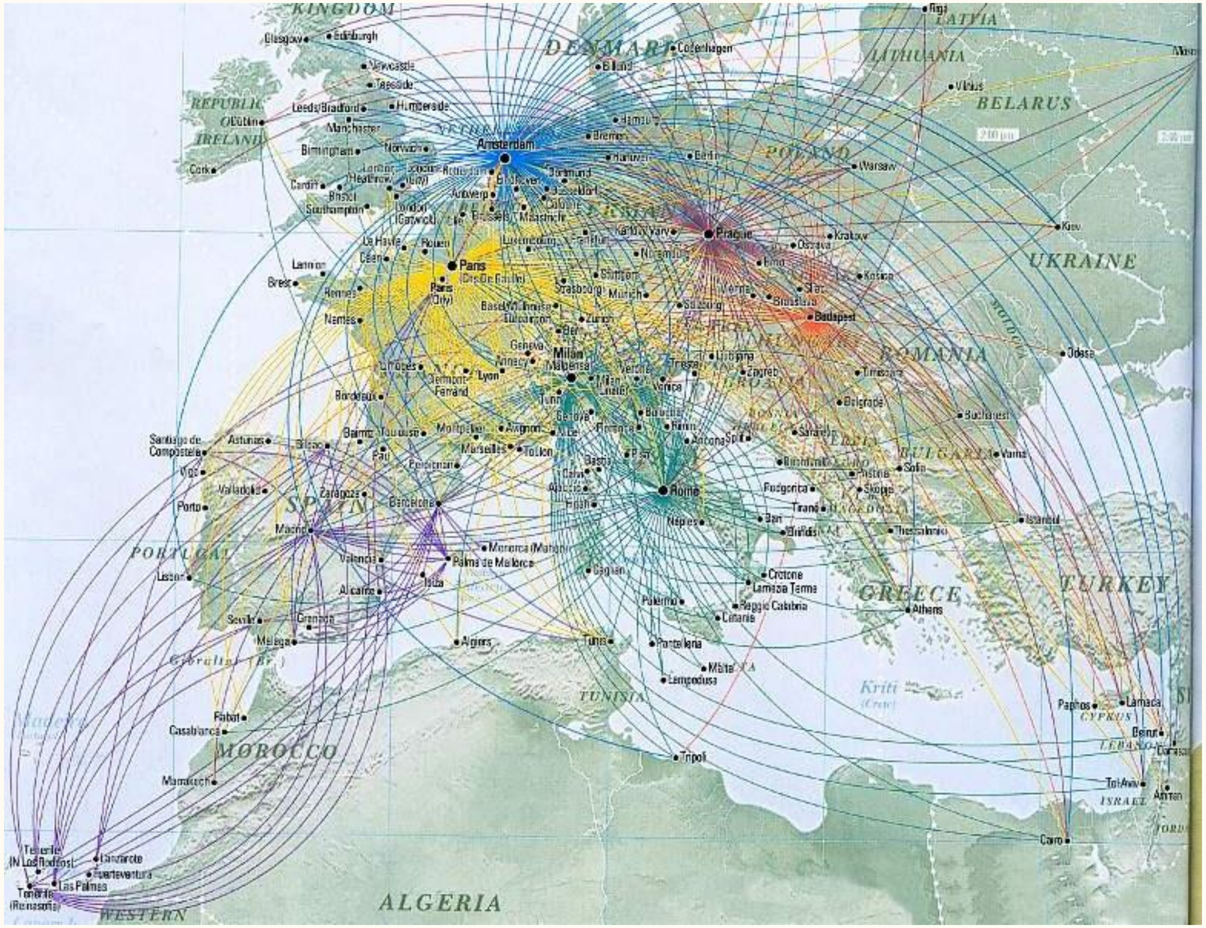
- 1.47 billion daily active users on average in Jun. 2018
- 2.23 billion monthly active users as of June 30, 2018

http://newsroom.fb.com/company-info/

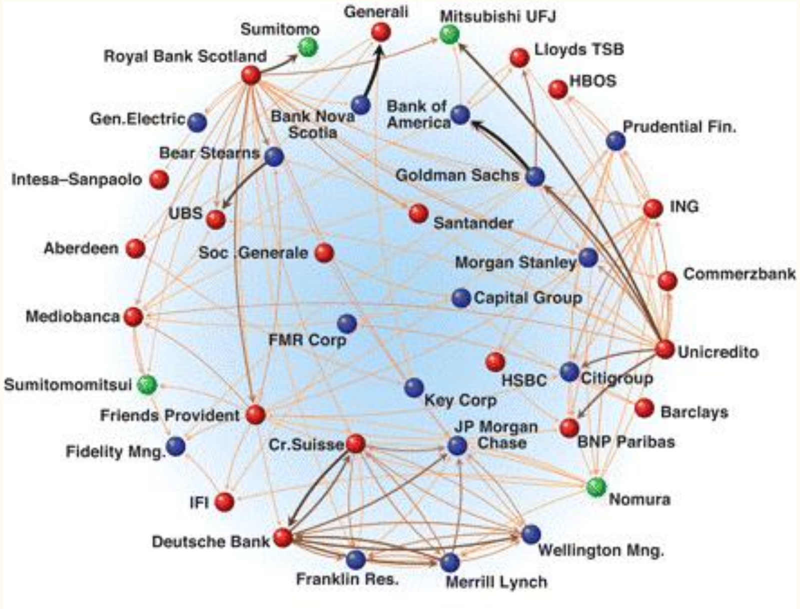
facebook

Transportation Networks: Airlines

Picture taken from a course by L Adamic



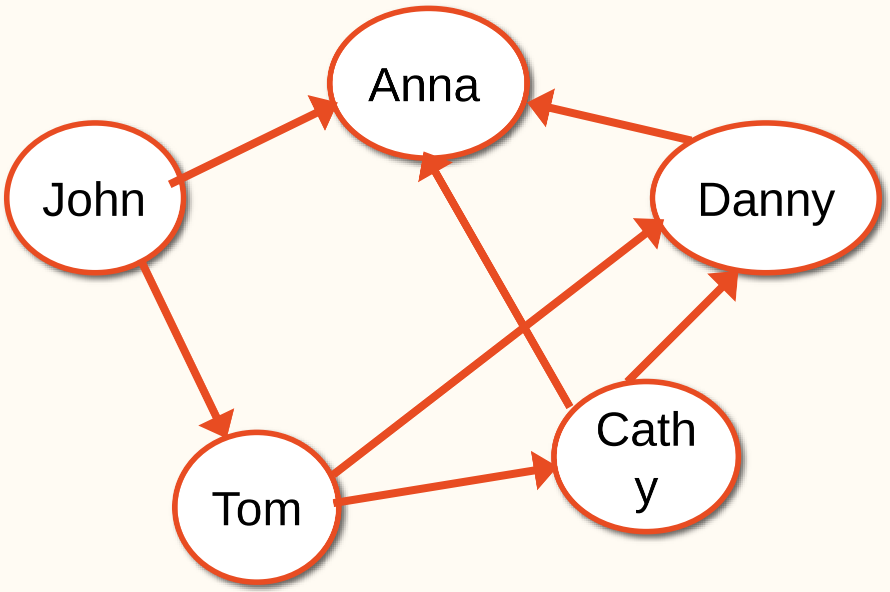
Financial Networks



What is a Graph?

G = (V, E), where

- *V* represents the set of vertices (entities)
- *E* represents the set of edges (relations)
- Both vertices and edges may contain additional information



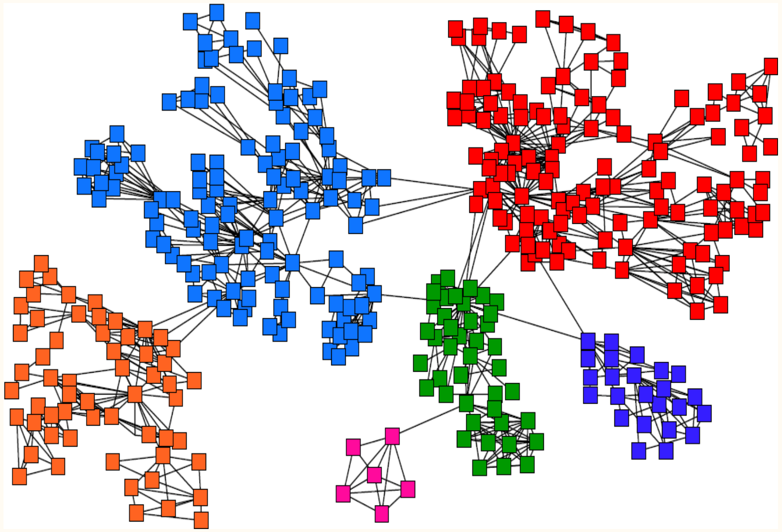
A twitter network

The Structural Analysis of Graphs

Communities

A community is a cohesive group of nodes that are connected more densely to each other than to the other nodes in other communities

- Edges within a community: high density
- Edges between communities: low density



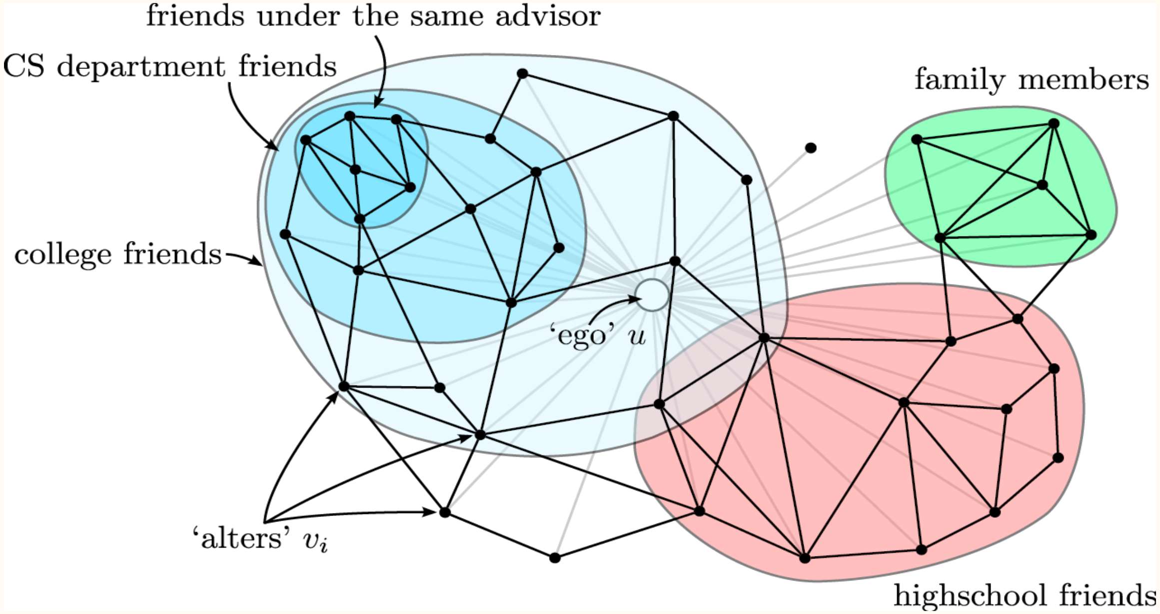
Community structures are common in real networks.

Finding Communities

People tend to work together.

Discovering social circles in ego networks (*McAuley*, *Leskovec*, *2012*)

Definition (ego network): a portion of a social network formed of a given individual, termed ego, and the other persons with whom she has a social relationship with



How to Measure Cohesiveness

A community is a subgraph in a network.

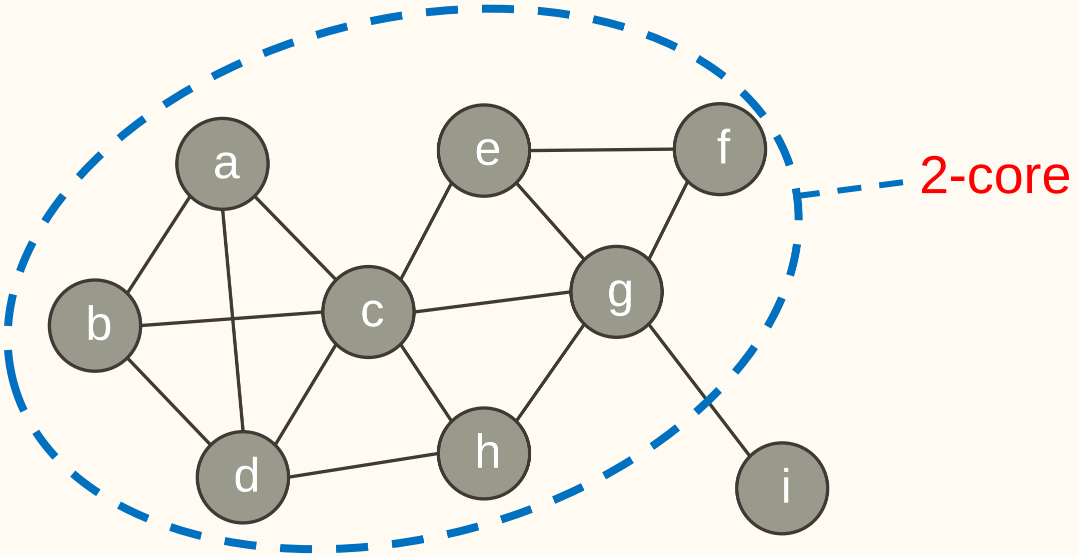
One proposed model is the K-core: Where every node in a **subgraph** connects to at least k other nodes.

K-core is one of the models to model communities

Definition (**subgraph**): A portion of a graph G obtained by either eliminating edges from G and/or eliminating some vertices and their associated edges. – *oxfordrederence.com*

K-cores

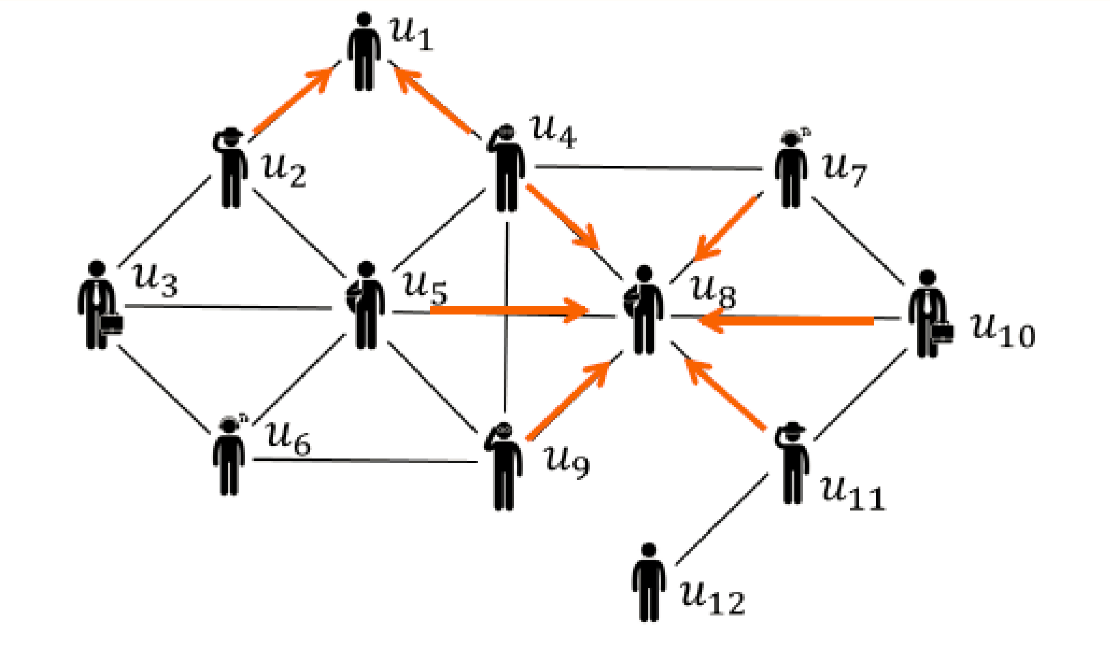
Given a graph G, the k-core of G is a **subgraph** where each vertex has at least k neighbors (i.e., k adjacent vertex, or a **degree** of k).



S. B. Seidman. Network structure and minimum degree. Social networks, 5(3):269–287, 1983.

Why Study K-core ?

The engagement of a user is influenced by the number of her engaged friends.

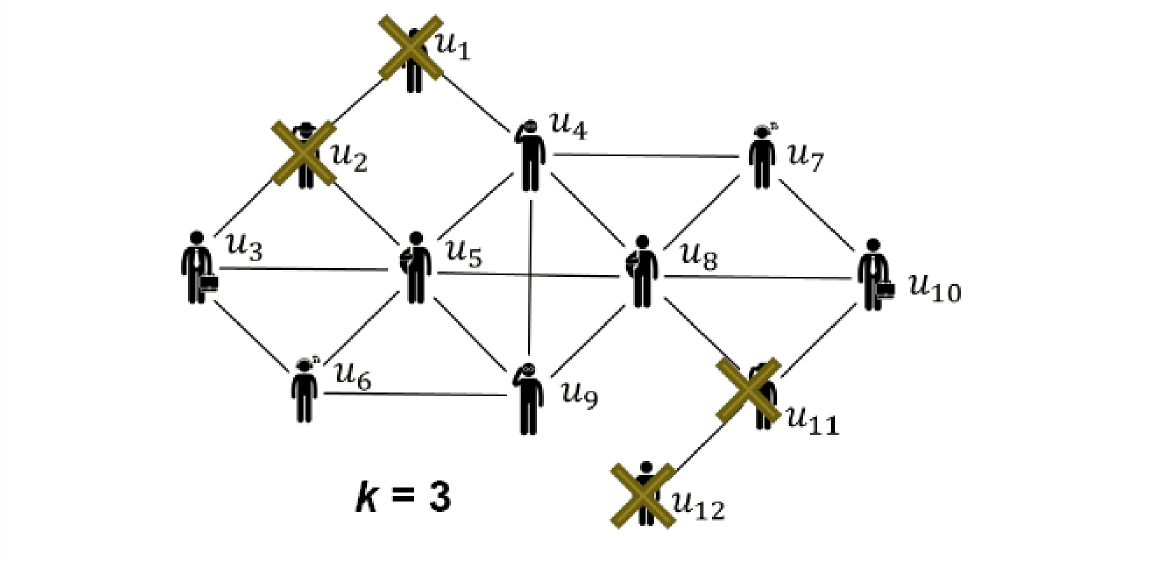


K. Bhawalkar, J. Kleinberg, K. Lewi, T. Roughgarden, and A. Sharma. Preventing unraveling in social networks: the anchored k-core problem. *SIAM Journal on Discrete Mathematics*, 29(3):1452–1475, 2015.

Why Study K-core ?

Assume a user will leave if less than k friends in the group

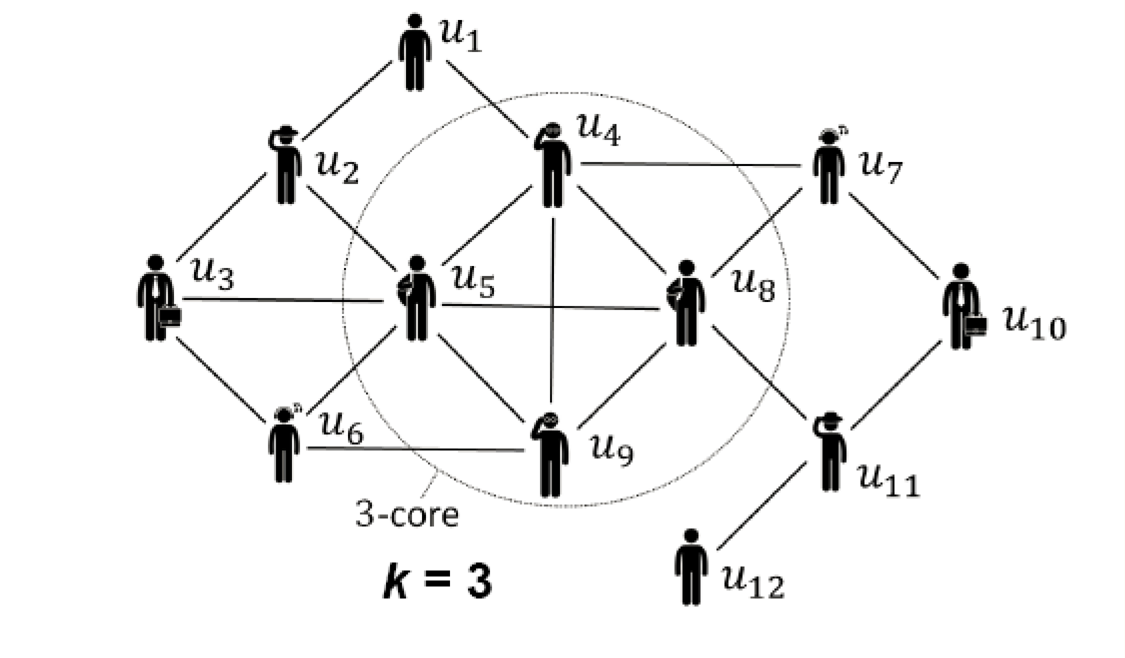
An equilibrium: a group has the minimum degree of *k,* namely *k-core*



K. Bhawalkar, J. Kleinberg, K. Lewi, T. Roughgarden, and A. Sharma. Preventing unraveling in social networks: the anchored k-core problem. *SIAM Journal on Discrete Mathematics*, 29(3):1452–1475, 2015.

Why Study K-core ?

A stable social group tends to be a k-core in the network



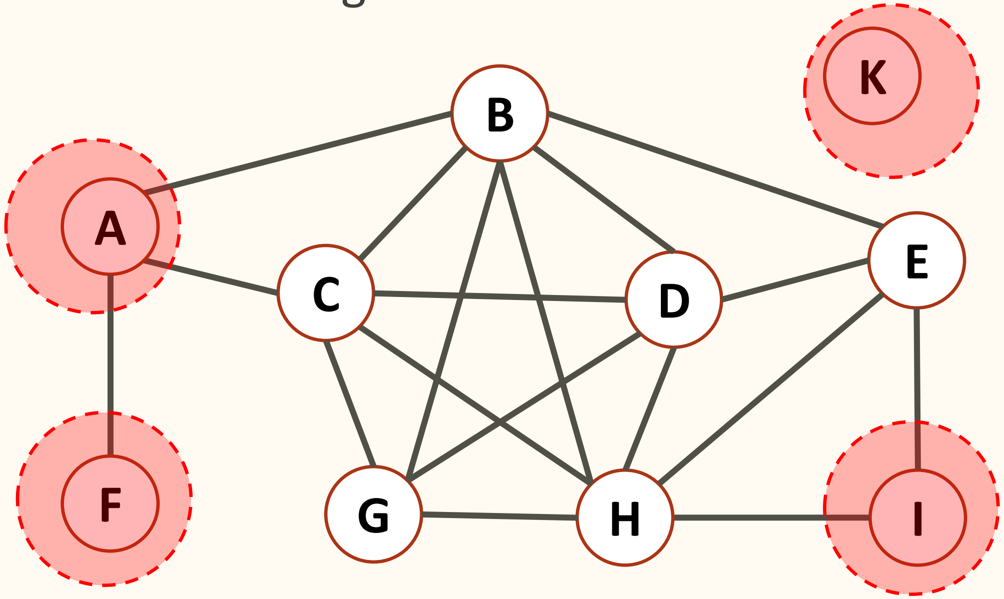
Computation of a *K*-core

Given a graph G, the k-core of G can be computed by recursively deleting every vertex and its adjacent edges if its degree is less than k.

Repeat until no vertex has a **degree** less than k.

Definition (degree): total number of edges connected to that vertex

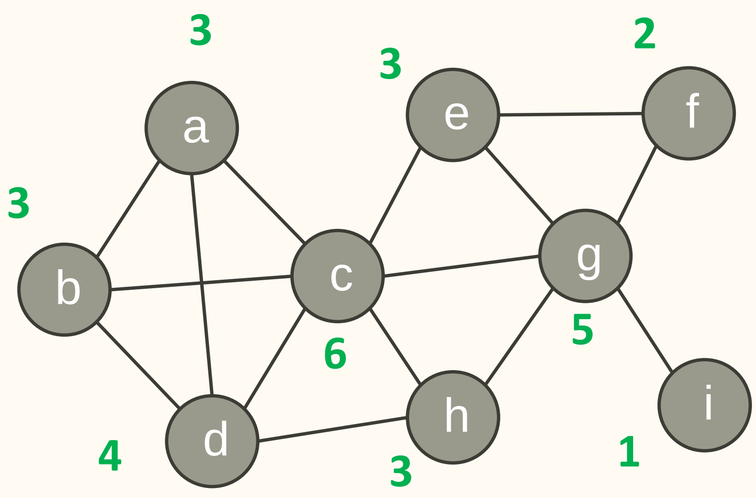
Maximal 3-core



S. B. Seidman. Network structure and minimum degree. Social networks, 5(3):269–287, 1983.

Core Decomposition Algorithm

Batagelj and Zaversnik Algorithm:



Algorithm 1 In the algorithm the core number of vertex v, core(v), is represented by the table element $core[v]$, and its degree by the table element $degree[v]$.

INPUT: graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ represented by lists of neighbors $Neighbors(v)$ for each vertex v
OUTPUT: table $core$ with core number $core[v]$ for each vertex v

1.1 compute the degrees of vertices;
1.2 order the set of vertices $\mathcal{V}$ in increasing order of their degrees;
2 for each $v \in \mathcal{V}$ in the order do begin
2.1 $core[v] := degree[v]$;
2.2 for each $u \in Neighbors(v)$ do
2.2.1 if $degree[u] > degree[v]$ then begin
2.2.1.1 $degree[u] := degree[u] - 1$;
2.2.1.2 reorder $\mathcal{V}$ accordingly
end
end;

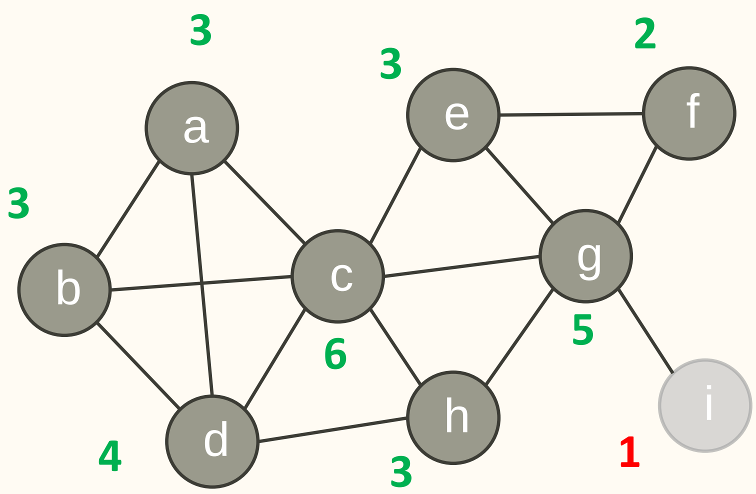
Number in green color: degree

Number in red color: core number

V. Batagelj and M. Zaversnik. An o(m) algorithm for cores decomposition of networks. CoRR, cs.DS/0310049, 2003

Core Decomposition Algorithm

Batagelj and Zaversnik Algorithm:



Algorithm 1 In the algorithm the core number of vertex v, core(v), is represented by the table element $core[v]$, and its degree by the table element $degree[v]$.

INPUT: graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ represented by lists of neighbors $Neighbors(v)$ for each vertex v
OUTPUT: table $core$ with core number $core[v]$ for each vertex v

```
1.1       compute the degrees of vertices;
1.2       order the set of vertices V in increasing order of their degrees;
2         for each v ∈ V in the order do begin
2.1           core[v] := degree[v];
2.2           for each u ∈ Neighbors(v) do
2.2.1             if degree[u] > degree[v] then begin
2.2.1.1               degree[u] := degree[u] − 1;
2.2.1.2               reorder V accordingly
                  end
          end;
```

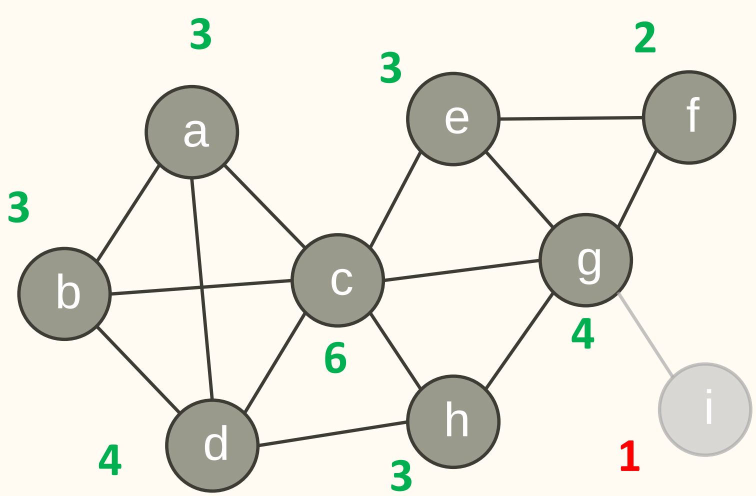
Number in green color: degree

Number in red color: core number

V. Batagelj and M. Zaversnik. An o(m) algorithm for cores decomposition of networks. CoRR, cs.DS/0310049, 2003

Core Decomposition Algorithm

Batagelj and Zaversnik Algorithm:



Algorithm 1 In the algorithm the core number of vertex v, core(v), is represented by the table element $core[v]$, and its degree by the table element $degree[v]$.

INPUT: graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ represented by lists of neighbors $Neighbors(v)$ for each vertex v
OUTPUT: table $core$ with core number $core[v]$ for each vertex v

1.1 compute the degrees of vertices;
1.2 order the set of vertices $\mathcal{V}$ in increasing order of their degrees;
2 for each $v \in \mathcal{V}$ in the order do begin
2.1 $core[v] := degree[v]$;
2.2 for each $u \in Neighbors(v)$ do
2.2.1 if $degree[u] > degree[v]$ then begin
2.2.1.1 $degree[u] := degree[u] - 1$;
2.2.1.2 reorder $\mathcal{V}$ accordingly
end
end;

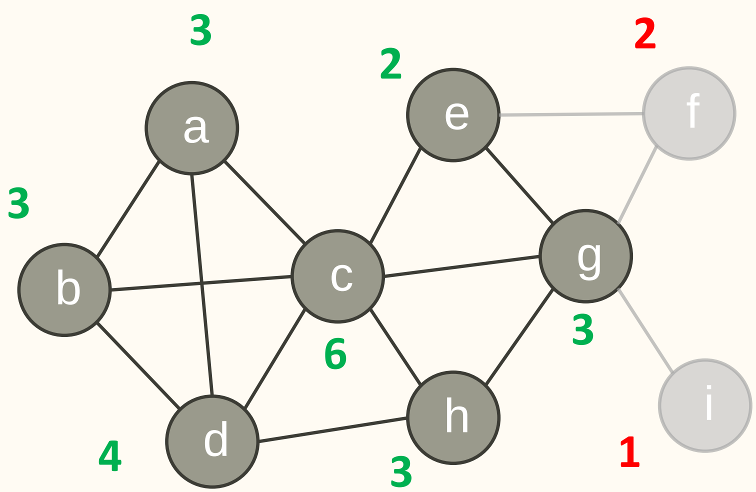
Number in green color: degree

Number in red color: core number

V. Batagelj and M. Zaversnik. An o(m) algorithm for cores decomposition of networks. CoRR, cs.DS/0310049, 2003

Core Decomposition Algorithm

Batagelj and Zaversnik Algorithm:



Algorithm 1 In the algorithm the core number of vertex v, core(v), is represented by the table element $core[v]$, and its degree by the table element $degree[v]$.

INPUT: graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ represented by lists of neighbors $Neighbors(v)$ for each vertex v
OUTPUT: table $core$ with core number $core[v]$ for each vertex v

1.1 compute the degrees of vertices;
1.2 order the set of vertices $\mathcal{V}$ in increasing order of their degrees;
2 for each $v \in \mathcal{V}$ in the order do begin
2.1 $core[v] := degree[v]$;
2.2   for each $u \in Neighbors(v)$ do
2.2.1   if $degree[u] > degree[v]$ then begin
2.2.1.1 $degree[u] := degree[u] - 1$;
2.2.1.2   reorder $\mathcal{V}$ accordingly
        end
    end;

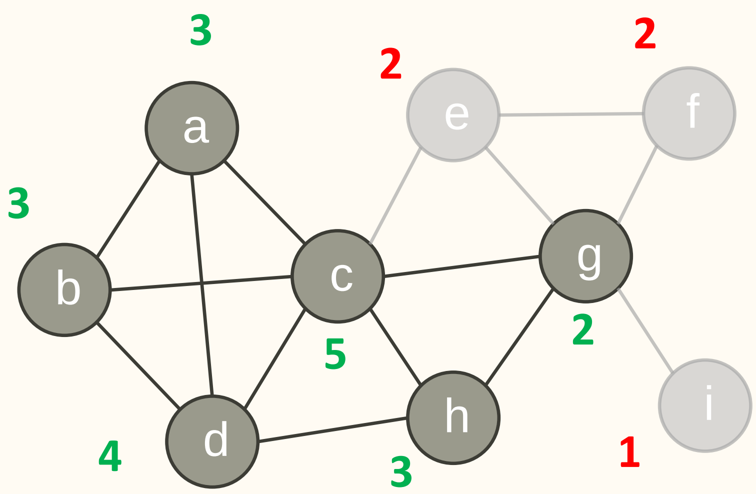
Number in green color: degree

Number in red color: core number

V. Batagelj and M. Zaversnik. An o(m) algorithm for cores decomposition of networks. CoRR, cs.DS/0310049, 2003

Core Decomposition Algorithm

Batagelj and Zaversnik Algorithm:



Algorithm 1 In the algorithm the core number of vertex v, core(v), is represented by the table element $core[v]$, and its degree by the table element $degree[v]$.

INPUT: graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ represented by lists of neighbors $Neighbors(v)$ for each vertex v
OUTPUT: table $core$ with core number $core[v]$ for each vertex v

```
1.1      compute the degrees of vertices;
1.2      order the set of vertices V in increasing order of their degrees;
2        for each v ∈ V in the order do begin
2.1          core[v] := degree[v];
2.2          for each u ∈ Neighbors(v) do
2.2.1            if degree[u] > degree[v] then begin
2.2.1.1              degree[u] := degree[u] − 1;
2.2.1.2              reorder V accordingly
                 end
         end;
```

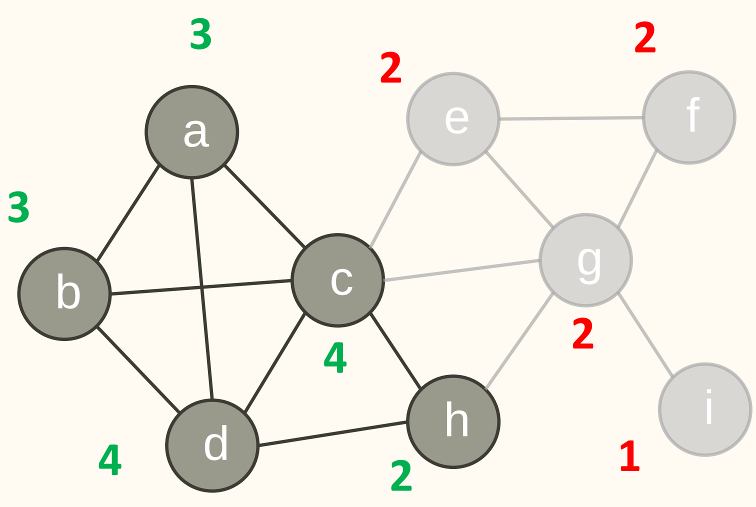
Number in green color: degree

Number in red color: core number

V. Batagelj and M. Zaversnik. An o(m) algorithm for cores decomposition of networks. CoRR, cs.DS/0310049, 2003

Core Decomposition Algorithm

Batagelj and Zaversnik Algorithm:



Algorithm 1 In the algorithm the core number of vertex v, core(v), is represented by the table element $core[v]$, and its degree by the table element $degree[v]$.

INPUT: graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ represented by lists of neighbors $Neighbors(v)$ for each vertex v
OUTPUT: table $core$ with core number $core[v]$ for each vertex v

1.1 compute the degrees of vertices;
1.2 order the set of vertices $\mathcal{V}$ in increasing order of their degrees;
2 for each $v \in \mathcal{V}$ in the order do begin
2.1 $core[v] := degree[v]$;
2.2 for each $u \in Neighbors(v)$ do
2.2.1 if $degree[u] > degree[v]$ then begin
2.2.1.1 $degree[u] := degree[u] - 1$;
2.2.1.2 reorder $\mathcal{V}$ accordingly
end
end;

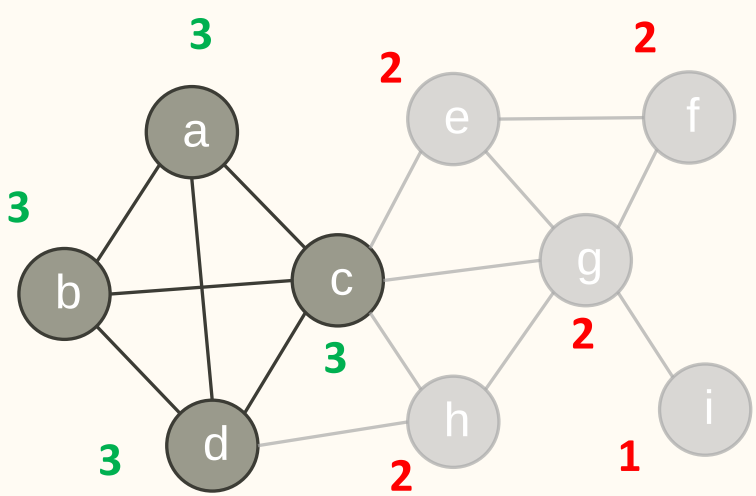
Number in green color: degree

Number in red color: core number

V. Batagelj and M. Zaversnik. An o(m) algorithm for cores decomposition of networks. CoRR, cs.DS/0310049, 2003

Core Decomposition Algorithm

Batagelj and Zaversnik Algorithm:



Algorithm 1 In the algorithm the core number of vertex v, core(v), is represented by the table element $core[v]$, and its degree by the table element $degree[v]$.

INPUT: graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ represented by lists of neighbors $Neighbors(v)$ for each vertex v
OUTPUT: table $core$ with core number $core[v]$ for each vertex v

1.1 compute the degrees of vertices;
1.2 order the set of vertices $\mathcal{V}$ in increasing order of their degrees;
2 for each $v \in \mathcal{V}$ in the order do begin
2.1 $core[v] := degree[v]$;
2.2 for each $u \in Neighbors(v)$ do
2.2.1 if $degree[u] > degree[v]$ then begin
2.2.1.1 $degree[u] := degree[u] - 1$;
2.2.1.2 reorder $\mathcal{V}$ accordingly
end
end;

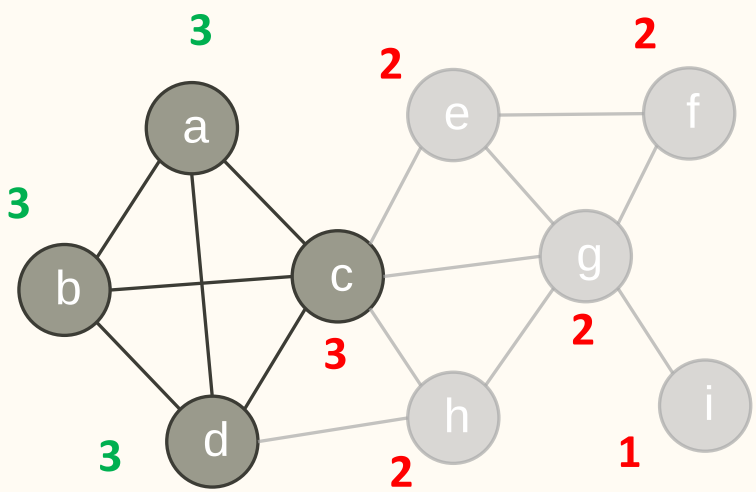
Number in green color: degree

Number in red color: core number

V. Batagelj and M. Zaversnik. An o(m) algorithm for cores decomposition of networks. CoRR, cs.DS/0310049, 2003

Core Decomposition Algorithm

Batagelj and Zaversnik Algorithm:



Algorithm 1 In the algorithm the core number of vertex v, core(v), is represented by the table element $core[v]$, and its degree by the table element $degree[v]$.

INPUT: graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ represented by lists of neighbors $Neighbors(v)$ for each vertex v
OUTPUT: table $core$ with core number $core[v]$ for each vertex v

```
1.1      compute the degrees of vertices;
1.2      order the set of vertices V in increasing order of their degrees;
2        for each v ∈ V in the order do begin
2.1          core[v] := degree[v];
2.2          for each u ∈ Neighbors(v) do
2.2.1            if degree[u] > degree[v] then begin
2.2.1.1              degree[u] := degree[u] − 1;
2.2.1.2              reorder V accordingly
                 end
         end;
```

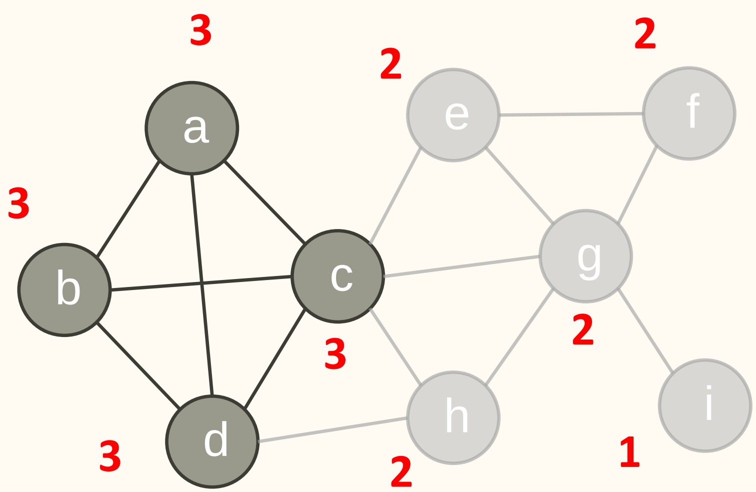
Number in green color: degree

Number in red color: core number

V. Batagelj and M. Zaversnik. An o(m) algorithm for cores decomposition of networks. CoRR, cs.DS/0310049, 2003

Core Decomposition Algorithm

Batagelj and Zaversnik Algorithm:



Algorithm 1 In the algorithm the core number of vertex v, core(v), is represented by the table element $core[v]$, and its degree by the table element $degree[v]$.

INPUT: graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ represented by lists of neighbors $Neighbors(v)$ for each vertex v
OUTPUT: table $core$ with core number $core[v]$ for each vertex v

```
1.1      compute the degrees of vertices;
1.2      order the set of vertices V in increasing order of their degrees;
2        for each v ∈ V in the order do begin
2.1          core[v] := degree[v];
2.2          for each u ∈ Neighbors(v) do
2.2.1            if degree[u] > degree[v] then begin
2.2.1.1              degree[u] := degree[u] − 1;
2.2.1.2              reorder V accordingly
                 end
         end;
```

Number in green color: degree

Number in red color: core number

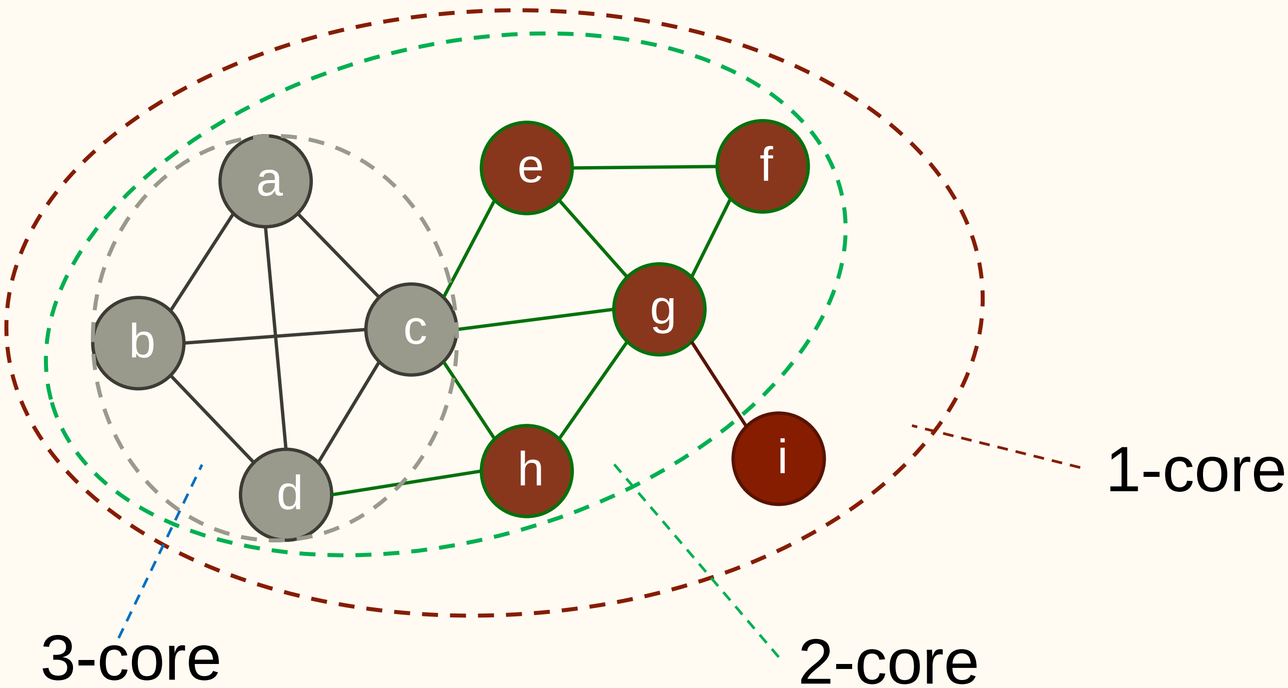
V. Batagelj and M. Zaversnik. An o(m) algorithm for cores decomposition of networks. CoRR, cs.DS/0310049, 2003

K-core Decomposition

The $(k+1)$-core is contained in the k-core, for each $k \geq 0$.

For a vertex v, its **coreness** is the maximum k such that v is in the k-core but not in the $(k+1)$-core.

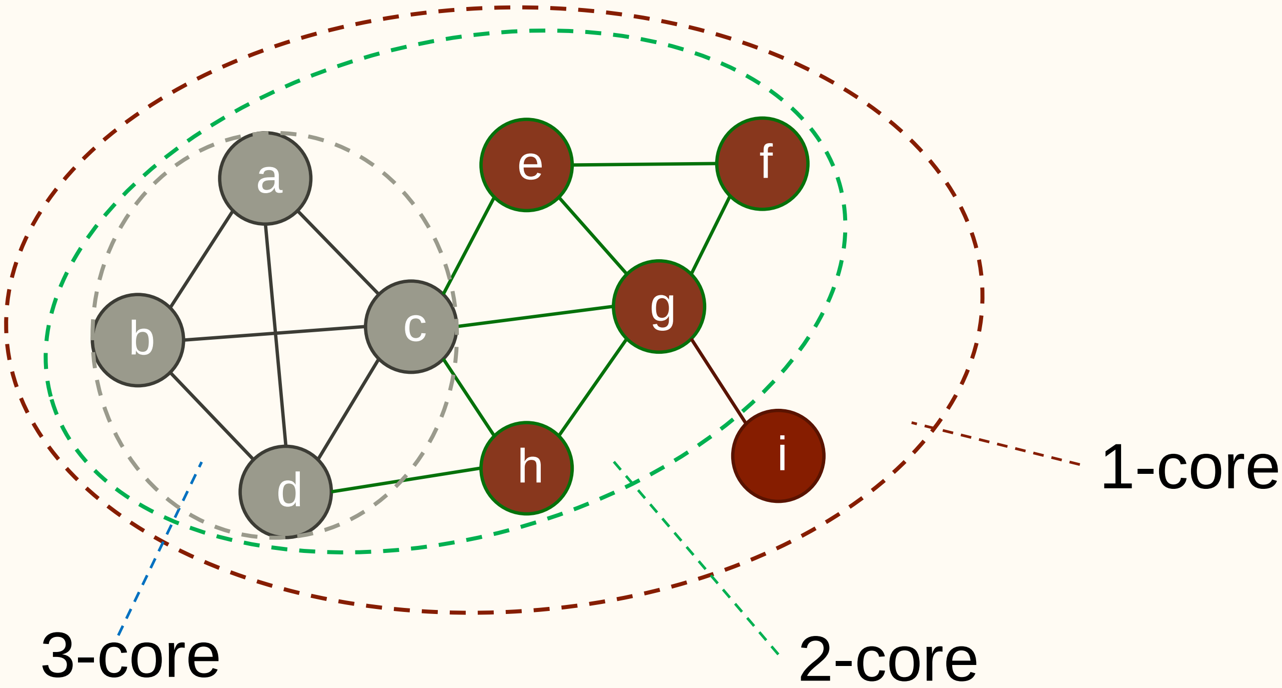
- A k-core contains vertices with **coreness** $\geq k$.



K-core Decomposition

i.e, Core number/coreness of a vertex v: the largest value of k such that there is a k-core containing v.

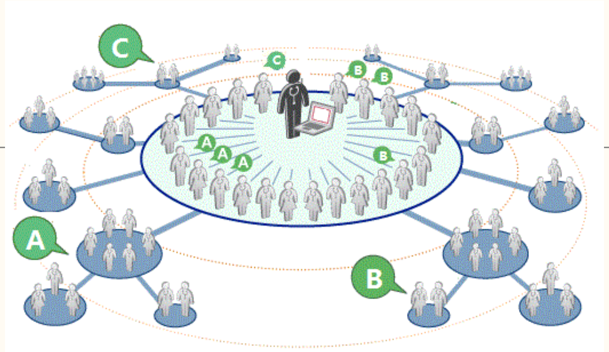
Core **decomposition**: computes the core number of each vertex in G.

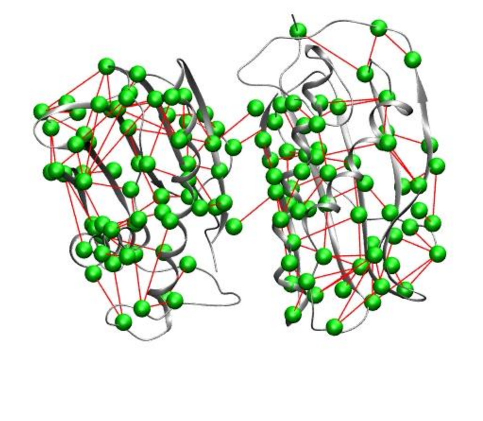


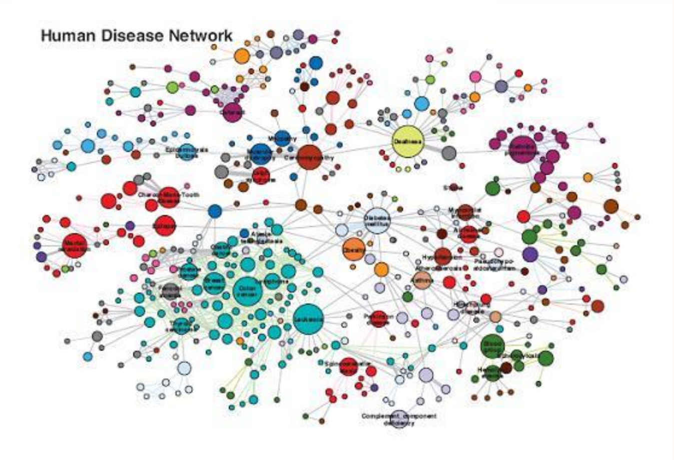
Other Applications

- Community detection
- User engagement
- Event detection
- Influence study
- Graph clustering
- Protein function prediction
- Network Visualization

…





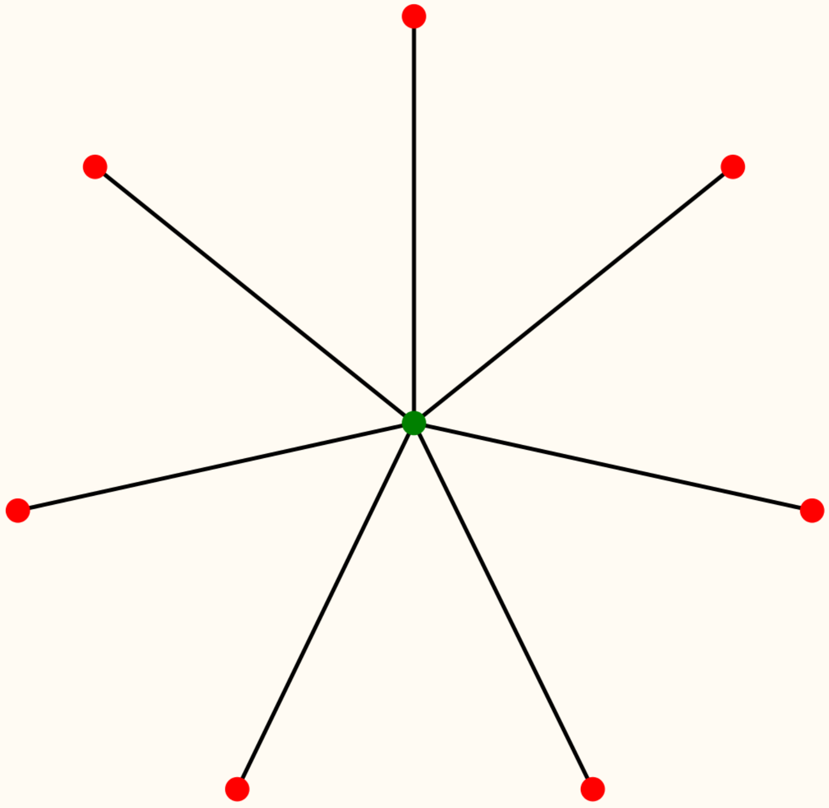


Practice:

What is the k-core of the graph on the right?

What is the core-number of the green vertex? 1

Can I draw conclusions based on its degree alone (degree of 7)? No…



Learning Outcome

K-core: definition and its computation