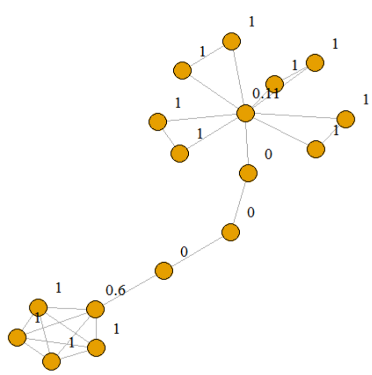
Home assignment 3 - Data Mining ITI8730. Urmas Pitsi, 192028IAPM.

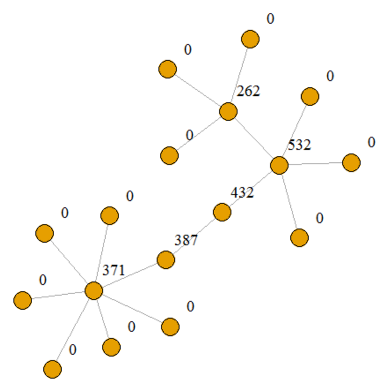
Exercise 1: Implemented 12 of 13 statistics (did not implement Hosoya).

First lets have a look on statistics that describe node in the graph.
1. Local clustering shows proportion of how tightly is neigborhood connected. For complete graph we see 1-s, for star-graph we see small value in the center.
2. Betweenness shows the number of shortest paths going through each node.

1. Local Clustering Coefficient (undirected)

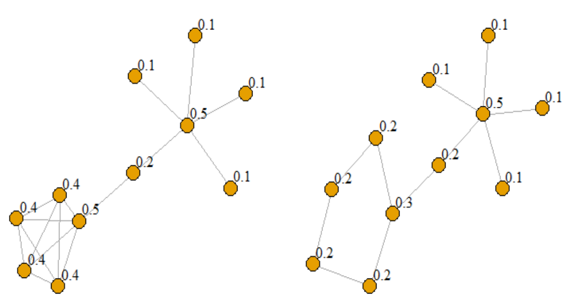


2. Betweenness Centrality (undirected)

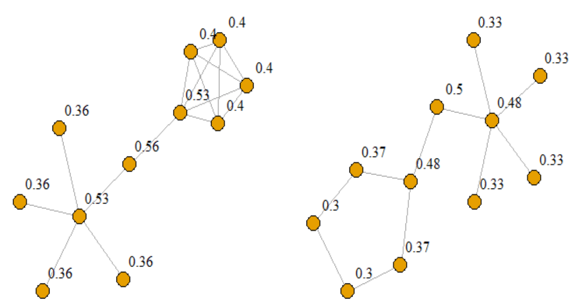


3. Degree Centrality is degree of the node divided by the maximum possible degree of the nodes.
4. Closeness Centrality is the inverse of the average distance of other nodes to i. It shows whether node is directly reachable from everywhere (1.0) or far away from others (close to 0).

3. Degree Centrality (undirected)

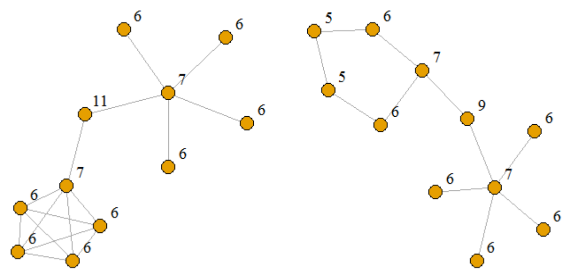


4. Closeness Centrality (undirected)

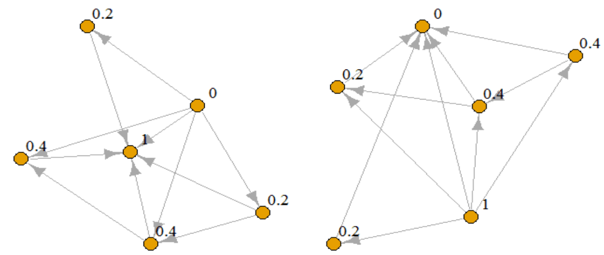


5. Morgan Index shows the number nodes in the k-neighbourhood.
6. Prestige and Gregariousness are relevant for directed graphs. Prestige of a node is 1 where all arrows are pointing in (eg everybody is following this person), and 0 where all arrows are pointing out (eg this person follows everybody). Gregariousness is the opposite: 1 for the person who follows everybody and 0 for person who follows nobody.

5. Morgan Index @ k=2 (undirected)



6. Prestige (left) and Gregariousness (right)



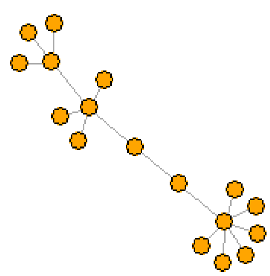
Some statistics that describe graph as a whole.

Wiener Index is the sum of the pairwise shortest path distances between all pairs of nodes.

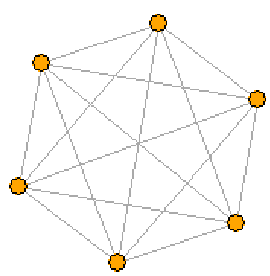
Estrada Index is the sum of e to the power of eigenvalues of the adjacency matrix.

Circuit Rank is the minimum number of edges that need to be removed from a graph in order to remove all cycles. CR = Nr of edges - nr of nodes + nr of connected components. For trees there are no cycles.

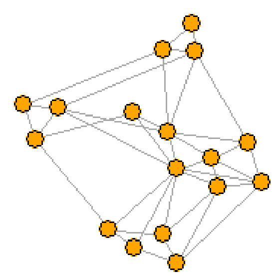
7. Wiener index, Estrada index and Circuit Rank for 3 different graphs: tree, complete graph and well connected graph.



- wiener index 458
- estrada index 42.8
- circuit rank 0



- wiener index 15
- estrada index 150.3
- circuit rank 10

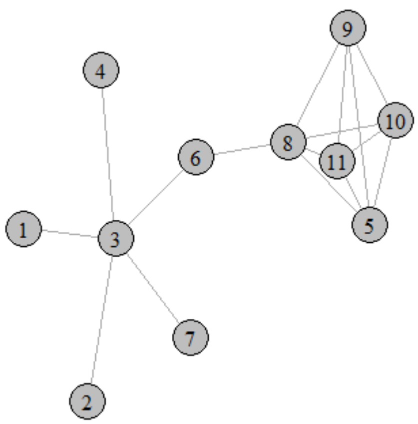


- wiener index 280
- estrada index 210.7
- circuit rank 22

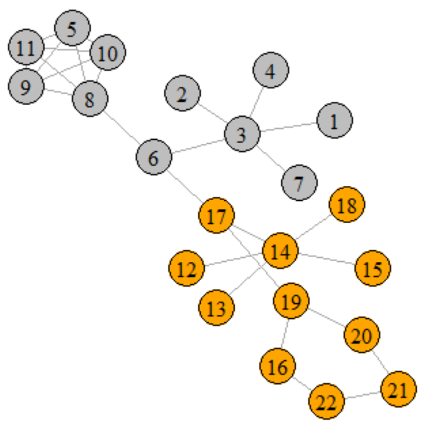
Exercise 2. Maximal Common Subgraph.

To find maximal common subgraph I implemented 2 alternative solutions. One with 'igraph'-s inbuilt function 'subgraph_isomorphisms' and the other one is my own creation: randomized connected component matcher (RCCM). It works as follows: 1. Find candidate nodes from graph1 (filtered with degrees that are possible); 2. Create random sample of connected components from graph1 with size k, starting from each canditate; 3. For each component from g1 we sample random connected components from graph2; 4. We check each candidate component from graph 2 whether it is a possible match. Candidate matching is performed by checking isomorphism between vertex induced subgraphs from graph1 and graph2.

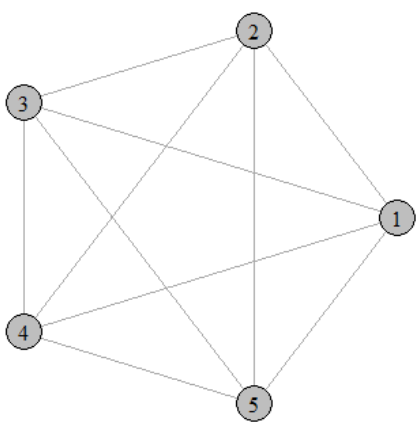
Query graph.



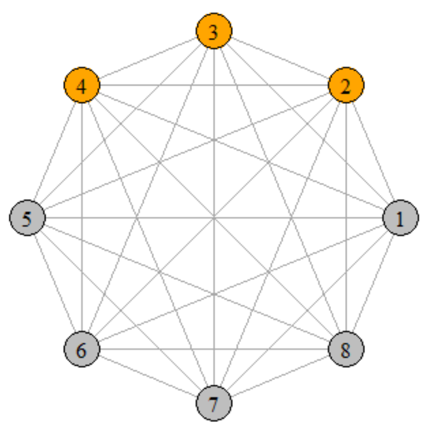
Data graph: max common subgraph in grey color.



Query: full graph with 5 vertices.

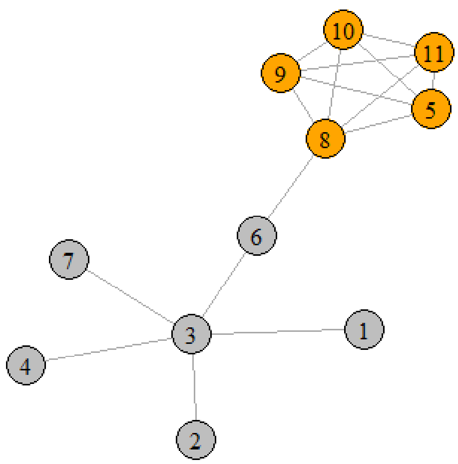


Data graph: full graph with 8 vertices.
Max common subgraph in grey color.



Randomized Connected Component Matcher identified 6-vertex common component.

Query graph.



Data graph: max common subgraph in grey color.