# Network representations, basic network algorithms

You are given three networks in Pajek format that was presented in lectures.

- A simple toy network for testing (tiny)
- Zachary karate club network (small)
- A part of Google web graph (large)

## I. Adjacency list representation

1. Assume that all networks are undirected. Implement your own adjacency list representation of the networks as an array of lists.

2. Assume now that all networks are directed and extend your network representation accordingly.

3. Does your network representation allow for multiple links between the nodes, loops on nodes and/or isolated nodes?

## II. Basic network statistics

1. Compute the basic statistic of all three networks. Namely, the number of nodes $n$, the number of links $m$, the average node degree $\langle k \rangle$ and the undirected density $\rho$. Are the results expected?

2. Find the number of isolated and the number of pendant nodes in the networks, and the maximum node degree $k_{\max}$. How do the values of $k_{\max}$ compare to $\langle k \rangle$?

3. What is the time complexity of the computations above?

## III. Network connected components

1. Study the following algorithm for computing (weakly) connected components by simple link traversal. Does the algorithm implement breadth-first or depth-first search? What is the time complexity of the algorithm?

2. Try to implement the algorithm, and compute the number of (weakly) connected components and the size of the largest (weakly) connected component of all three networks. Are the results expected?

```
input  graph G, nodes N
output  network components {C}
1:  {C} ← empty list
2:  while not N empty do
3:      {C}.add(component(G, N, N.next()))
4:  return {C}
```

```
input  graph G, nodes N, node i
output  weak component C
1:  C ← empty list
2:  S ← empty stack
3:  N.remove(S.push(i))
4:  while not S empty do
5:      C.add(i ← S.pop())
6:      for neighbors j ∈ Γ_i do
7:          if N.remove(j) then
8:              S.push(j)
9:  return C
```