

```
# Tony Hyun Kim
# CS 224W, PS0, Problem 1
# 2013 09 27

import snap

source = "wiki-Vote.txt"
#source = "example.txt"
G = snap.LoadEdgeList(snap.PNGraph, source, 0, 1) # SrcColId=0, DstColId=1

# a) Number of nodes in the network
na = sum(1 for _ in G.Nodes())
print "a) Number of nodes in the network: {}".format(na)

# b) Number of nodes with a self edge
nb = snap.CntSelfEdges(G)
print "b) Number of nodes with a self-edge: {}".format(nb)

# c) Number of directed edges in the network a->b, where a, b are distinct
nc = snap.CntUniqDirEdges(G)
print "c) Number of directed edges in the network: {}".format(nc)

# d) Number of undirected edges in the network
nd = snap.CntUniqUndirEdges(G)
print "d) Number of undirected edges in the network: {}".format(nd)

# e) Number of reciprocated edges in the network
ne = snap.CntUniqBiDirEdges(G)
print "e) Number of reciprocated edges in the network: {}".format(ne)

# f) Number of nodes of zero out-degree
nf = snap.CntOutDegNodes(G, 0)
print "f) Number of nodes of zero out-degree: {}".format(nf)

# g) Number of nodes of zero in-degree
ng = snap.CntInDegNodes(G, 0)
print "g) Number of nodes of zero in-degree: {}".format(ng)

# h) Number of nodes with more than 10 outgoing edges
nh = sum(1 if n.GetOutDeg()>10 else 0 for n in G.Nodes())
print "h) Number of nodes with more than 10 outgoing edges: {}".format(nh)

# i) Number of nodes with less than 10 incoming edges
ni = sum(1 if n.GetInDeg()<10 else 0 for n in G.Nodes())
print "i) Number of nodes with less than 10 incoming edges: {}".format(ni)
```