Librarys Import

```
#Import librarys
import networkx as nx
import matplotlib.pyplot as plt
import random
from random import sample
```

Importing data from a .tsv file

```
#Open and extract the network edge list from the tsv file.
fh=open("/content/net1000-005 - net1000-005.tsv", 'rb')
G=nx.read_edgelist(fh)
fh.close()

#Com ou sem reposição, 0: sem, 1: com
replacement = 1
```

1. Use the attached file net1000-005.tsv for this question. It contains a list of links for a network having N = 1000. The nodes are named with numbers from 1 to 1000.

(a) Select at random 100 nodes $v_0, v_1, \ldots, v_{99}$ from this network. Hand in a plain text file (no formatting) with 100 lines of the form

> v_i j

showing the name j of each selected node v_i.

```
# Create a file named 'a' what store the plain text required, with 100 random nc
f = open("a.txt", "w")
#Select random nodes from the network G
if replacement == 0:
  random_nodes = sample(list(G.nodes()), 100)
elif replacement == 1:
  random nodes = random.choices(list(G.nodes()), k=100)
```

Hand in a text file with your paths, one per line, with nodes separated by whitespace.

```
# Create a file 'b' what will store the paths from the selected random nodes v2i
#Create the 'b' file
f = open("b.txt", "w")
# Open and get the data from the 'a' file
aFile = open('a.txt','r').read().split('\n')
# For 0 to 49, verify the shortest path from v2i to
for i in range(49):
  pathI = (nx.shortest_path(G,source=aFile[2*i].split()[1],target=aFile[(2*i)+1]
  # Store the path in 'b'
  for j in pathI:
    f.write(f'{j} ')
  f.write('\n')
f.close()
```

(c) Plot a distribution of the distances you found in item 1b. Hand in this plot.

```
# Open and get the data from 'b' file with the paths from the selected random nc
bFile = open('b.txt','r').read().split('\n')
# Store the distances from each node i a list 'listOfDistances'
listOfDistances = []
for i in bFile:
  listOfDistances.append(len(i.split())-1)
f.close()
```

```
# Plot the histogram of the distances and a list of distances
# plt.hist(listOfDistances[:-1], bins=((len(Counter(listOfDistances).values()))-
plt.hist(listOfDistances[:-1])
print(listOfDistances[:-1])
```

```
[1, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
```