Import libraries:

In [1]:
```
import numpy as np
import operator
from collections import Counter
import matplotlib.pyplot as plt
```

Directed graph data:

In [2]:
```
f = open("/content/drive/My Drive/wiki-Vote.txt", "r",encoding='utf-8',errors="ignore") #data loadede is directed graph
x=f.read()   #read the data
```

In [3]:
```
y=x.split("\n")  #saparated the edges which are having \n inbteween them
```

In [4]:
```
table=[]   #list of table which carrying all the edges
for i in y:
  table.append(i.split("\t"))  #saprate two nodes by \t and carring edges between them
```

In [5]:
```
n=[]  #store all the nodes of the graph here
for i in table:
  n.append(int(i[0]))
  n.append(int(i[1]))
```

In [6]:
```
print("total number of nodes in graph is n: ", len(set(n)))
```

```
total number of nodes in graph is n:  7115
```

In [7]:
```
indegree={}   #dictionary for indegree of nodes
outdegree={}  # dictionary for outdegree for all the nodes
for i in table:  #intialise them as an empty list
  indegree[int(i[0])]=[]
  indegree[int(i[1])]=[]
  outdegree[int(i[0])]=[]
  outdegree[int(i[1])]=[]
```

In [8]:
```
for i in table:  # append all the indegree and outdegrees
  indegree[int(i[0])].append(int(i[1]))
  outdegree[int(i[1])].append(int(i[0]))
```

In [9]:
```
indegree_count={}   #count for total indegree of key k in indegree dictionary
outdegree_count={}   #count for total outdegree of key k in outdegree dictionary
for key,value in indegree.items():
  indegree_count[key]=len(value)  #add total list of node as a indegree
for key,value in outdegree.items():
  outdegree_count[key]=len(value)  #add total list of node as a outdegree
```

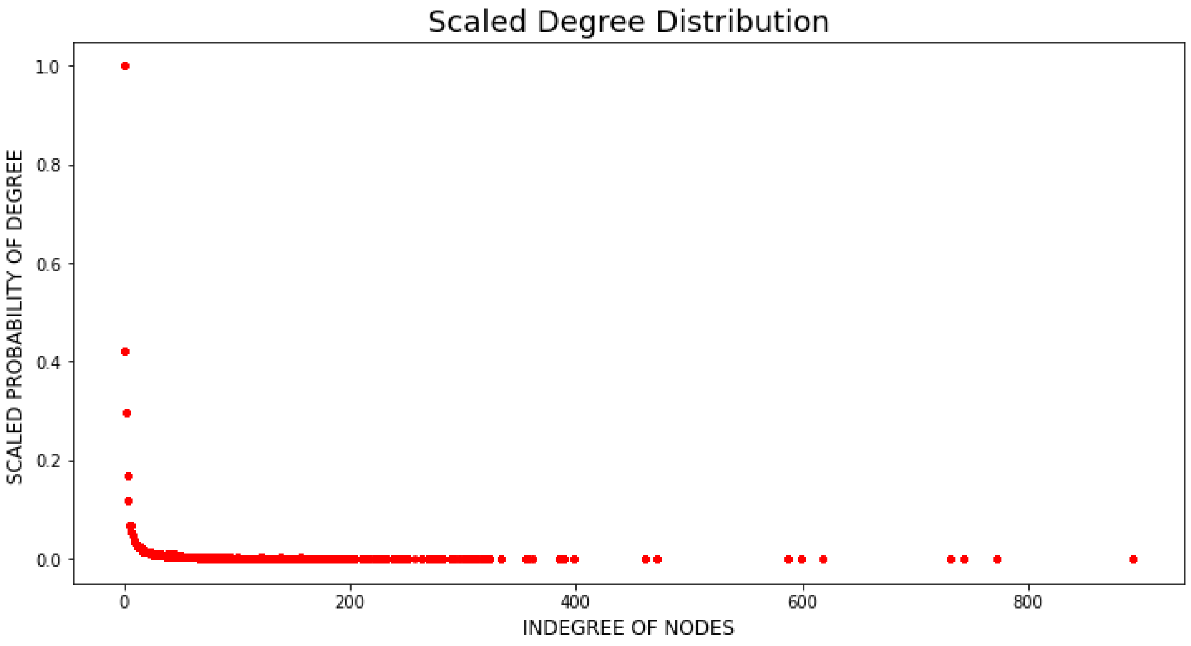
indegree distribution:

In [10]:
```
indegree_frequncy={}   #indegree frequncy of all the degrees peresent here
for key,value in indegree_count.items():  #count the frquency of all the unique degree present in the graph
  if value not in indegree_frequncy.keys():
    indegree_frequncy[value]=0
    indegree_frequncy[value]+=1
  else:
    indegree_frequncy[value]+=1     #incerment the count by 1
```

In [11]:
```
kmax=max(list(indegree_frequncy.values()))#kmax is the max degree in the graph
```

In [12]:
```
import matplotlib.pyplot as plt
import numpy as np
fg, ax = plt.subplots(figsize =(12, 6))
x_axis=[]
y_axis=[]
for key,value in  indegree_frequncy.items():
  x_axis.append(key)
  y_axis.append((value/kmax))
  ax.scatter(x_axis,y_axis,s=np.pi*3.2,c=("red"), alpha=0.5)

plt.xlabel("INDEGREE OF NODES", fontsize=12)
plt.ylabel("SCALED PROBABILITY OF DEGREE ", fontsize=12)
plt.title("Scaled Degree Distribution",fontsize=18)
plt.show()
```

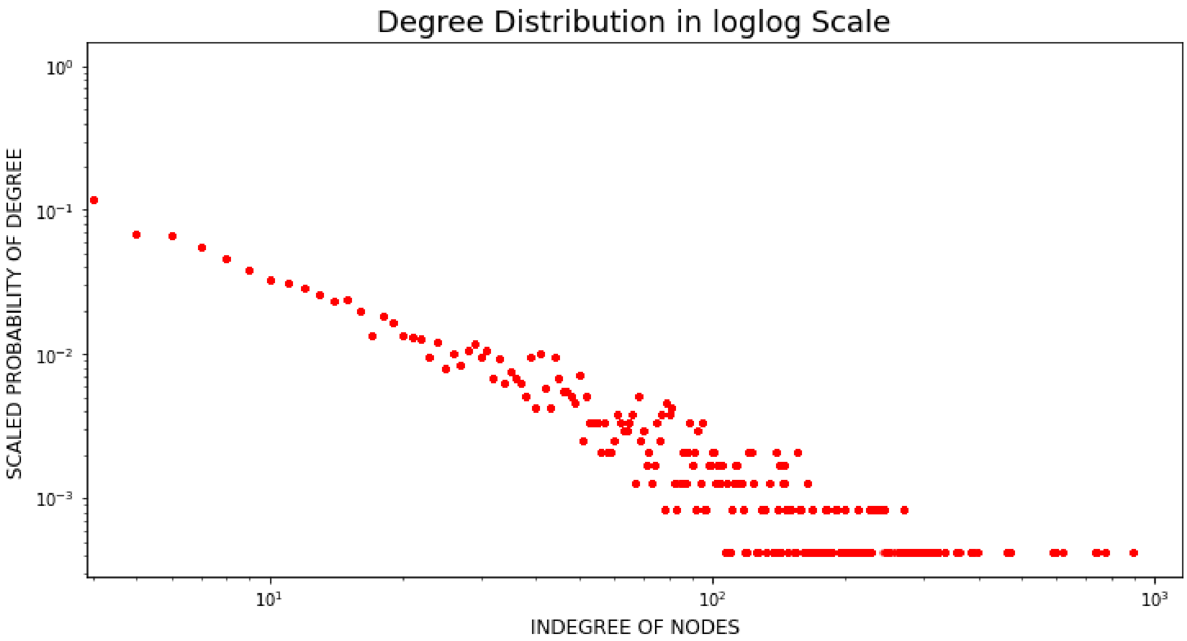


In [13]:

```
#Loglog Scaled Degree Distribution

import matplotlib.pyplot as plt
import numpy as np
fg, ax = plt.subplots(figsize =(12, 6))
x_axis=[]
y_axis=[]
for key,value in  indegree_frequncy.items():
  x_axis.append(key)
  y_axis.append(value/kmax)
  ax.scatter(x_axis,y_axis,s=np.pi*3.2,c=("red"), alpha=0.5)
  plt.yscale('log')
  plt.xscale('log')

plt.xlabel("INDEGREE OF NODES", fontsize=12)
plt.ylabel("SCALED PROBABILITY OF DEGREE ", fontsize=12)
plt.title("Degree Distribution in loglog Scale",fontsize=18)
plt.show()
```



outdegree distribution:

In [14]:

```
outdegree_frequncy={}   #outdegree frequncy of all the degrees peresent here
for key,value in outdegree_count.items():     #count the frquency of all the u
nique degree present in the graph
  if value not in outdegree_frequncy.keys():
    outdegree_frequncy[value]=0
    outdegree_frequncy[value]+=1
  else:
    outdegree_frequncy[value]+=1
```

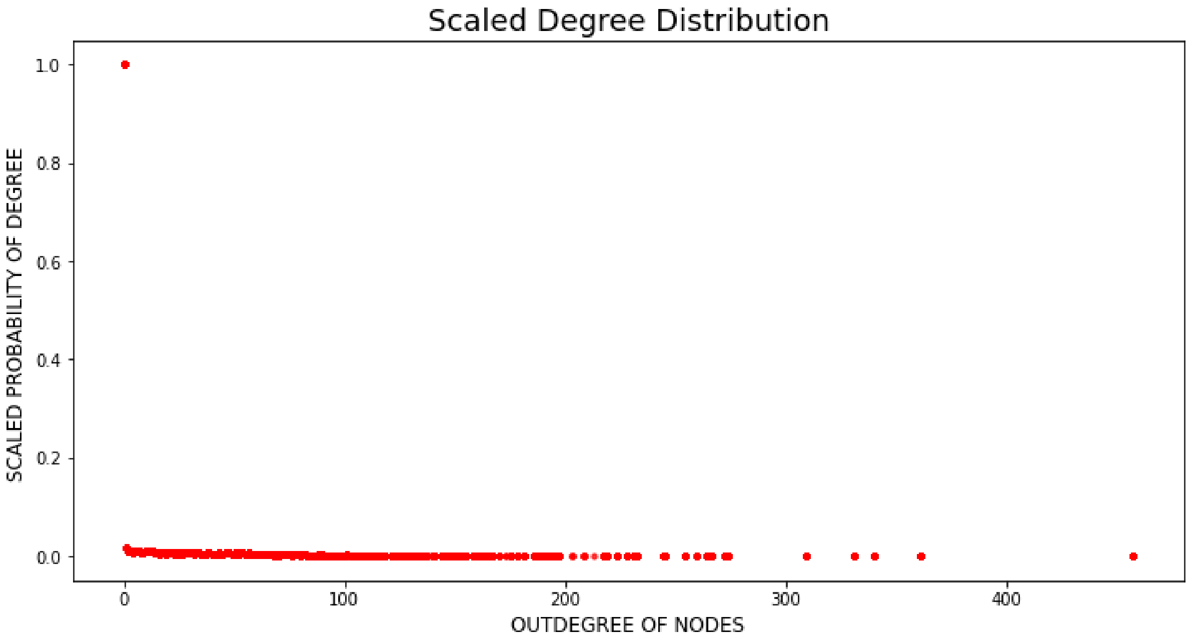
In [15]:

```
kmax=max(list(outdegree_frequncy.values()))#kmax is the max degree in the grap
h
```

In [16]:

```
import matplotlib.pyplot as plt
import numpy as np
fg, ax = plt.subplots(figsize =(12, 6))
x_axis=[]
y_axis=[]
for key,value in  outdegree_frequncy.items():
  x_axis.append(key)
  y_axis.append(value/kmax)
  ax.scatter(x_axis,y_axis,s=np.pi*3.2,c=("red"), alpha=0.5)

plt.xlabel("OUTDEGREE OF NODES", fontsize=12)
plt.ylabel("SCALED PROBABILITY OF DEGREE ", fontsize=12)
plt.title("Scaled Degree Distribution",fontsize=18)
plt.show()
```

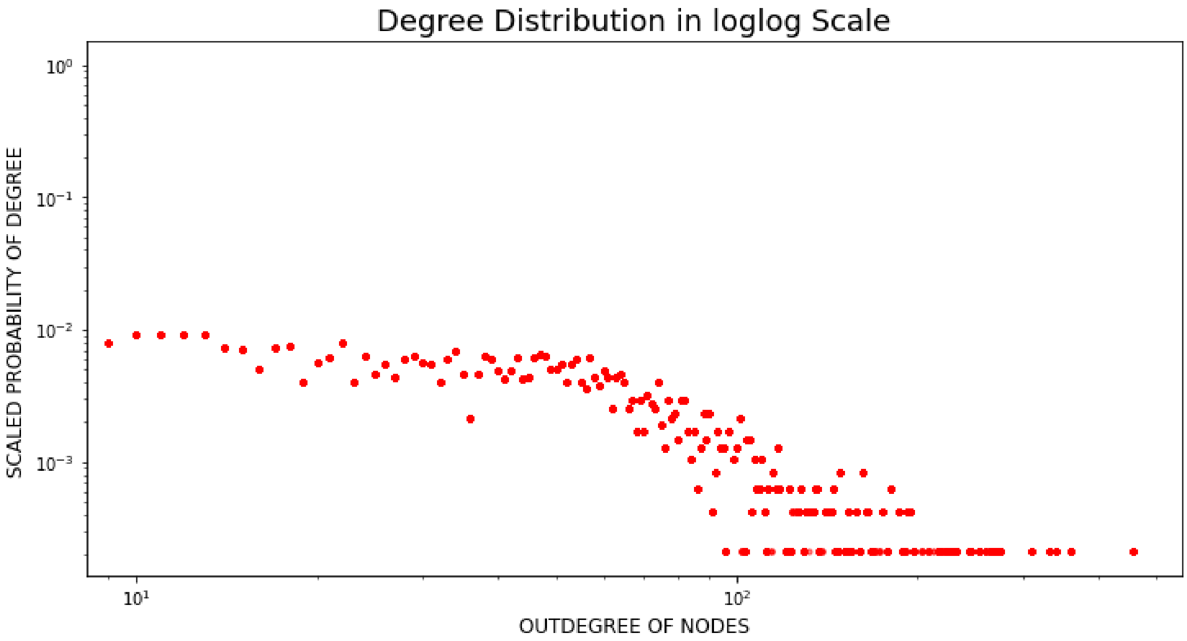


In [17]:

```
#Loglog Scaled Degree Distribution

import matplotlib.pyplot as plt
import numpy as np
fg, ax = plt.subplots(figsize =(12, 6))
x_axis=[]
y_axis=[]
for key,value in  outdegree_frequncy.items():
  x_axis.append(key)
  y_axis.append((value/kmax))
  ax.scatter(x_axis,y_axis,s=np.pi*3.2,c=("red"), alpha=0.5)
  plt.yscale('log')
  plt.xscale('log')

plt.xlabel("OUTDEGREE OF NODES", fontsize=12)
plt.ylabel("SCALED PROBABILITY OF DEGREE ", fontsize=12)
plt.title("Degree Distribution in loglog Scale",fontsize=18)
plt.show()
```



In [17]: