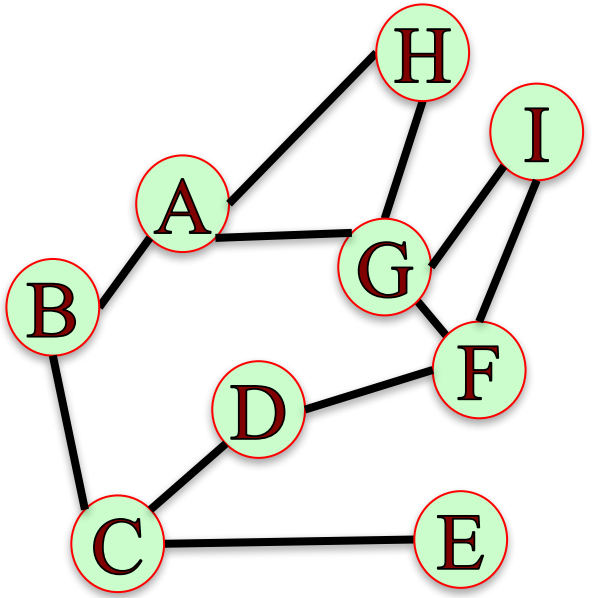
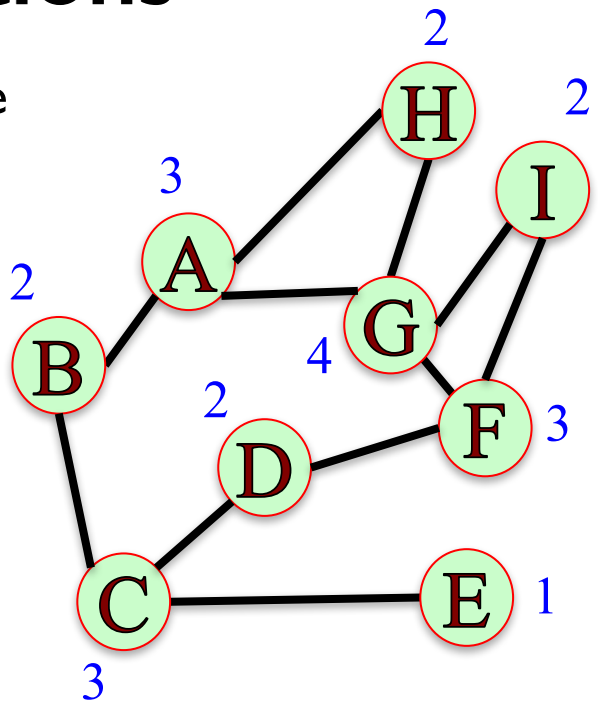
Degree Distributions

The **degree** of a node in an undirected graph is the number of neighbors it has.



Degree Distributions

The **degree** of a node in an undirected graph is the number of neighbors it has.



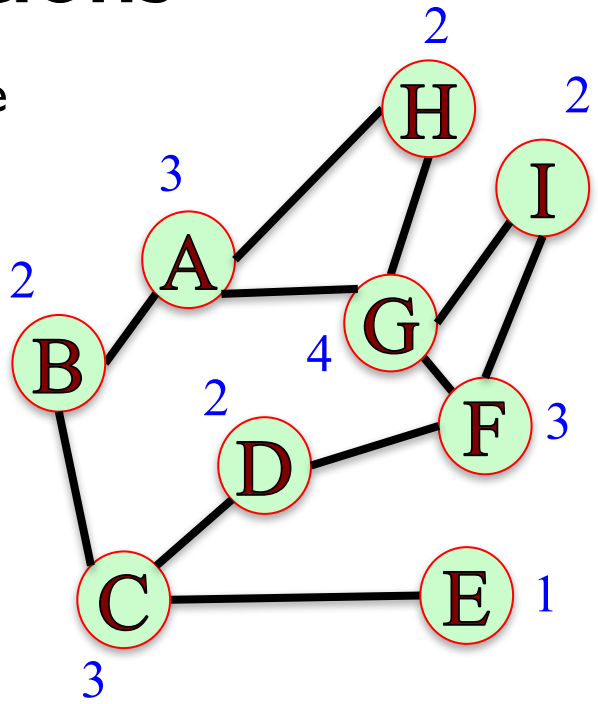
Degree Distributions

The **degree** of a node in an undirected graph is the number of neighbors it has.

The **degree distribution** of a graph is the probability distribution of the degrees over the entire network.

The degree distribution, $P(k)$, of this network has the following values:

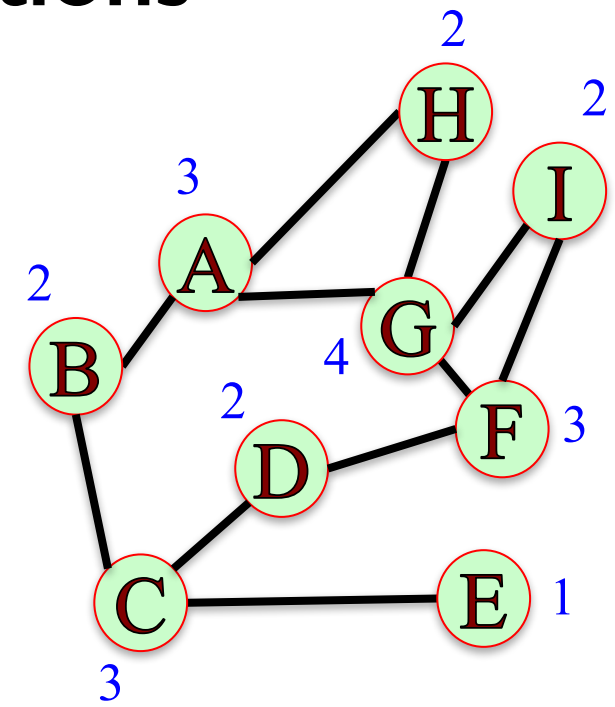
$$P(1) = \frac{1}{9}, P(2) = \frac{4}{9}, P(3) = \frac{1}{3}, P(4) = \frac{1}{9}$$



Degree Distributions

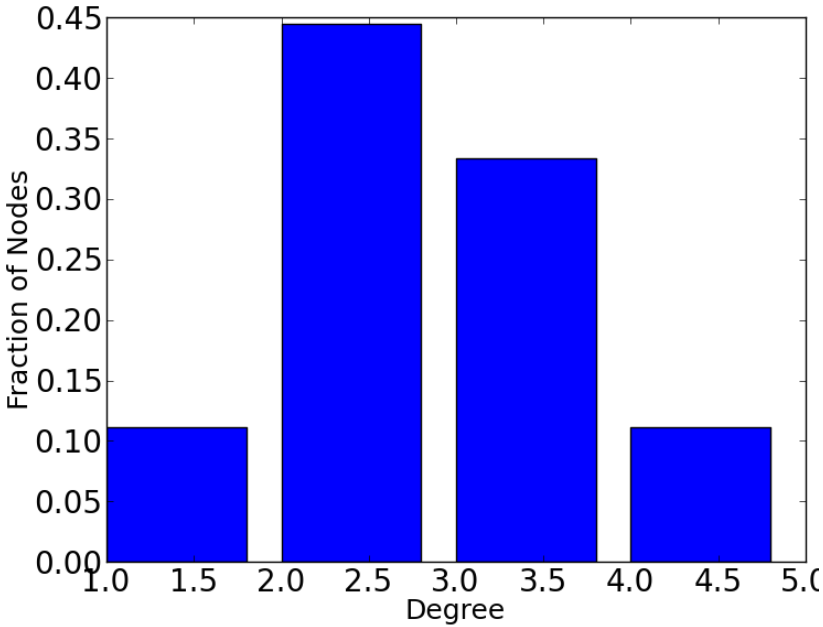
Plot of the degree distribution of this network:

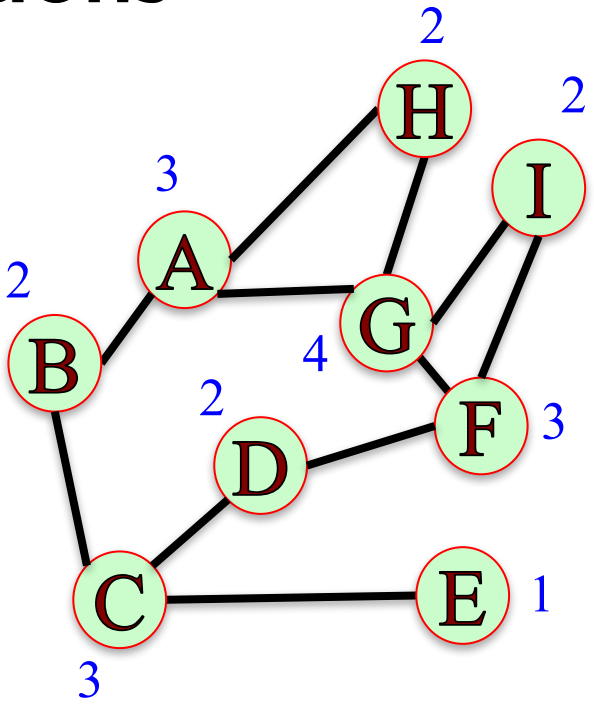
```
degrees = G.degree()
degree_values = sorted(set(degrees.values()))
histogram =
[list(degrees.values()).count(i)/float(nx.number_of_nodes(
G)) for i in degree_values]

import matplotlib.pyplot as plt
plt.bar(degree_values,histogram)
plt.xlabel('Degree')
plt.ylabel('Fraction of Nodes')
plt.show()
```



Degree Distributions

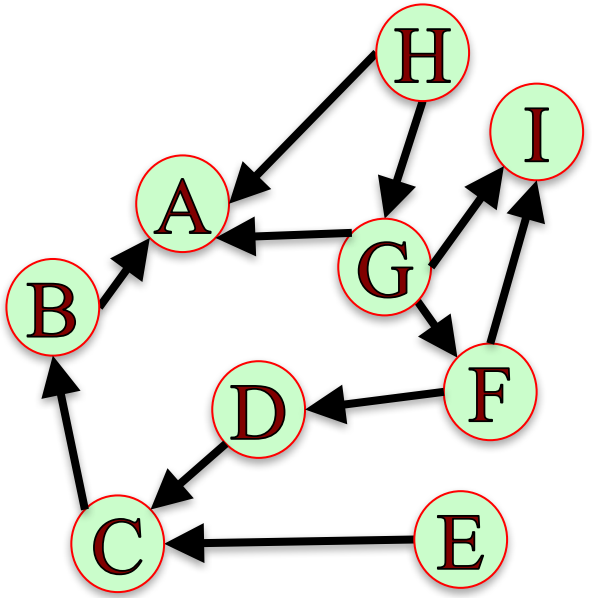
Plot of the degree distribution of this network:





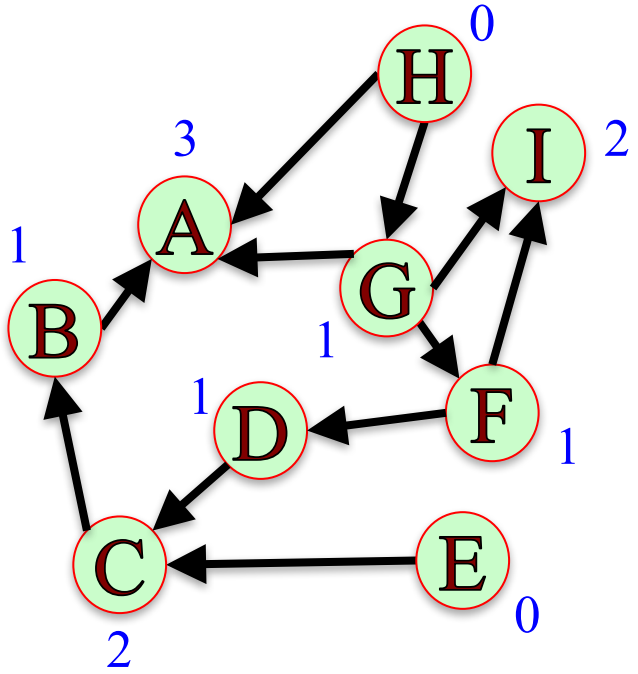
In-Degree Distributions

The **in-degree** of a node in a directed graph is the number of in-links it has.



In-Degree Distributions

The **in-degree** of a node in a directed graph is the number of in-links it has.



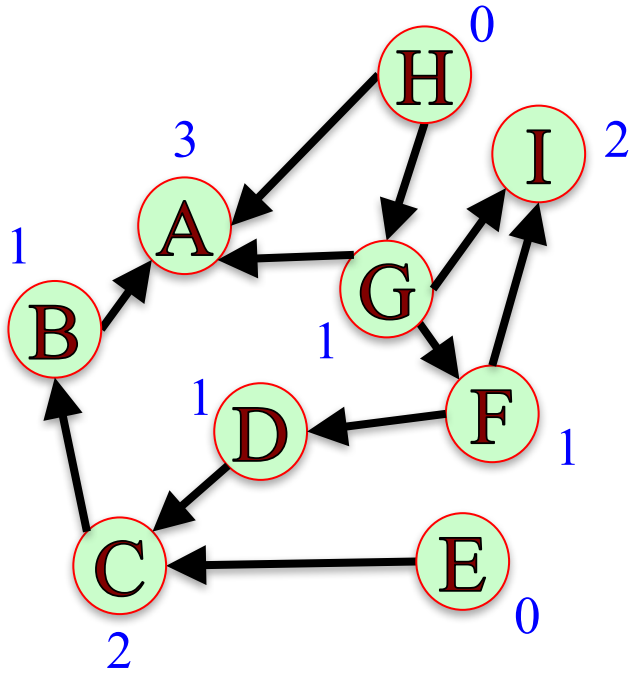
The in-degree distribution, $P_{in}(k)$, of this network has the following values:

$$P_{in}(0) = \frac{2}{9}, P_{in}(1) = \frac{4}{9}, P_{in}(2) = \frac{2}{9}, P_{in}(3) = \frac{1}{9}$$

In-Degree Distributions

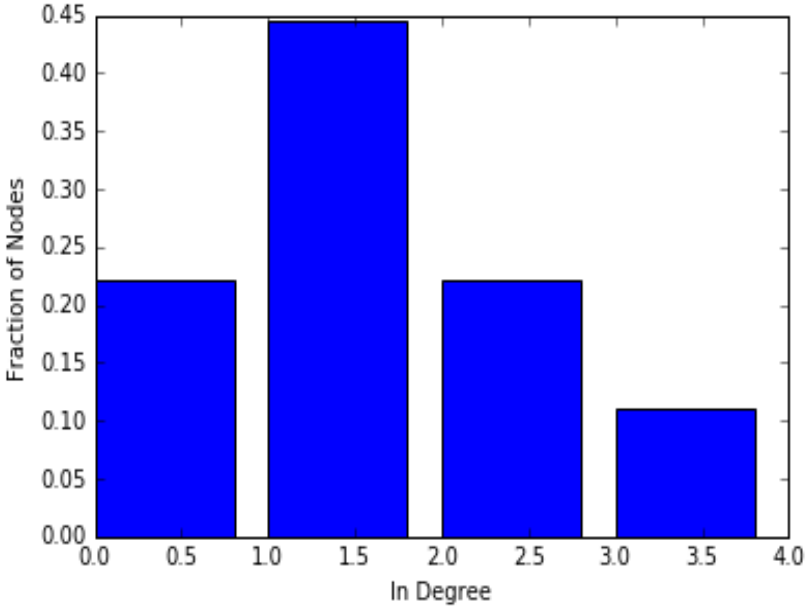
Plot of the degree distribution of this network:

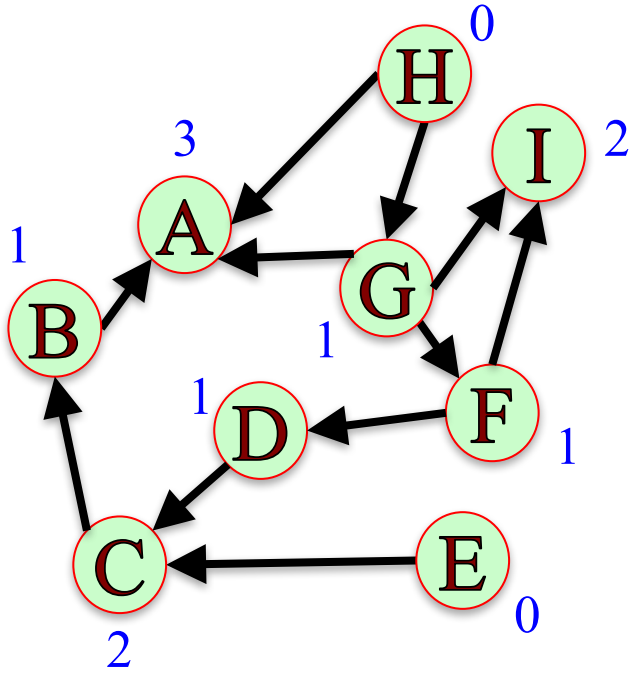
```
in_degrees = G.in_degree()
in_degree_values = sorted(set(in_degrees.values()))
histogram =
[list(in_degrees.values()).count(i)/float(nx.number_of_nodes(
G)) for i in in_degree_values]

plt.bar(in_degree_values,histogram)
plt.xlabel('In Degree')
plt.ylabel('Fraction of Nodes')
plt.show()
```



In-Degree Distributions

Plot of the degree distribution of this network:



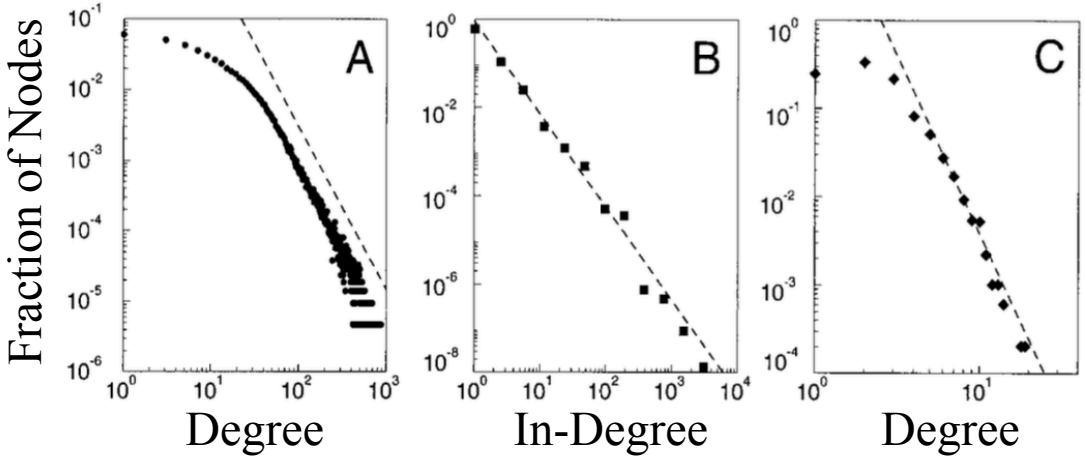


Degree Distributions in Real Networks

A – **Actors**: network of 225,000 actors connected when they appear in a movie together.

B – **The Web**: network of 325,000 documents on the WWW connected by URLs.

C – **US Power Grid**: network of 4,941 generators connected by transmission lines.



[Barabasi-Albert 99]

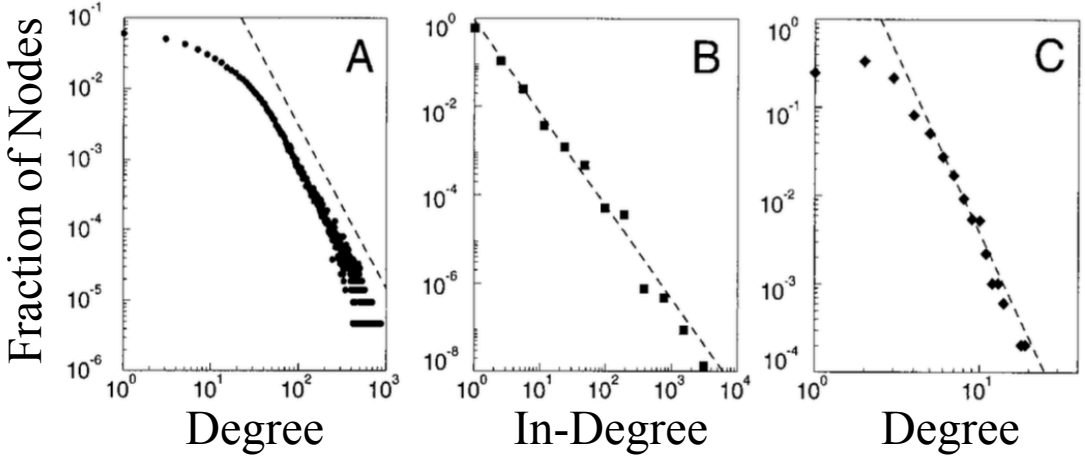
Degree distribution looks like a straight line when on a log-log scale. **Power law**: $P(k) = Ck^{-\alpha}$, where α and C are constants. α values: A: 2.3, B:2.1, C:4.

Modeling Networks

Networks with power law distribution have many nodes with small degree and a few nodes with very large degree.

What could explain power law degree distribution we observe in many networks?

Can we find a set of basic assumptions that explain this phenomenon?



[Barabasi-Albert 99]

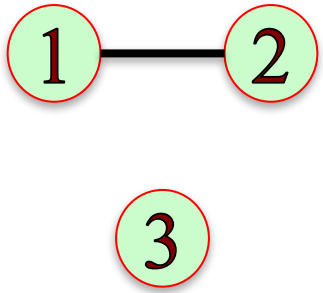
Degree distribution looks like a straight line when on a log-log scale. **Power law**: $P(k) = Ck^{-\alpha}$, where α and C are constants. α values: A: 2.3, B:2.1, C:4.

Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.

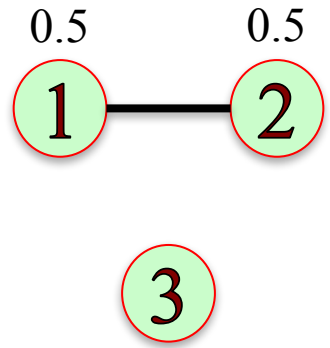
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



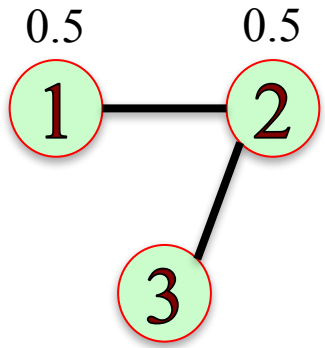
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



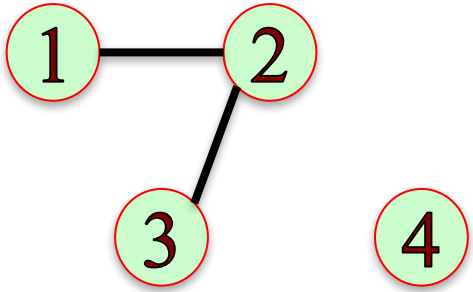
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



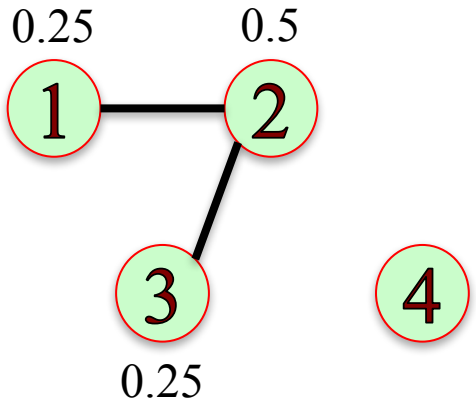
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



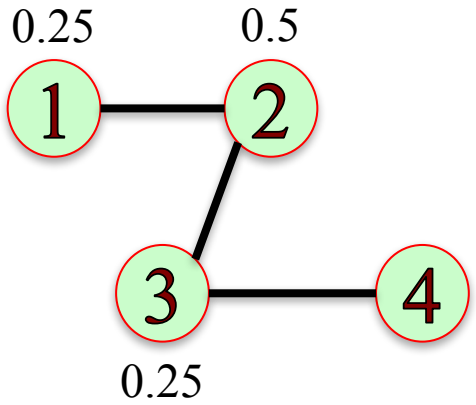
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



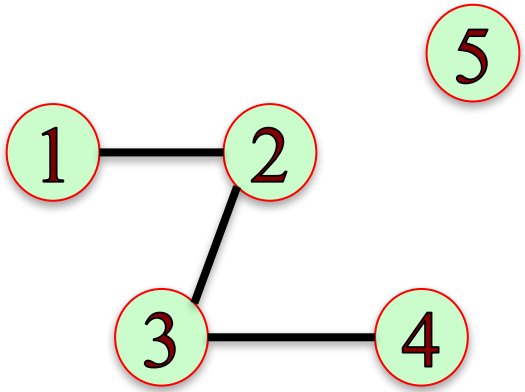
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



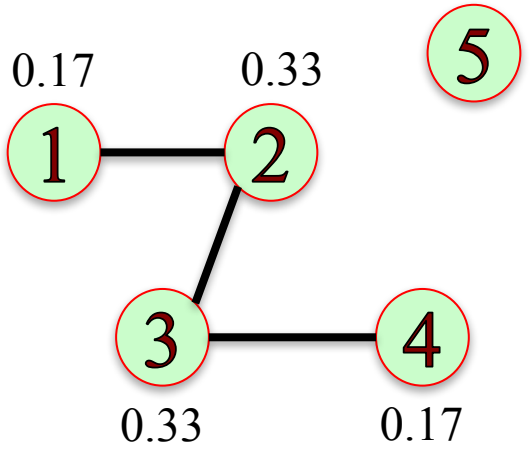
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



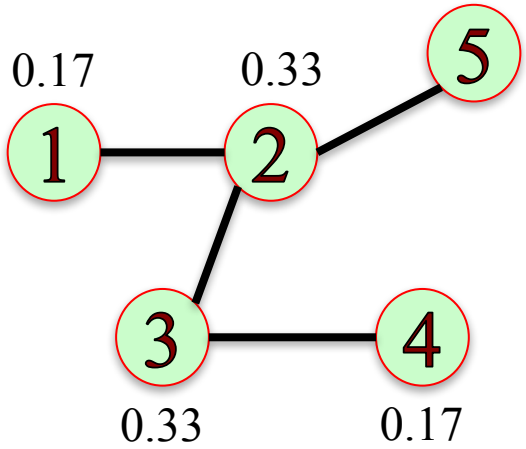
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



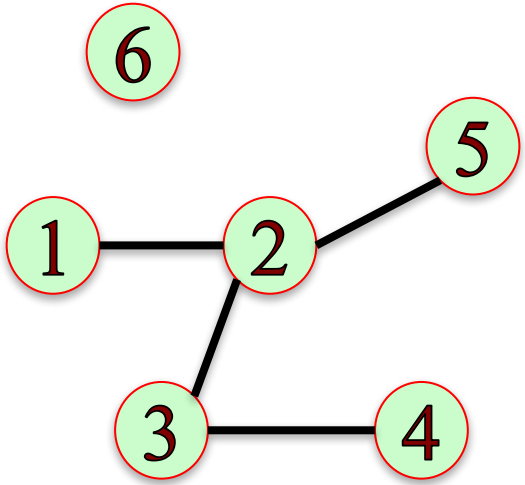
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



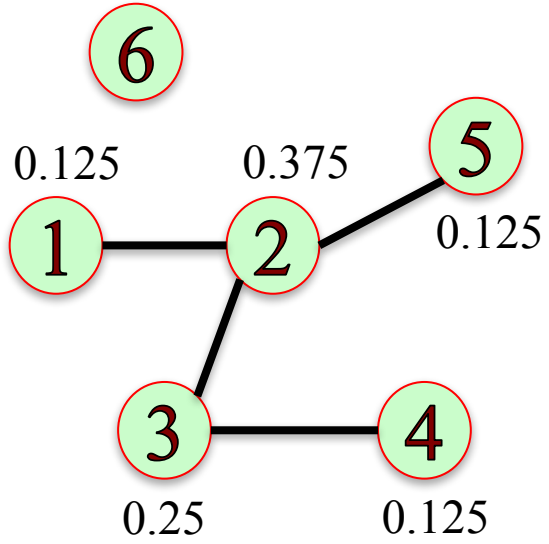
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



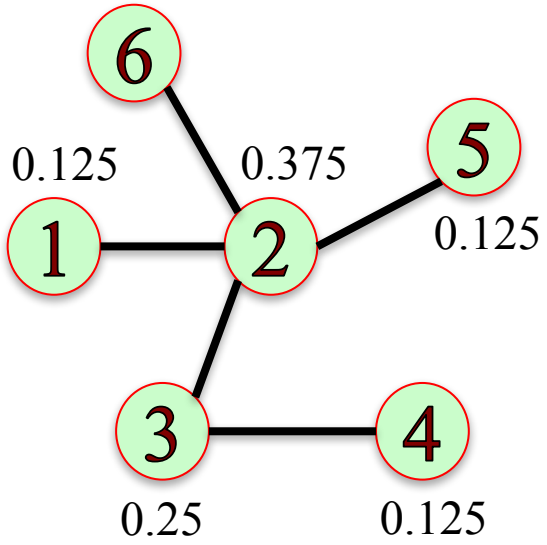
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



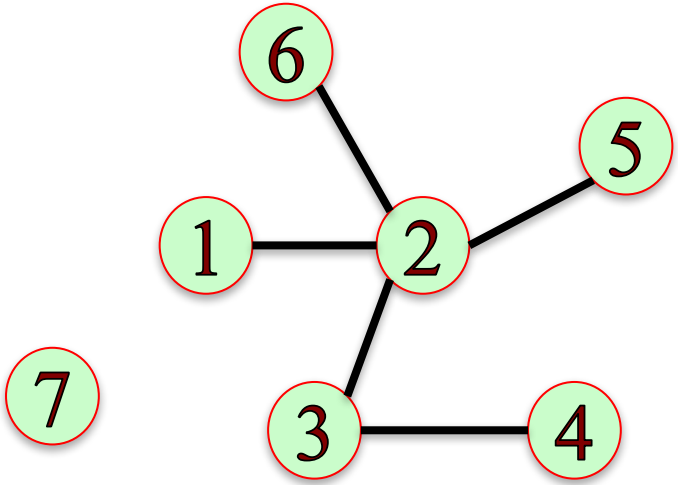
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



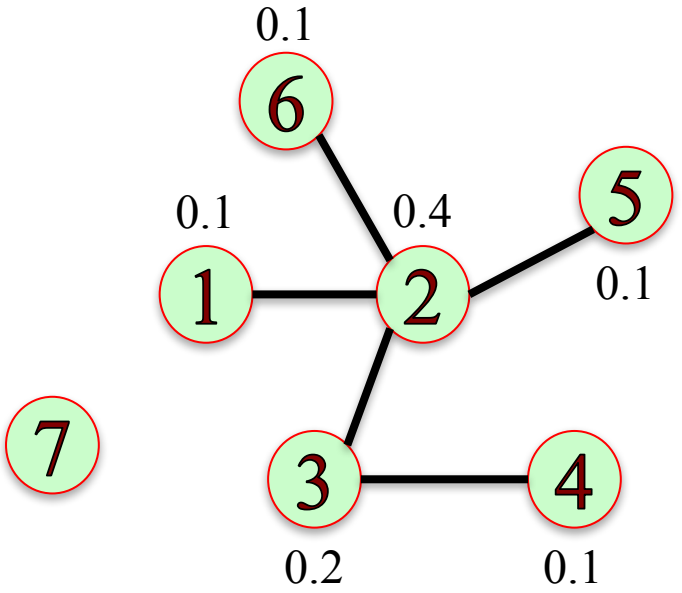
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



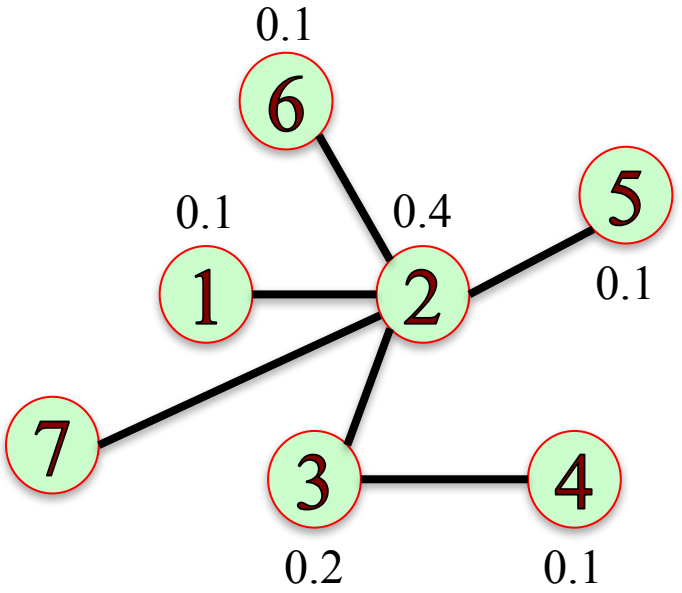
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



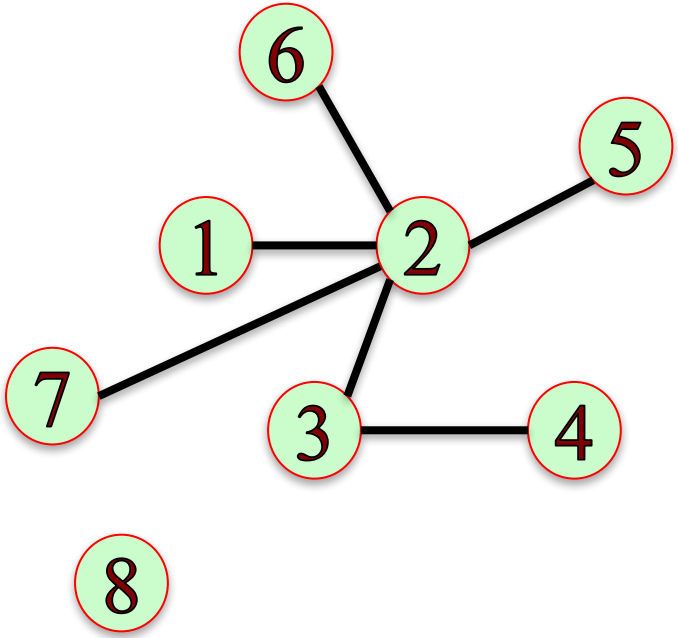
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



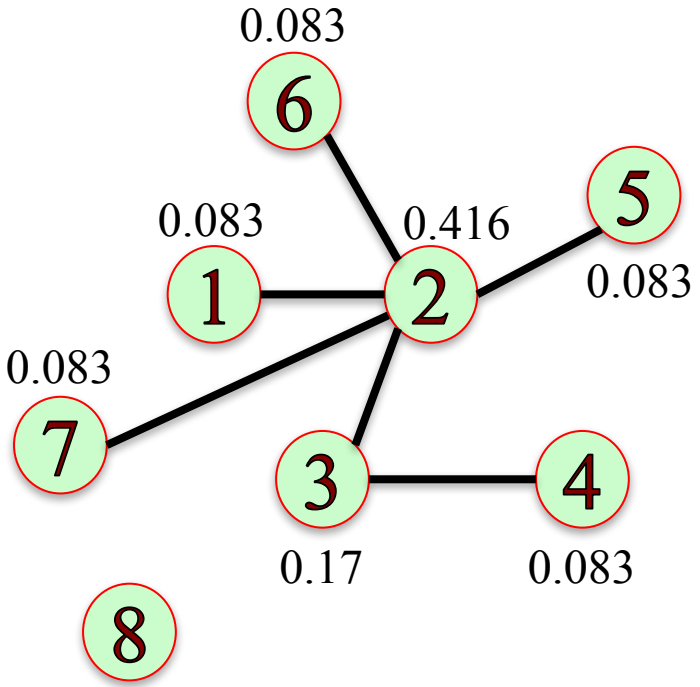
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



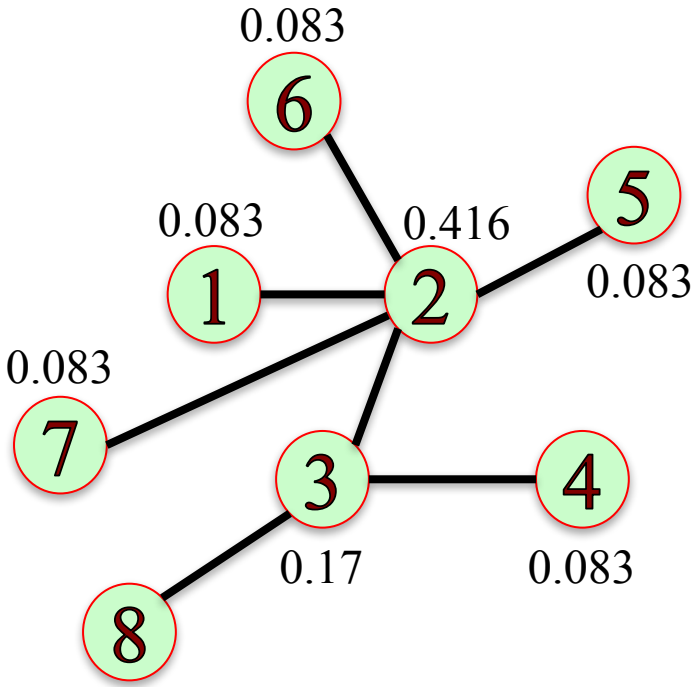
Preferential Attachment Model

- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



Preferential Attachment Model

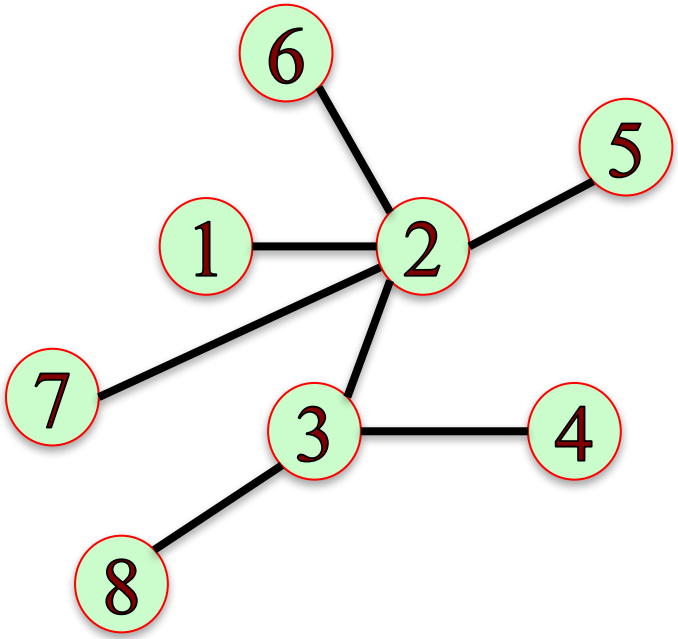
- Start with two nodes connected by an edge.
- At each time step, add a new node with an edge connecting it to an existing node.
- Choose the node to connect to at random with probability proportional to each node's degree.
- The probability of connecting to a node u of degree k_u is $k_u / \sum_j k_j$.



Preferential Attachment Model

As the number of nodes increases, the degree distribution of the network under the preferential attachment model approaches the power law $P(k) = Ck^{-3}$ with constant C.

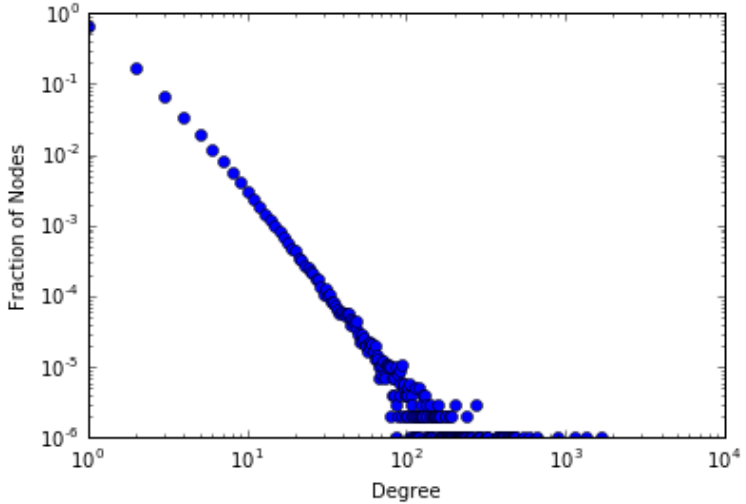
The preferential attachment model produces networks with degree distributions similar to real networks.



Preferential Attachment in NetworkX

barabasi_albert_graph(n, m) returns a network with *n* nodes. Each new node attaches to *m* existing nodes according to the Preferential Attachment model.

```
G = nx.barabasi_albert_graph(1000000,1)
degrees = G.degree()
degree_values = sorted(set(degrees.values()))
histogram =
[list(degrees.values().count(i))/float(nx.number_of_node
s(G)) for i in degree_values]

plt.plot(degree_values,histogram, 'o')
plt.xlabel('Degree')
plt.ylabel('Fraction of Nodes')
plt.xscale('log')
plt.yscale('log')
plt.show()
```



Summary

- The degree distribution of a graph is the probability distribution of the degrees over the entire network.
- Many real networks have degree distributions that look like power laws ($P(k) = Ck^{-\alpha}$).
- Models of network generation allow us to identify mechanisms that give rise to observed patterns in real data.
- The Preferential Attachment Model produces networks with a power law degree distribution.
- You can use barabasi_albert_graph(*n*, *m*) to construct a *n*-node preferential attachment network, where each new node attaches to *m* existing nodes.