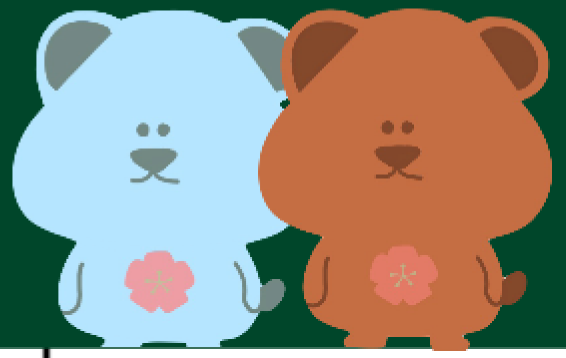


# Generative Models for Graphs

최예은, 최지우

EWHA EURON

# # Graph Generation

Image credit: Medium

**Social Networks**

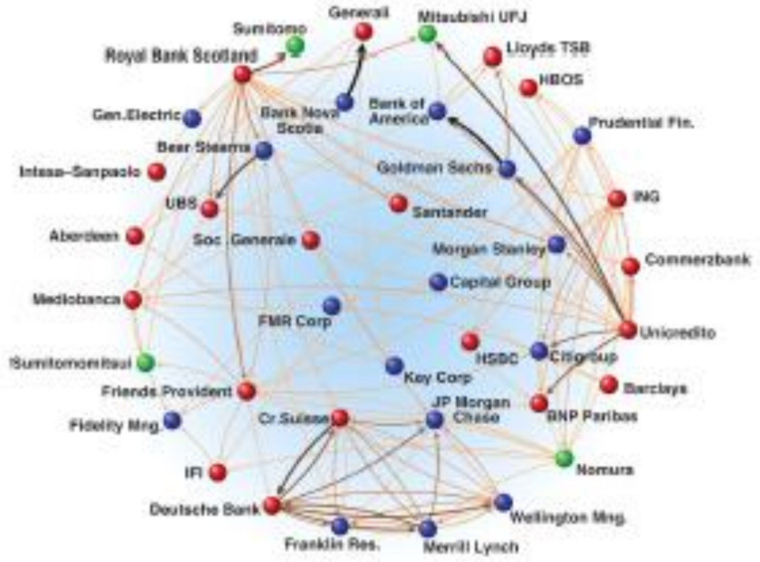


Image credit: Science

**Economic Networks**

Image credit: Lumen Learning

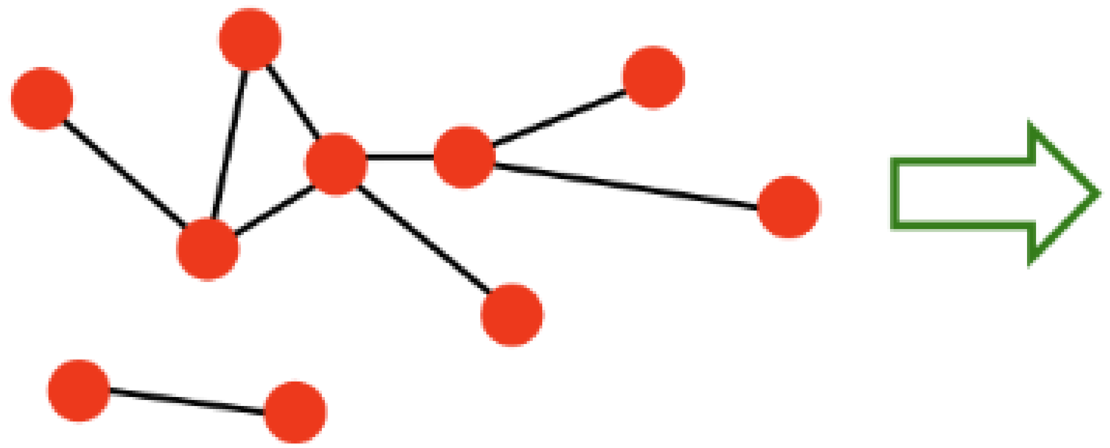
**Communication Networks**

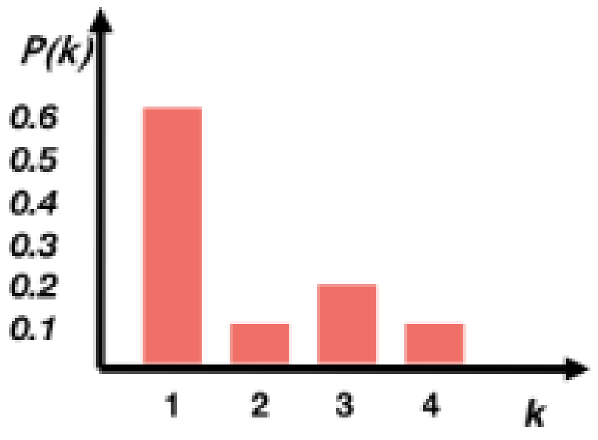
Graph에 대해 공부했으나 이제 graph를 어떻게 생성할 것인지에 대해 고민함

→ Graph generative models을 이용하여 실제 그래프와 유사한 그래프를 생성한다!

# # Properties of Real-world Graphs

Degree distribution



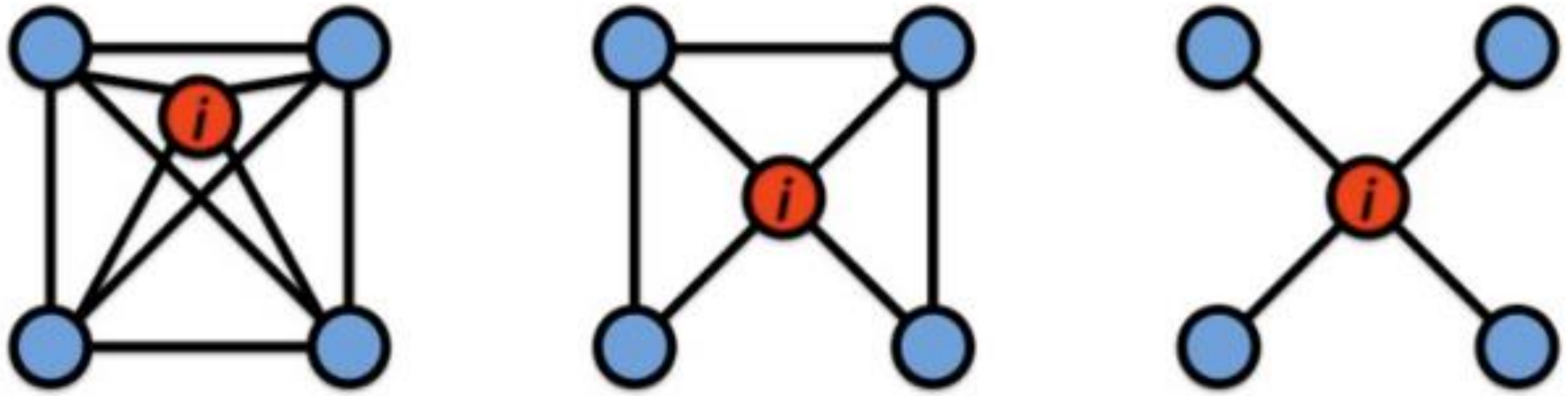



임의로 선택된 node가 degree K를 가질 확률을 의미

# # Properties of Real-world Graphs

Clustering coefficient

$$C_i = \frac{2e_i}{k_i(k_i - 1)}$$

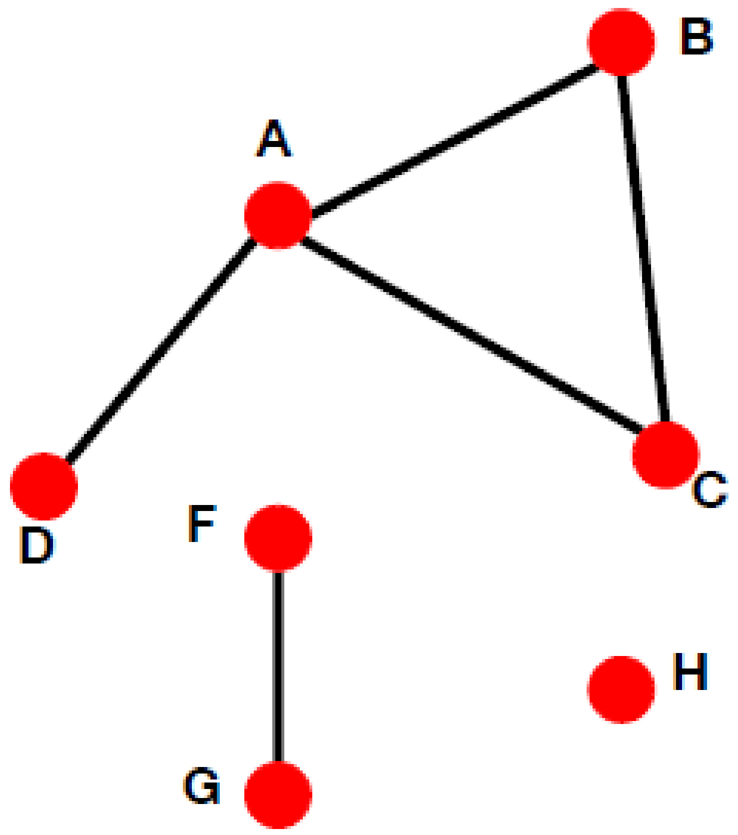


$C_i = 1$ $C_i = 1/2$ $C_i = 0$

Node i가 이웃들과 어떻게 연결되어 있는지 의미
ei = node I의 이웃 노드들간의 엣지의 수를 의미한다.

# # Properties of Real-world Graphs

Connectivity




-S로 표기하고 가장 큰 component의 크기를 나타낸다. S가 크면 giant component 라고도 한다.

-BFS 너비우선탐색을 통해 network가 연결됨을 확인한다.

# # Properties of Real-world Graphs

Path Length

$$\bar{h} = \frac{1}{2E_{\max}} \sum_{i,j \neq i} h_{ij}$$

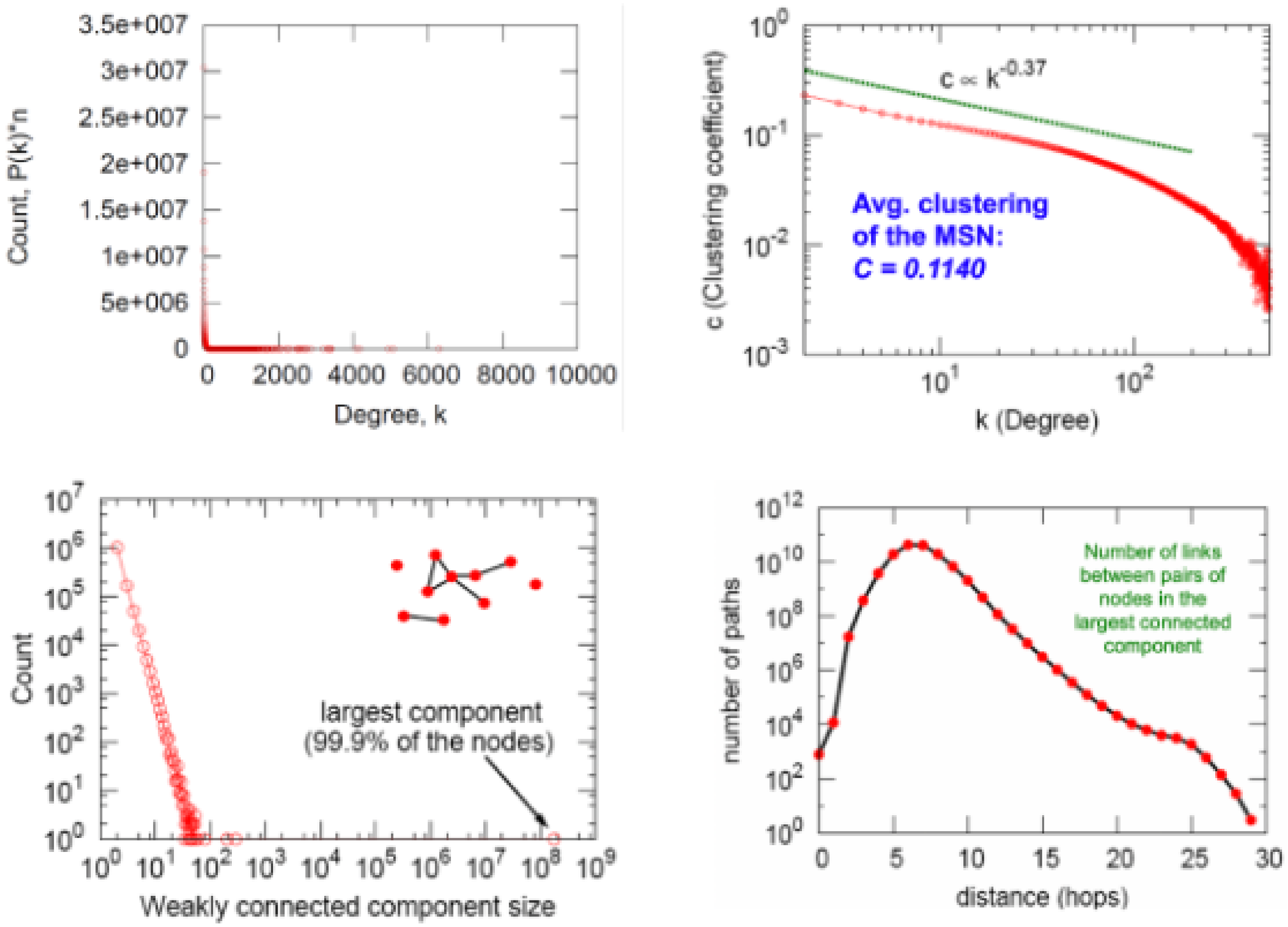
- $h_{ij}$ is the distance from node $i$ to node $j$
- $E_{max}$ is the max number of edges (total number of node pairs) $= n(n-1)/2$

최단 경로 기준 그래프 내 노드 쌍의 최대 거리를 diameter라고 한다. Average path length는 무한한 길이의 경로를 무시하기 위해 connected graph에서만 계산한다.

EWHA EURON

# # Case-Study

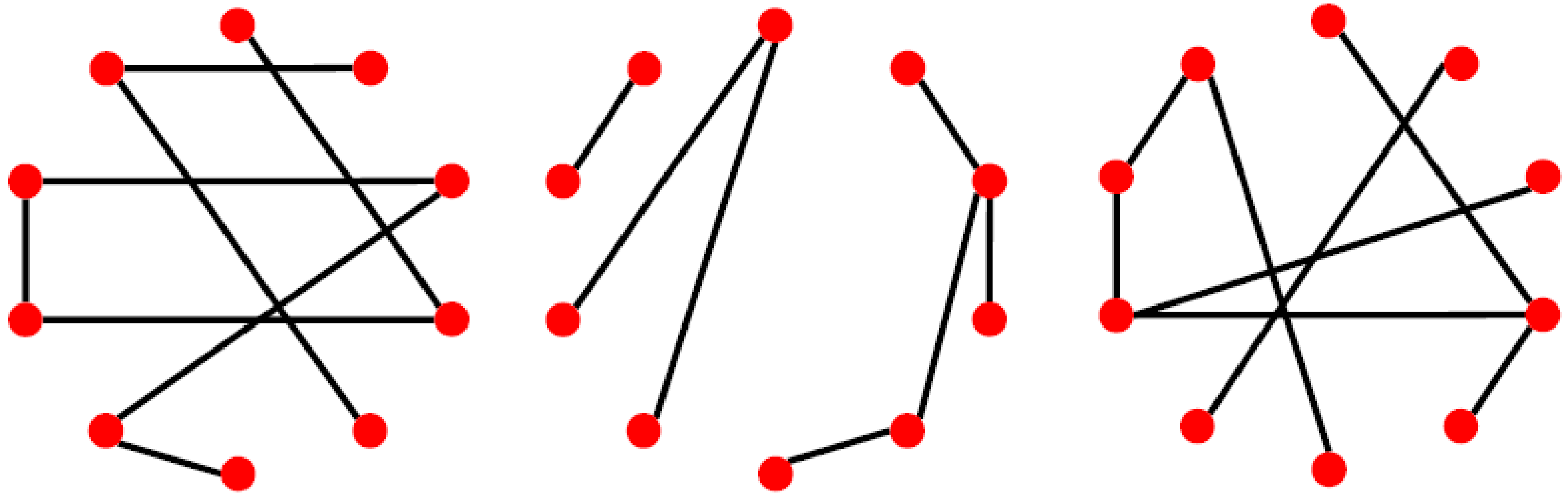

- **MSN Messenger:**
- **1 month of activity**
  - 245 million users logged in
  - 180 million users engaged in conversations
  - More than 30 billion conversations
  - More than 255 billion exchanged messages



# # Erdos-Renyi Random Graphs

Gnp : 엣지가 독립 항등 분포 (i.i.d.)의 확률 p를 가지는 노드 n에 대한 undirected graph

Properties of Gnp



n = 10
p= 1/6

n, p에 의해 그래프가 unique하게 결정되지 않는다.
Gnp는 properties로 P(k), C, h를 가진다.

# # Erdos-Renyi Random Graphs

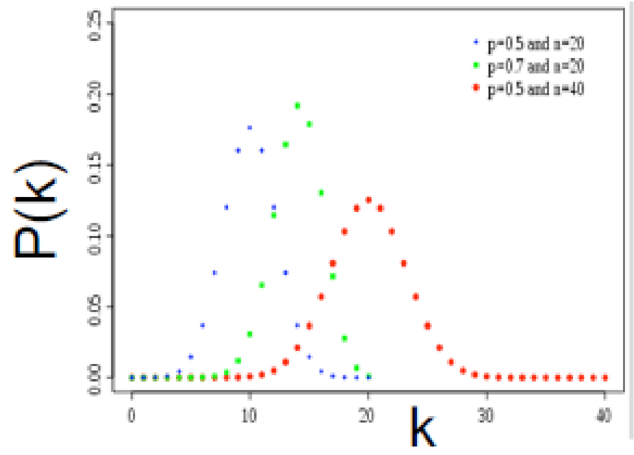
Degree distribution

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

Select $k$ nodes out of $n$-$1$

Probability of having $k$ edges

Probability of missing the rest of the $n$-$1$-$k$ edges




**Mean, variance of a binomial distribution**

$$\bar{k} = p(n-1)$$

$$\sigma^2 = p(1-p)(n-1)$$

Gnp의 P(k)는 이항분포를 따른다!
n-1개 중 선택한 k개는 엣지로 연결되고 나머지는 연결되지 않을 확률이 P(k)가 된다.

# # Erdos-Renyi Random Graphs

Clustering coefficient

$$C_i = \frac{2e_i}{k_i(k_i - 1)}$$

**So, <u>expected</u> E[$e_i$] is:** $= p\frac{k_i(k_i-1)}{2}$

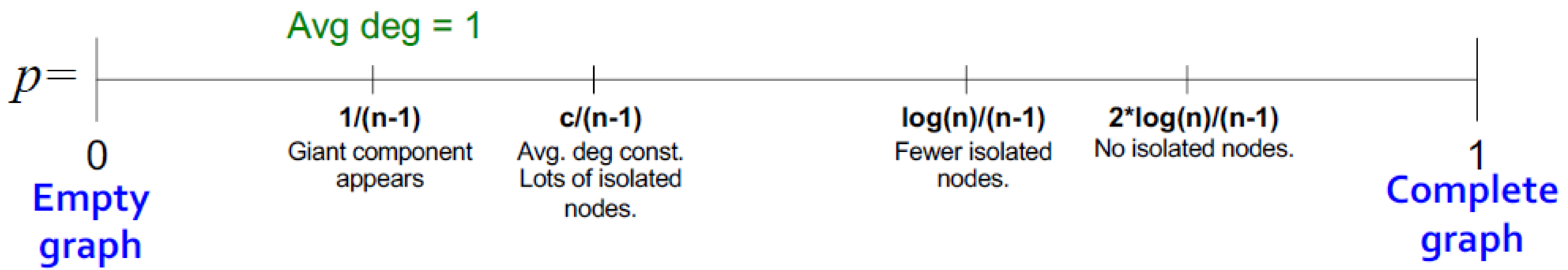
Each pair is connected with prob. $p$

Number of distinct pairs of neighbors of node $i$ of degree $k_i$

**Then *E[$C_i$]*:** $= \frac{p \cdot k_i(k_i-1)}{k_i(k_i-1)} = p = \frac{\bar{k}}{n-1} \approx \frac{\bar{k}}{n}$

# # Erdos-Renyi Random Graphs

Connected Components



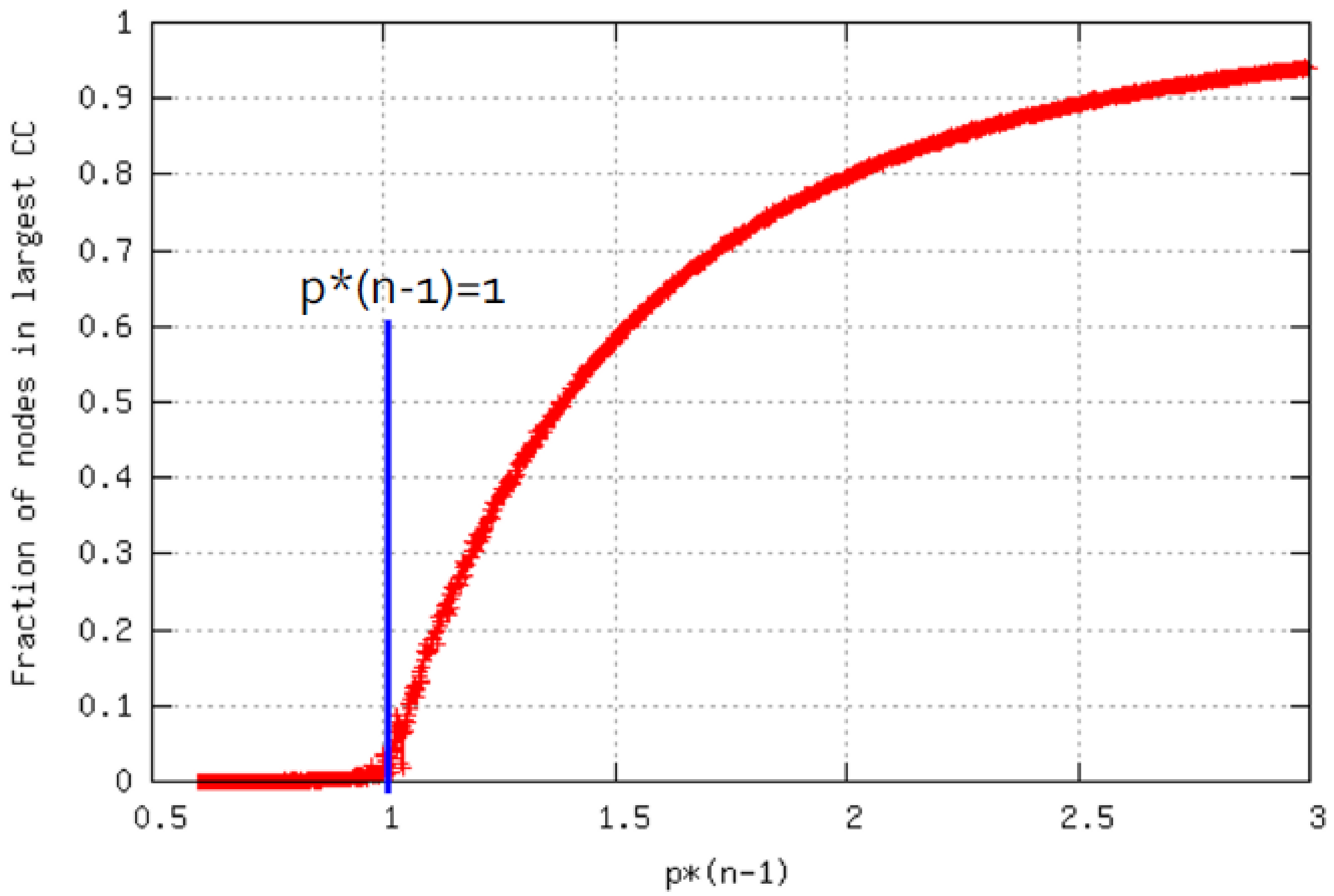

**Degree $k=1-\varepsilon$:** all components are of size $\Omega(\log n)$

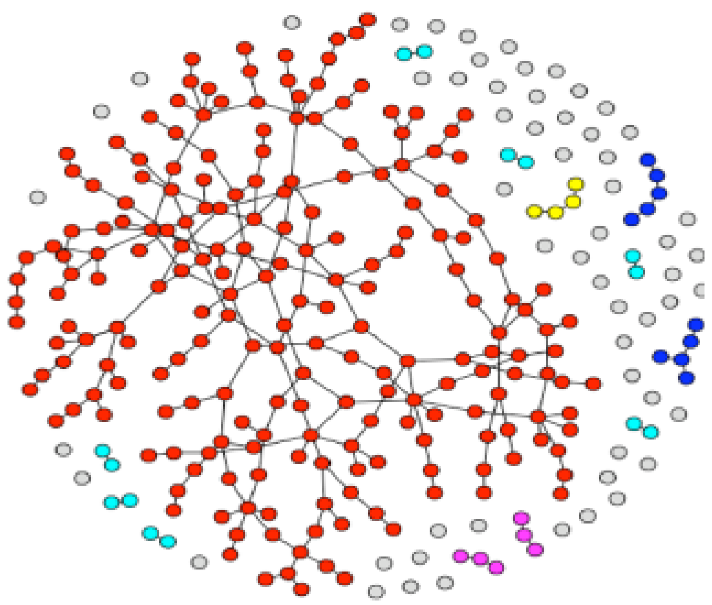
**Degree $k=1+\varepsilon$:** 1 component of size $\Omega(n)$, others have size $\Omega(\log n)$

- Each node has at least one edge in expectation

# # Erdos-Renyi Random Graphs

Connected Components



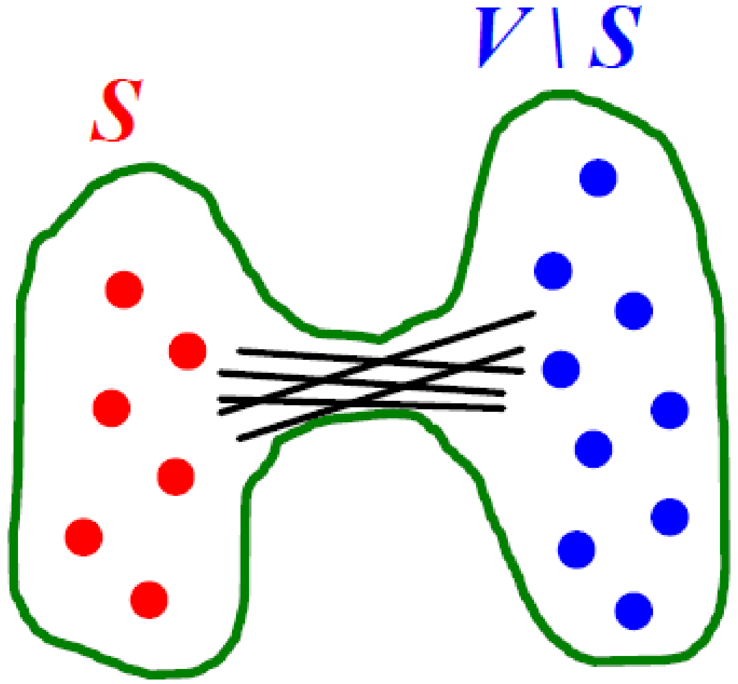
Fraction of nodes in the largest component

$G_{np}$, $n$=100,000, $k=p(n-1)$ = 0.5 ... 3

# Expansion

Path length를 이해하기 위해서 Expansion 개념을 살펴볼 필요가 있다!

$$\alpha = \min_{S \subseteq V} \frac{\#\, edges\ leaving\ S}{\min(|S|, |V \setminus S|)}$$




그래프 G(V,E)에 대해 V의 subset S를 만들기 위해 끊어주어야 하는 edge의 비율

# Expansion : Random Graphs

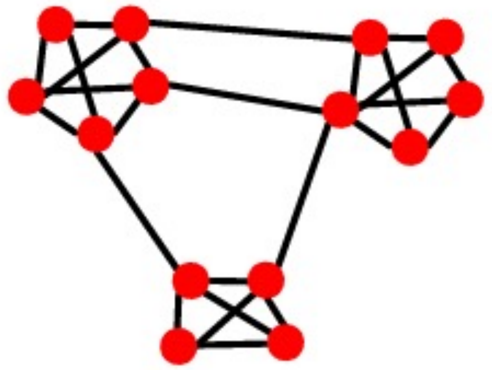
Expansion은 measure of robustness이다.

⇒ ex. L 노드를 disconnect 하려면 a(알파)*L 개 이상의 엣지를 끊어야 함.

- Low expansion:

- High expansion:

- Social networks:
  - “Communities”

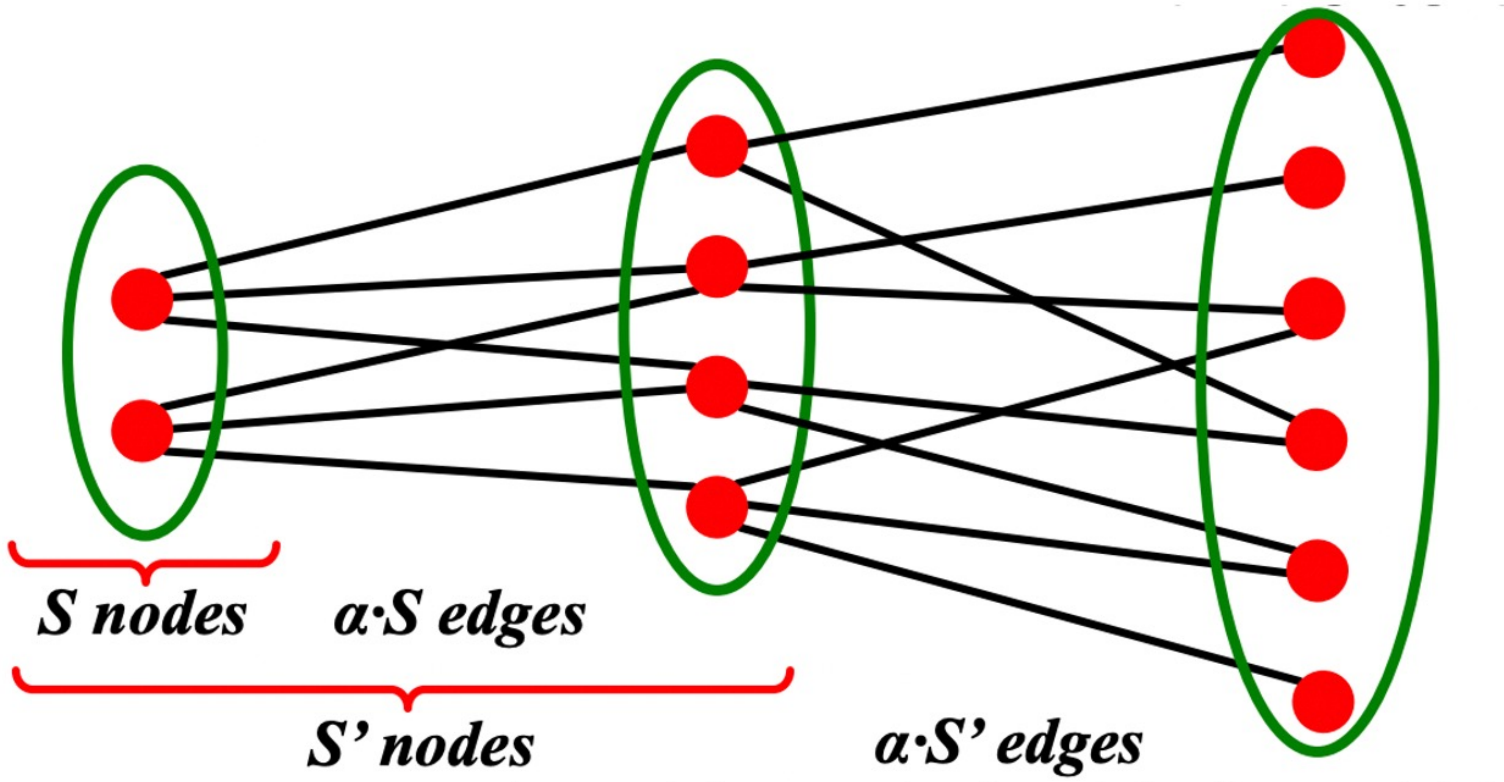


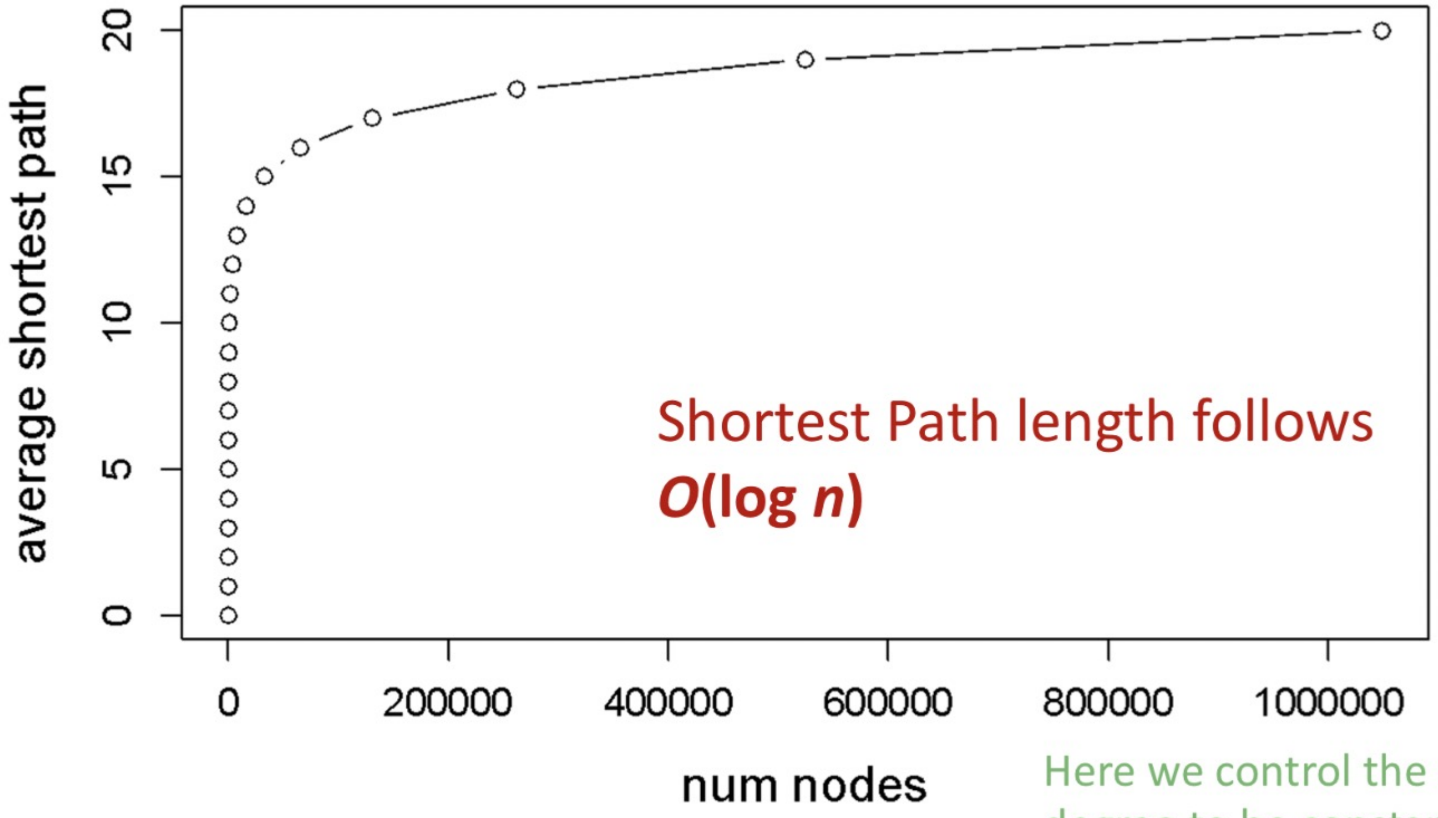
subset을 쉽게 만들 수 있는 구조면 low expansion을,
어려운 구조면 high expansion을 가진다.

# Expansion : Random Graphs

Expansion은 measure of robustness이다.

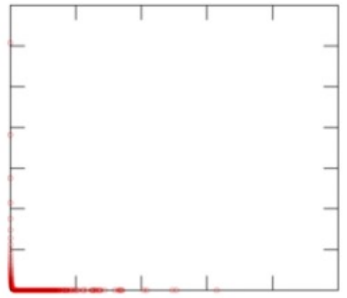
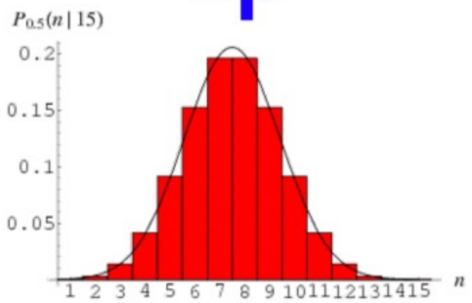
- Expansion이 $\alpha$이고 $n$개의 노드를 가지는 그래프의 모든 노드 쌍에 대한 path of length는 $O((logn)/\alpha)$이다.
- Expansion이 크면 알파도 커지고, path length 는 작아짐




Here we control the average degree to be constant

# MSN vs G_np

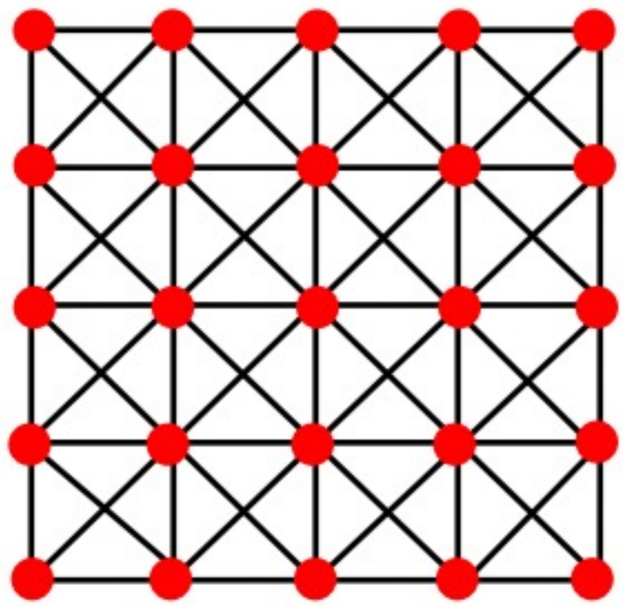
| | MSN | $G_{np}$ | n=180M |
|---|---|---|---|
| **Degree distribution:** |  |  | ☒ |
| **Avg. path length:** | **6.6** | $O(\log n)$<br>$h \approx 8.2$ | ☑ |
| **Avg. clustering coef.:** | *0.11* | $\bar{k} / n$<br>$C \approx 8 \cdot 10^{-8}$ | ☒ |
| **Largest Conn. Comp.:** | *99%* | GCC exists when $\bar{k}$>1.<br>$\bar{k} \approx 14.$ | ☑ |

Real world network 는 G_np처럼 random 하지 않다! !

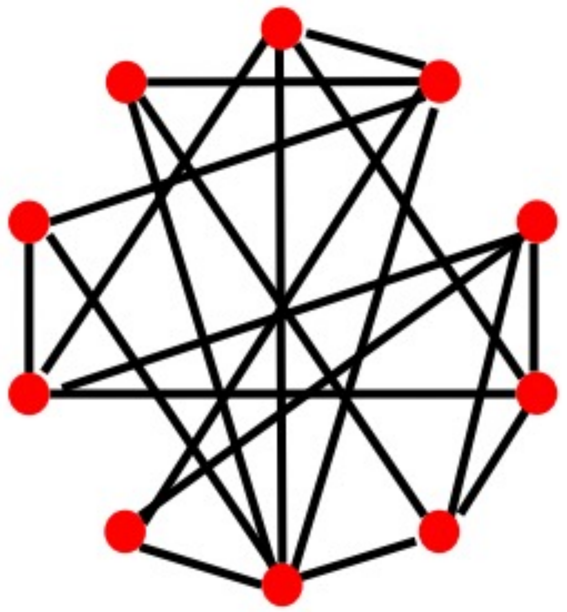
# Small-World Model

# Motivation for small world

- **실제 그래프**는 local한 구조를 가져 **clustering coefficient가 높으면서도 낮은 diameter**를 가진다.
- *Gnp*는 낮은 clustering coefficient를 가져 이를 제대로 반영하지 못한다.



**Vs.**

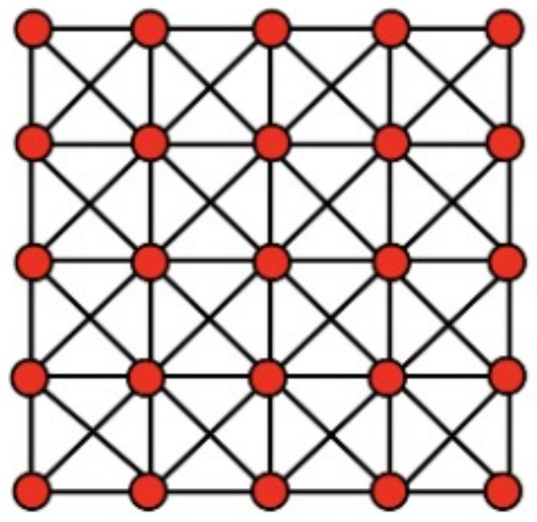


**Regular lattice graph:**
High clustering coefficient
High diameter

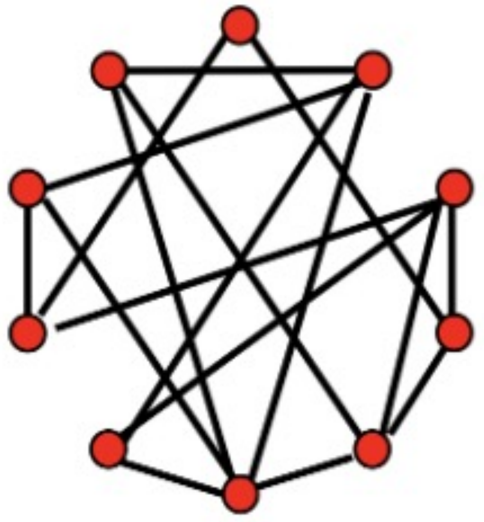
**$G_{np}$ random graph:**
Low clustering coefficient
Low diameter

**Can we have high clustering while also having short paths?**

# Small-world Graphs: idea



Interpolate



Regular lattice graph:
High clustering coefficient
High diameter

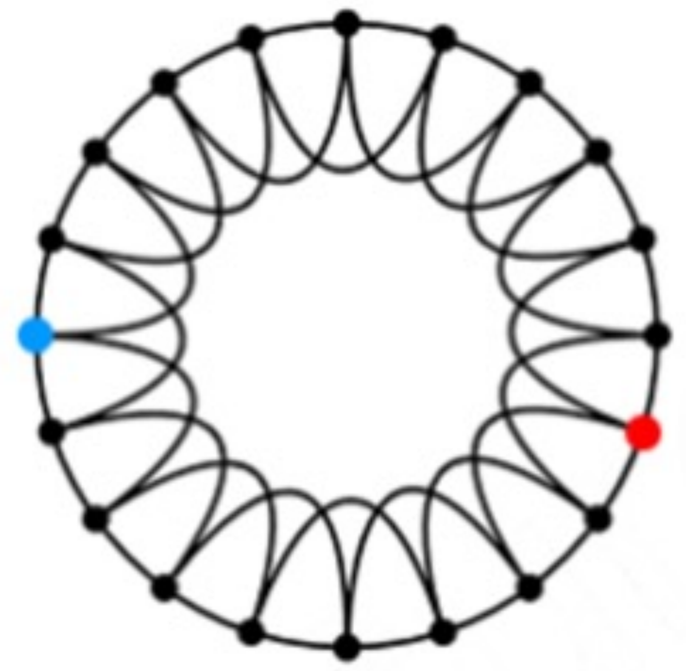
Small-world graph:
High clustering coefficient
Low diameter

$G_{np}$ random graph:
Low clustering coefficient
Low diameter

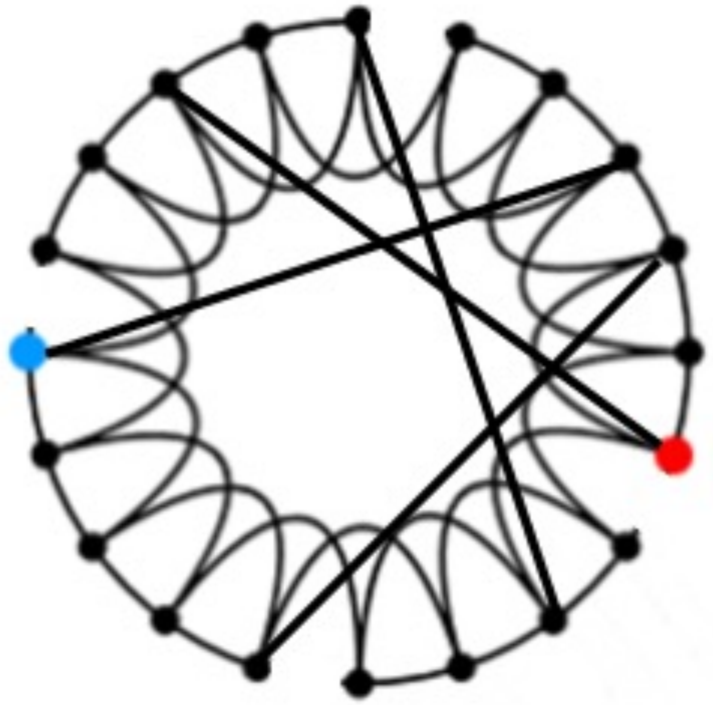
# Solution: the small-world model

Small world model 에는 2가지 components를 따름

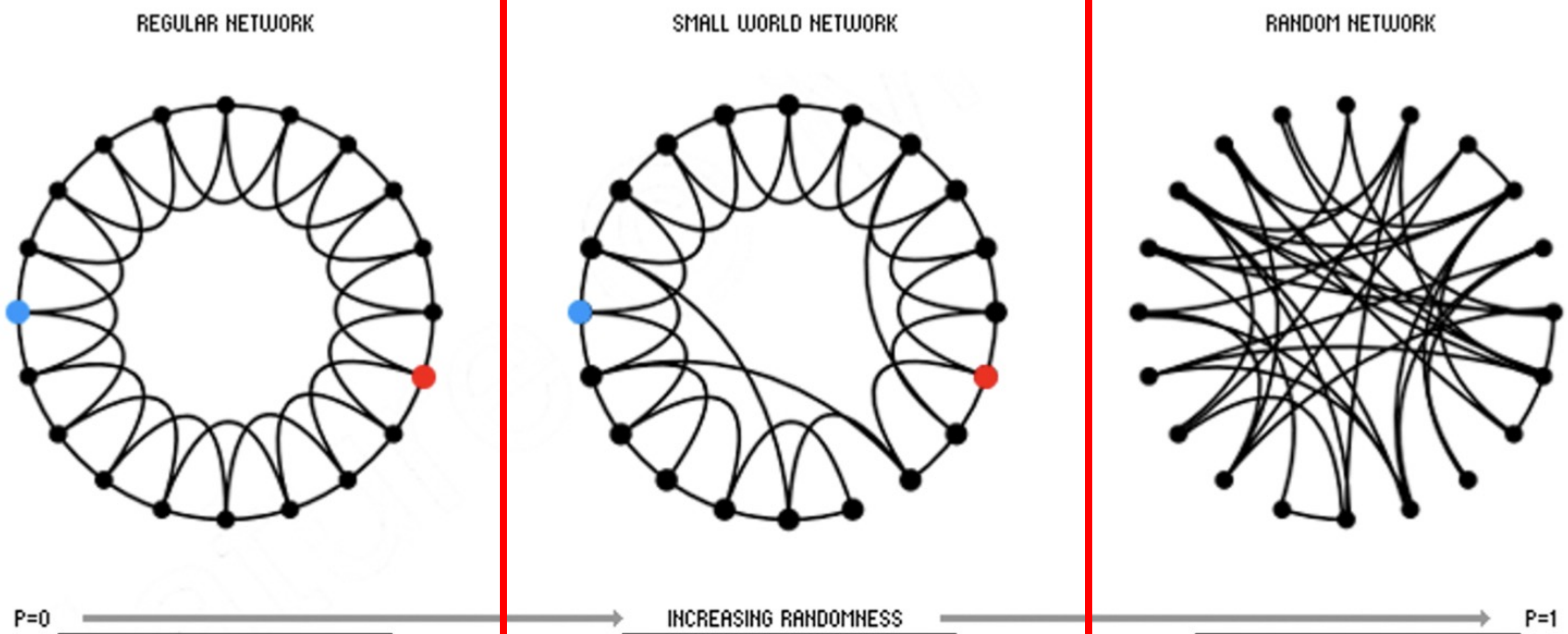
1. Start with a low-dimensional regular lattice



2. Rewire: introduce randomness ("shortcuts")

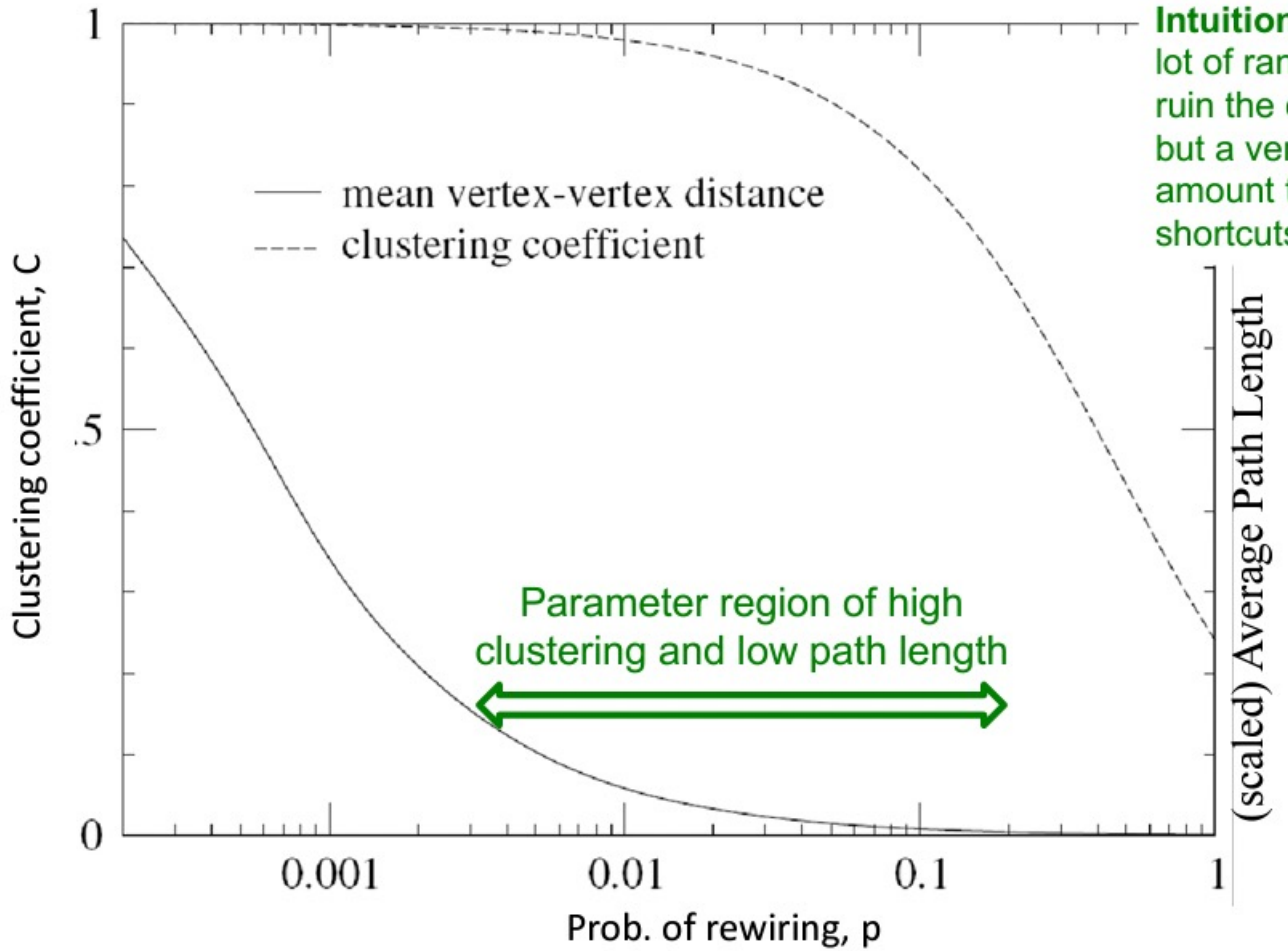


# Solution: the small-world model




| Regular network | Small world network | Random network |
| --- | --- | --- |
| High clustering<br>High diameter | High clustering<br>Low diameter | Low clustering<br>Low diameter |
| $h = \frac{N}{2\bar{k}}$ $\quad C = \frac{1}{2}$ | | $h = \frac{\log N}{\log \alpha}$ $\quad C = \frac{\bar{k}}{N}$ |

# Solution: the small-world model

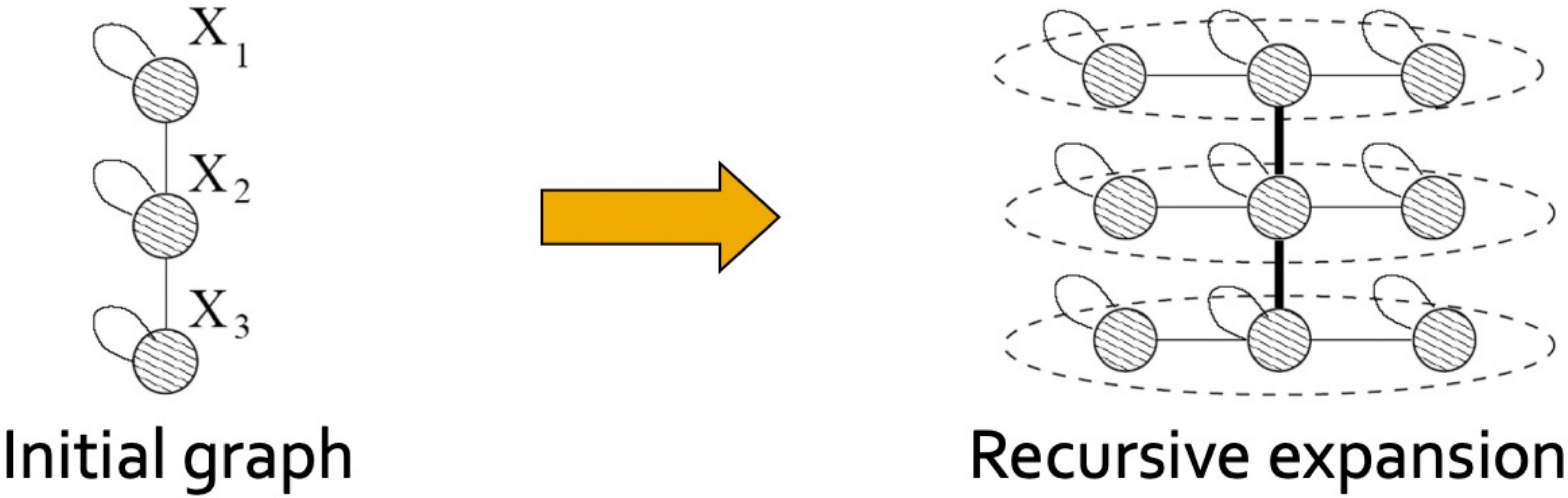



**Intuition:** It takes a lot of randomness to ruin the clustering, but a very small amount to create shortcuts.

# Kronecker graph Model

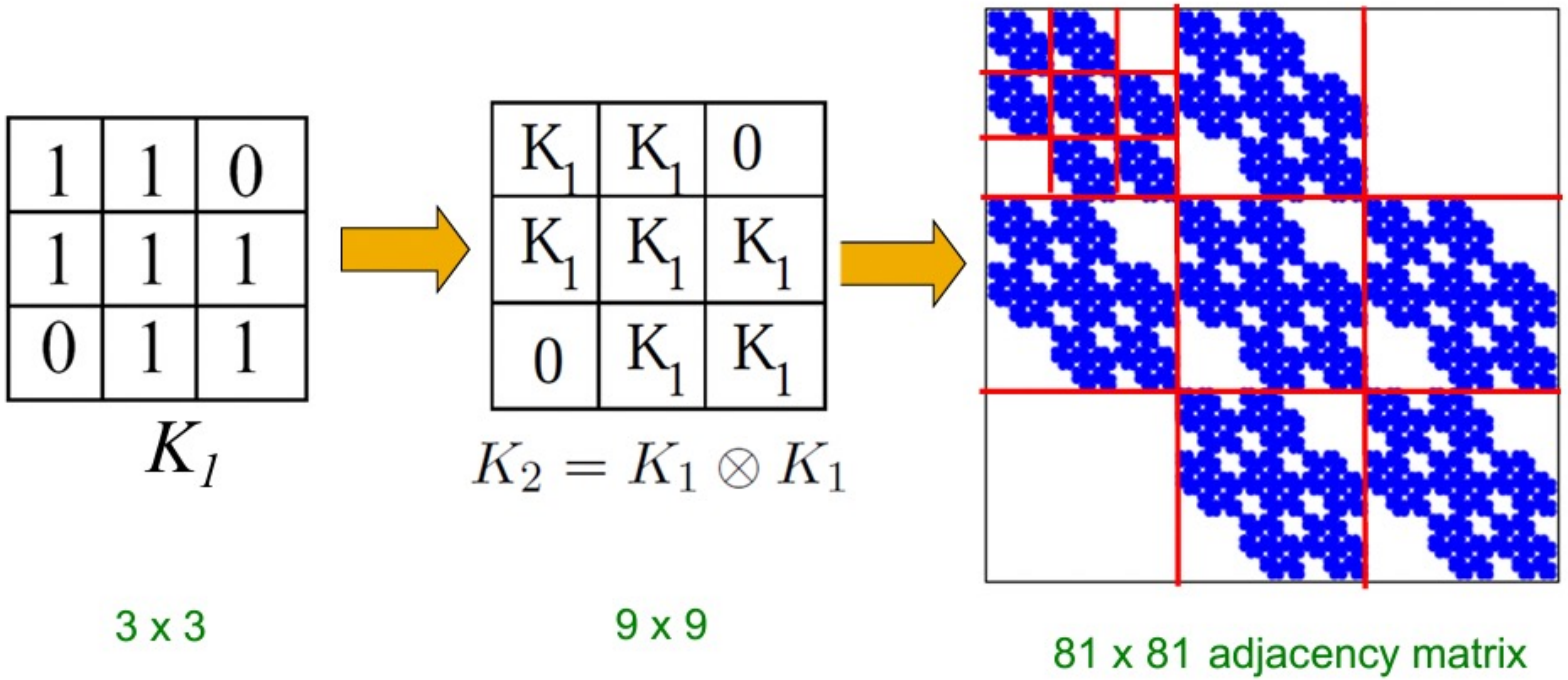
# Idea: resursive graph generation

Resursive한 네트워크 구조! => Self-similarity




- **Object는 자기 자신의 일부와 비슷**하므로 네트워크를 **재귀적**으로 구성할 수 있다.
- Kronecker product를 통해 **self-similar 행렬**을 만든다.

# Kronecker Graph




Kronecker graph는 kronecker product를 초기 행렬 $K1$에 반복적으로 행하여 만들 수 있다.

# * Kronecker Product

환 $R$ 위의 $m \times n$ 행렬 $M$과 $p \times q$ 행렬 $N$이 주어졌다고 하자.

$$M = \begin{pmatrix} M_{11} & \cdots & M_{1n} \\ \vdots & & \vdots \\ M_{m1} & \cdots & M_{mn} \end{pmatrix} \in \operatorname{Mat}(m, n; R)$$

$$N = \begin{pmatrix} N_{11} & \cdots & N_{1q} \\ \vdots & & \vdots \\ N_{p1} & \cdots & N_{pq} \end{pmatrix} \in \operatorname{Mat}(p, q; R)$$

그렇다면, $M$과 $N$의 **크로네커 곱**

$$M \otimes N \in \operatorname{Mat}(mp, nq; R)$$

은 다음과 같은 성분을 갖는 $mp \times nq$ 행렬이다.

$$M \otimes N = \begin{pmatrix} M_{11}N & \cdots & M_{1n}N \\ \vdots & & \vdots \\ M_{m1}N & \cdots & M_{mn}N \end{pmatrix}$$

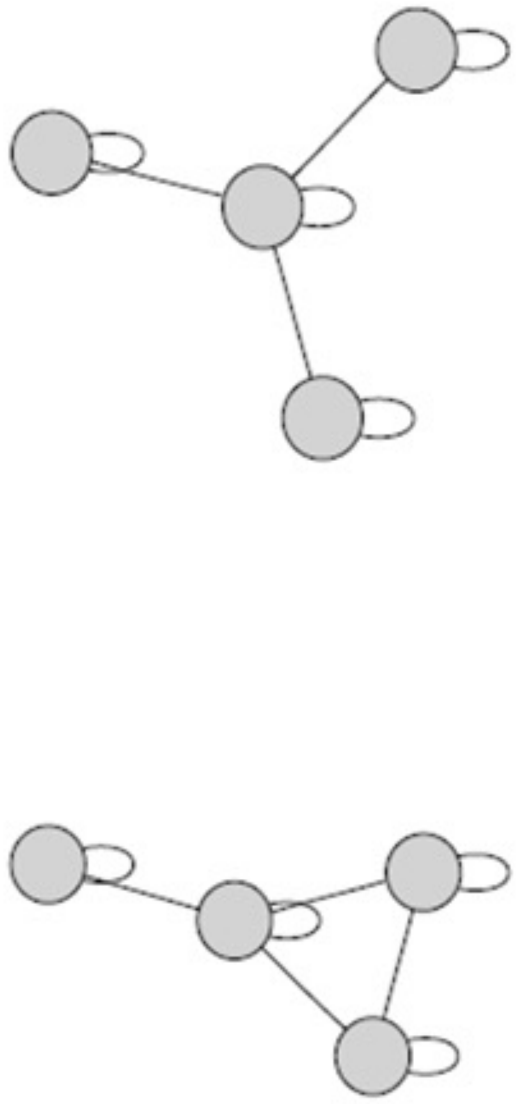
즉,

$$(M \otimes N)_{(a-1)p+i,(b-1)q+j} = M_{ab}N_{ij}$$

이다.

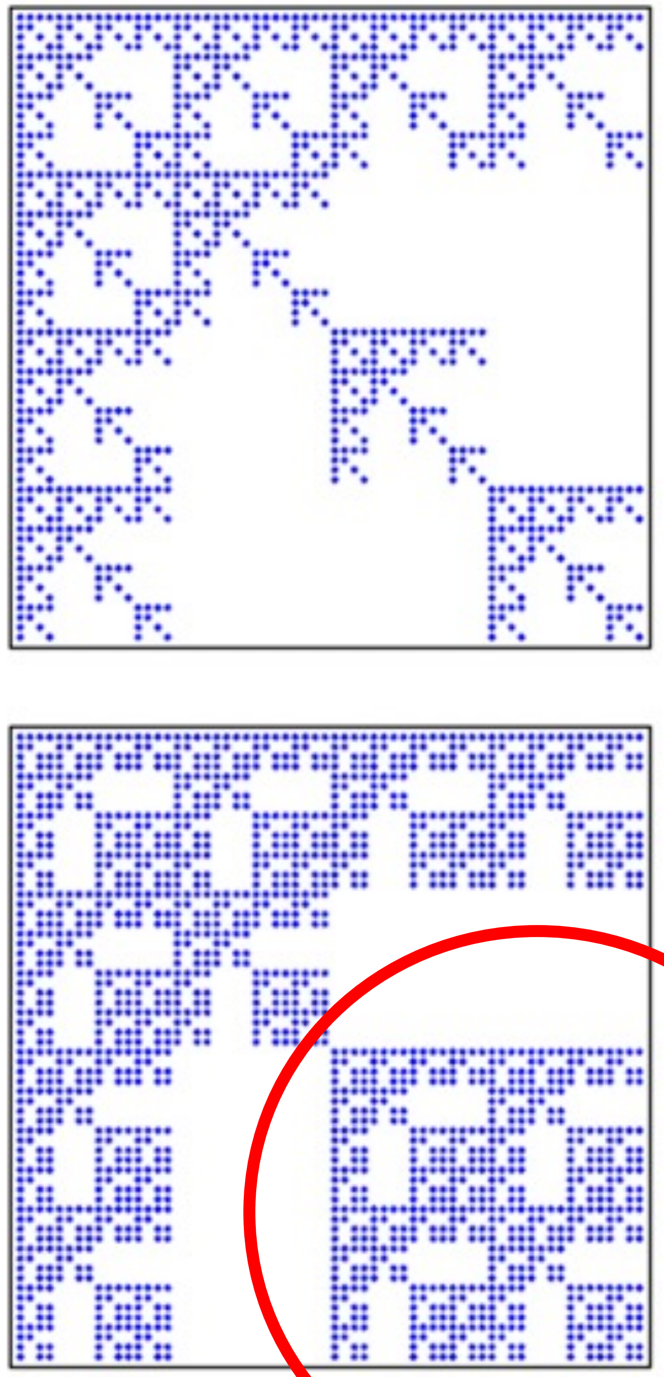
선형대수학에서 **크로네커 곱**(Kronecker product)은 두 행렬의 텐서곱을 구체적으로 표현하는 행렬이다. $m \times n$ 행렬과 $p \times q$ 행렬의 크로네커 곱은 크기 $mp \times nq$의 더 큰 행렬이다.

# Kronecker Graph example



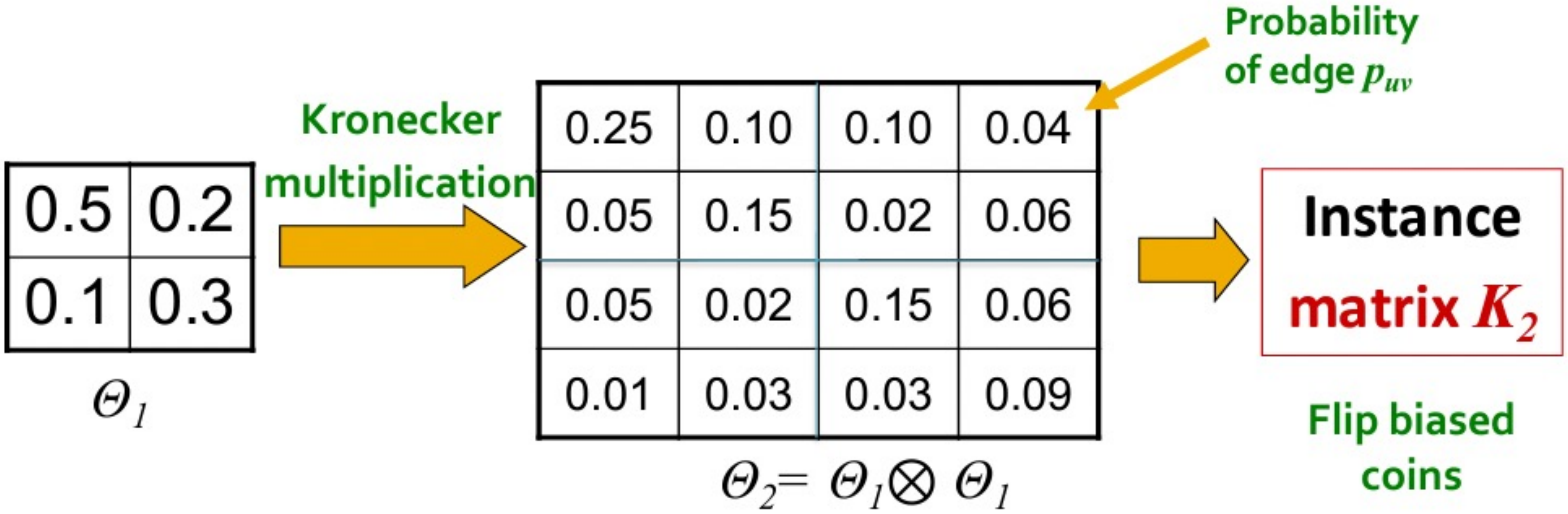
| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 |



Initiator $K_1$    $K_1$ adjacency matrix    $K_3$ adjacency matrix

# Stochastic Kronecker graphs




- $N1 \times N1$ 의 확률 matrix $\theta 1$을 만든다.
- *kth* Kronecker power $\theta k$를 계산한다.
- $\theta k$의 entry $p_uv$에 따라 엣지를 생성한다.

# Generation of Kronecker graphs

Probability of edge $p_{uv}$

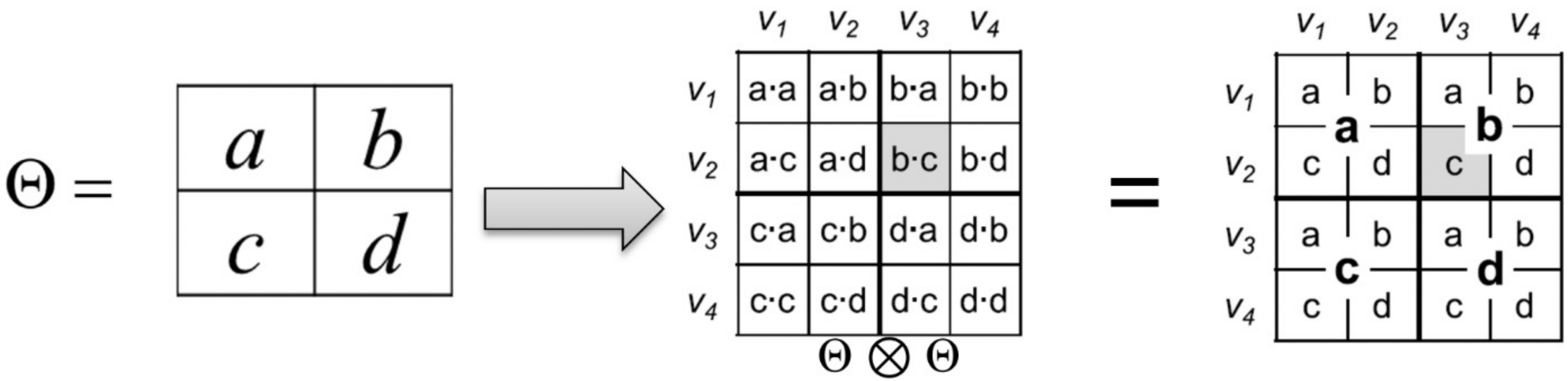
| | | | |
|---|---|---|---|
| 0.25 | 0.10 | 0.10 | 0.04 |
| 0.05 | 0.15 | 0.02 | 0.06 |
| 0.05 | 0.02 | 0.15 | 0.06 |
| 0.01 | 0.03 | 0.03 | 0.09 |

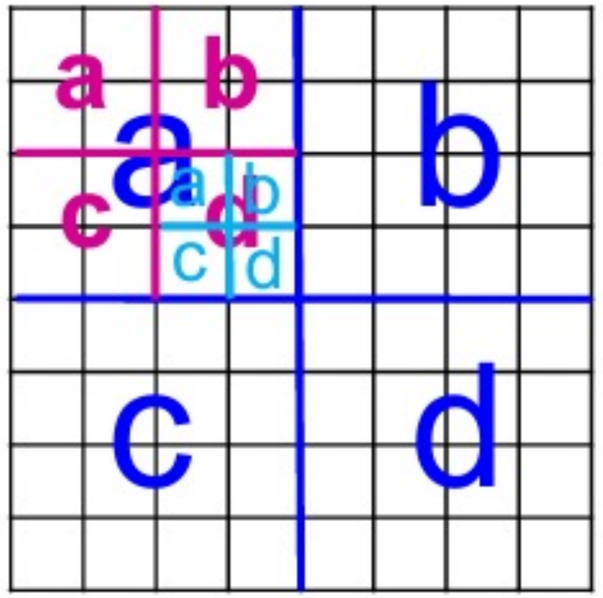
Flip biased coins

| | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |

Need to flip $n^2$ coins!! Way too slow!!

# Generation of Kronecker graphs



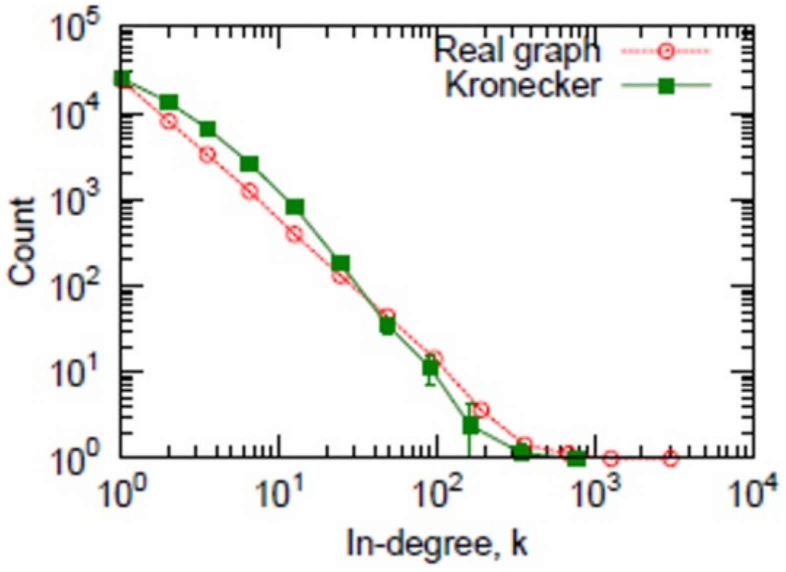




- $\theta$로부터 **normalized matrix** $L$을 만든다.
- $L$의 확률에 따라 가장 큰 **4분할 영역 중 한 영역**을 선택한다.
- 그 영역 또한 분할되어 있다면 **재귀적으로 확률에 따라 선택**한다.
- **단일 cell**이 나올 때까지 반복하며 최종적으로 선택되면 **1을 할당하여 edge를 만든다.**
- 위 과정을 기대 엣지 수 $E=(a+b+c+d)m$가 될 때까지 반복한다.

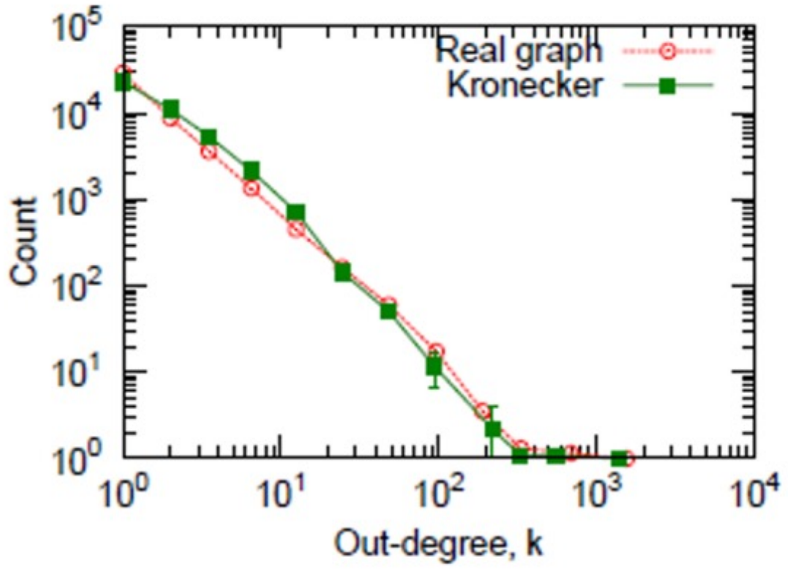
# Real and Kronecker

- **Real and Kronecker are very close:**

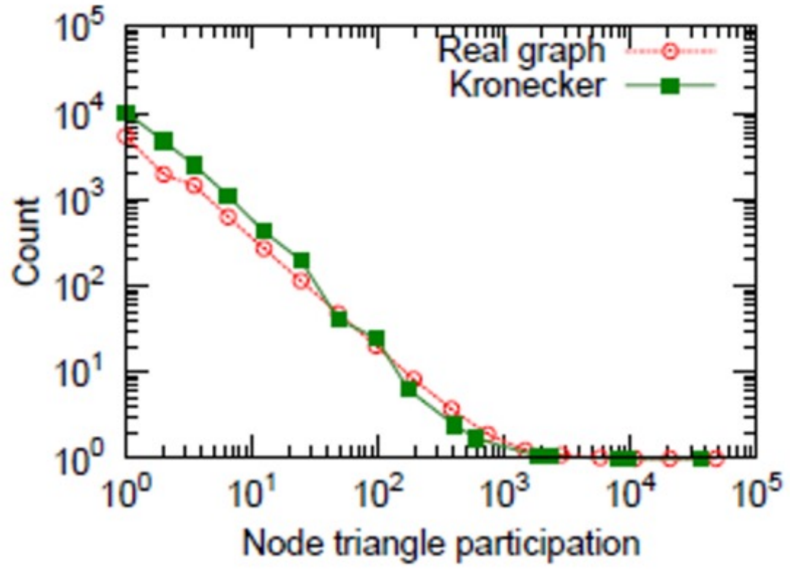
$$\Theta_1 = \begin{bmatrix} 0.99 & 0.54 \\ 0.49 & 0.13 \end{bmatrix}$$



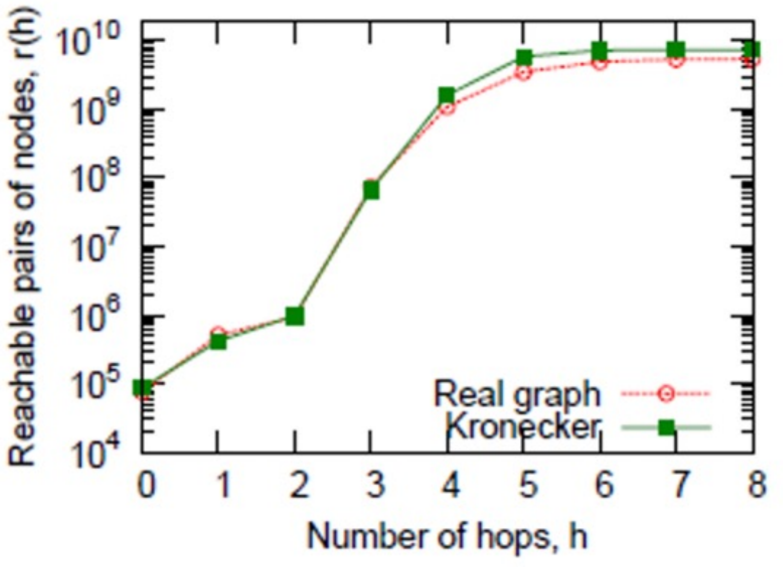

(a) In-Degree



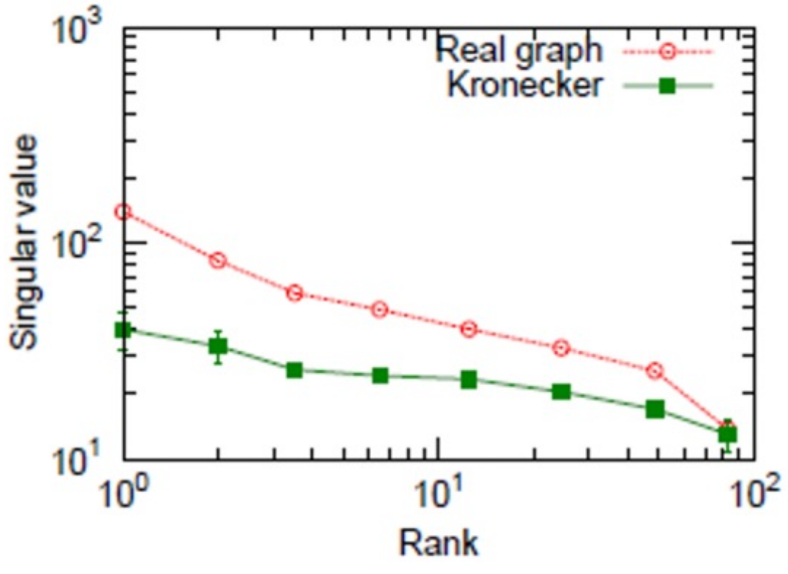

(b) Out-degree



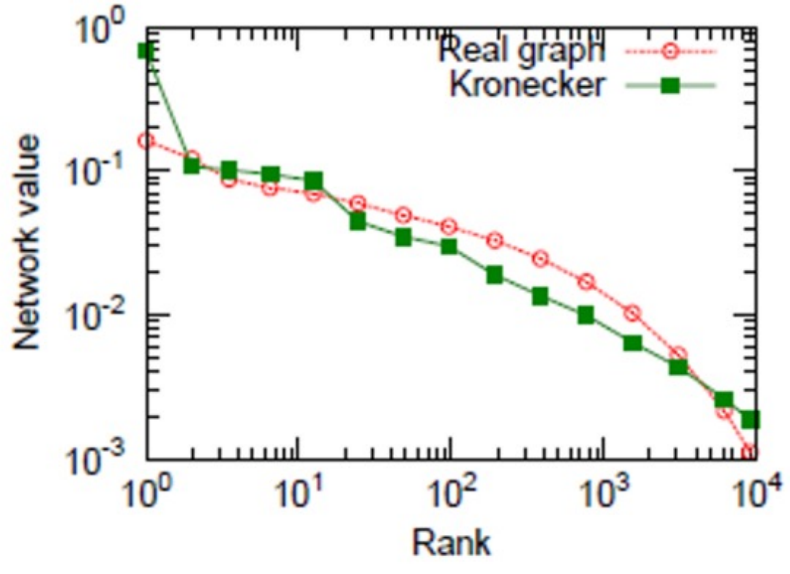

(c) Triangle participation




(d) Hop plot




(e) Scree plot




(f) “Network” value

# THANK YOU