그래프를 이용한 기계 학습

#2 실제 그래프는 어떻게 생겼을까?

신기정
(KAIST AI대학원)

1. 실제 그래프 vs 랜덤 그래프

2. 작은 세상 효과

3. 연결성의 두터운-꼬리 분포

4. 거대 연결 요소

5. 군집 구조

6. 실습: 군집 계수 및 지름 분석

1. 실제 그래프 vs 랜덤 그래프

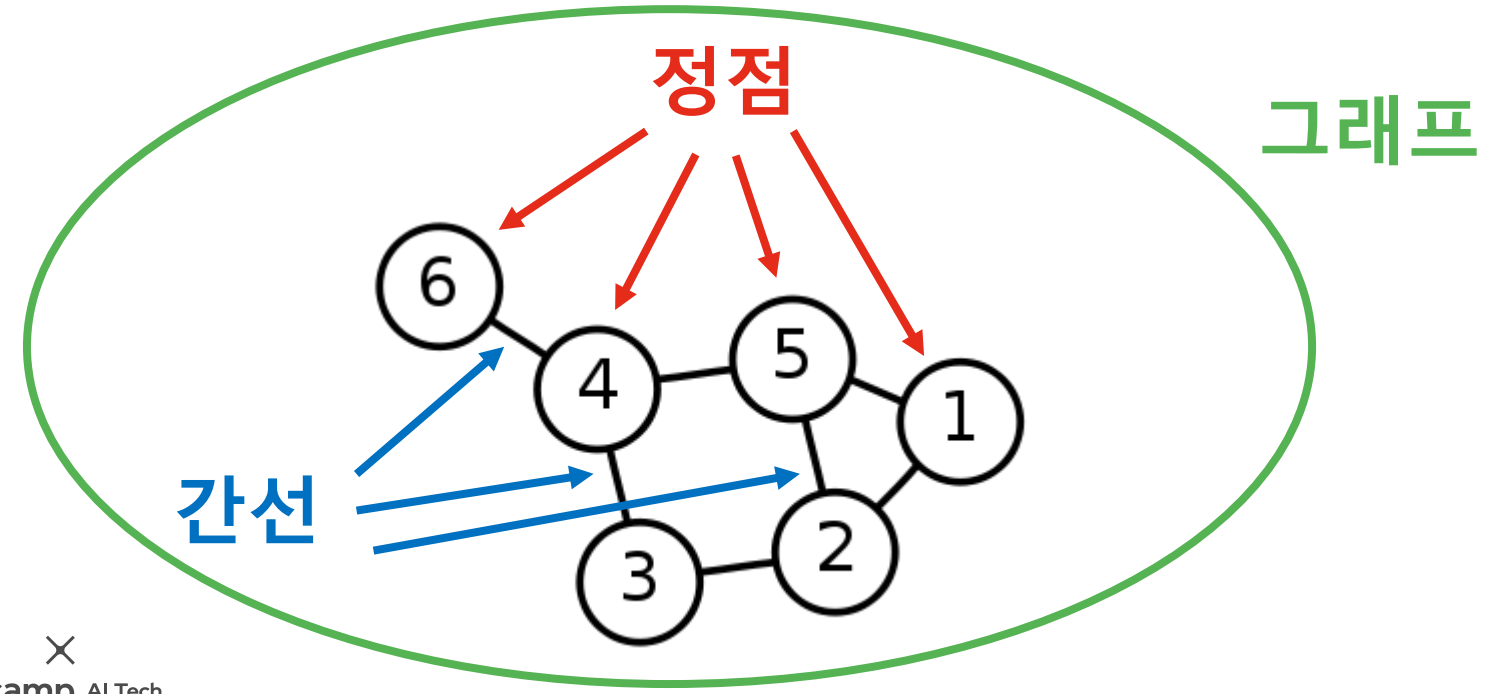
1.1 주요 개념 복습

1.2 실제 그래프 vs 랜덤 그래프

1.1 주요 개념 복습

그래프(Graph)는 정점 집합과 간선 집합으로 이루어진 수학적 구조입니다

보통 정점들의 집합을 $\boldsymbol{V}$, 간선들의 집합을 $\boldsymbol{E}$, 그래프를 $\boldsymbol{G = (V, E)}$로 적습니다



1.1 주요 개념 복습

정점의 이웃(Neighbor)은 그 정점과 연결된 다른 정점을 의미합니다

정점 v의 이웃들의 집합을 보통 $\boldsymbol{N(v)}$ 혹은$\boldsymbol{N_v}$로 적습니다

예시:

$N(1) = \{2,5\}$

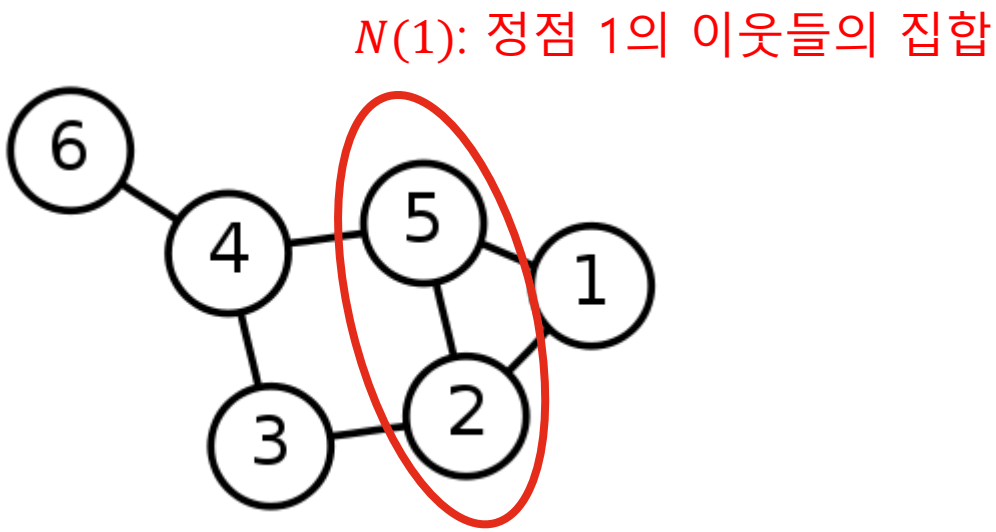
$N(2) = \{1,3,5\}$

$N(3) = \{2,4\}$

$N(4) = \{3,5,6\}$

$N(5) = \{1,2,4\}$

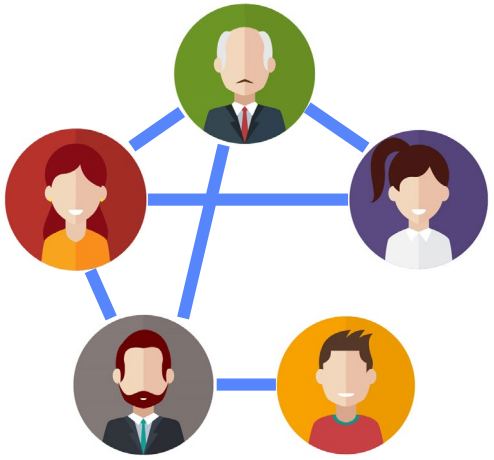
$N(6) = \{4\}$



1.2 실제 그래프 vs 랜덤 그래프

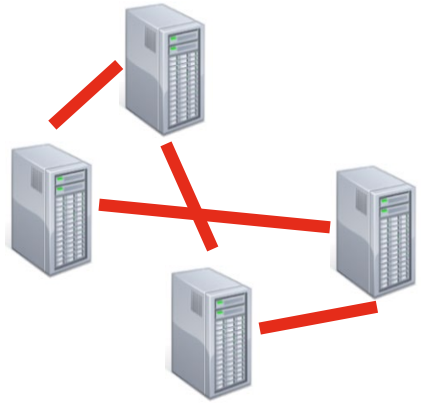
실제 그래프(Real Graph)란 다양한 복잡계로 부터 얻어진 그래프를 의미합니다

소셜 네트워크, 전자상거래 구매 내역, 인터넷, 웹, 뇌, 단백질 상호작용, 지식 그래프 등

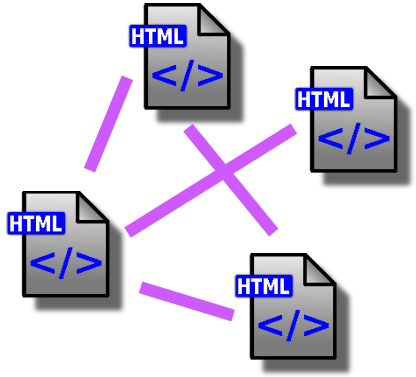


소셜 네트워크

전자상거래 구매 내역



인터넷



웹 그래프

1.2 실제 그래프 vs 랜덤 그래프

본 수업에서는 MSN 메신저 그래프를 실제 그래프의 예시로 사용합니다

MSN 메신저 그래프

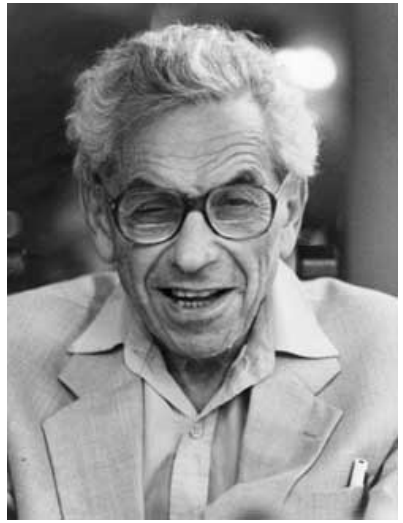
- 1억 8천만 정점 (사용자)
- 13억 간선 (메시지를 주고받은 관계)

MSN 메신저

1.2 실제 그래프 vs 랜덤 그래프

랜덤 그래프(Random Graph)는 확률적 과정을 통해 생성한 그래프를 의미합니다

본 수업에서는 에르되스(Erdős)와 레니(Rényi)가 제안한 랜덤 그래프 모델을 사용합니다



Paul Erdős

Alfréd Rényi

1.2 실제 그래프 vs 랜덤 그래프

에르되스-레니 랜덤 그래프(Erdős-Rényi Random Graph)

임의의 두 정점 사이에 간선이 존재하는지 여부는 동일한 확률 분포에 의해 결정됩니다

에르되스-레니 랜덤그래프 $G(n,p)$는

✓ n개의 정점을 가집니다

✓ 임의의 두 개의 정점 사이에 간선이 존재할 확률은 p입니다

✓ 정점 간의 연결은 서로 독립적(Independent)입니다

1.2 실제 그래프 vs 랜덤 그래프

에르되스-레니 랜덤그래프(Erdős-Rényi Random Graph)

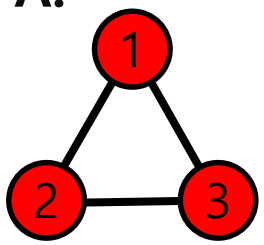
Q. $G(3, 0.3)$에 의해 생성될 수 있는 그래프와 각각의 확률은?

1.2 실제 그래프 vs 랜덤 그래프

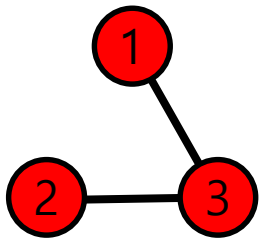
에르되스-레니 랜덤그래프(Erdős-Rényi Random Graph)

Q. $G(3, 0.3)$에 의해 생성될 수 있는 그래프와 각각의 확률은?

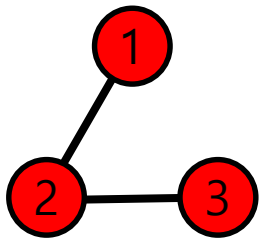
A.



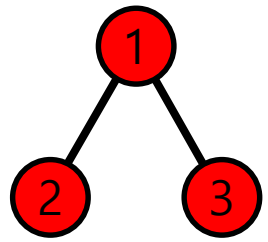
0.3^3



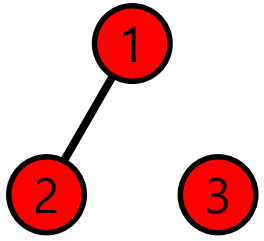
$0.3^2 \times 0.7$



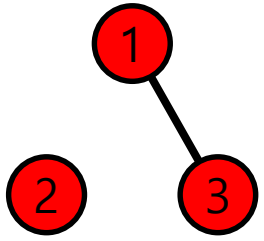
$0.3^2 \times 0.7$



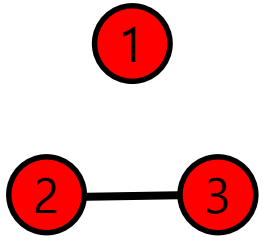
$0.3^2 \times 0.7$



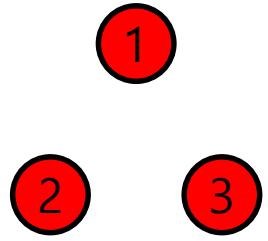
0.3×0.7^2



0.3×0.7^2



0.3×0.7^2



0.7^3

×

2. 작은 세상 효과

2.1 필수 개념: 경로, 거리 및 지름

2.2 작은 세상 효과

2.1 필수 개념: 경로, 거리 및 지름

정점 u와 v의 사이의 경로(Path)는 아래 조건을 만족하는 정점들의 순열(Sequence)입니다

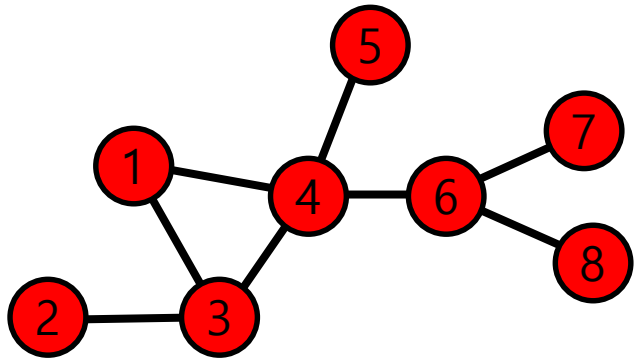
(1) u에서 시작해서 v에서 끝나야 합니다
(2) 순열에서 연속된 정점은 간선으로 연결되어 있어야 합니다

2.1 필수 개념: 경로, 거리 및 지름

정점 u와 v의 사이의 경로(Path)는 아래 조건을 만족하는 정점들의 순열(Sequence)입니다

(1) u에서 시작해서 v에서 끝나야 합니다
(2) 순열에서 연속된 정점은 간선으로 연결되어 있어야 합니다

정점 1과 8 사이의 경로 예시:
1, 4, 6, 8
1, 3, 4, 6, 8
1, 4, 3, 4, 6, 8

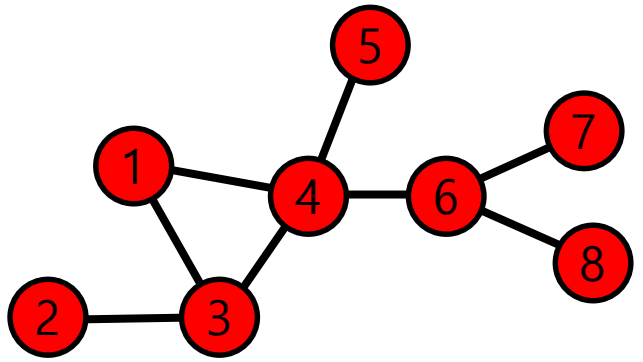


2.1 필수 개념: 경로, 거리 및 지름

두 정점 u와 v의 사이의 경로(Path)는 아래 조건을 만족하는 정점들의 순열(Sequence)입니다

(1) u에서 시작해서 v에서 끝나야 합니다
(2) 순열에서 연속된 정점은 간선으로 연결되어 있어야 합니다

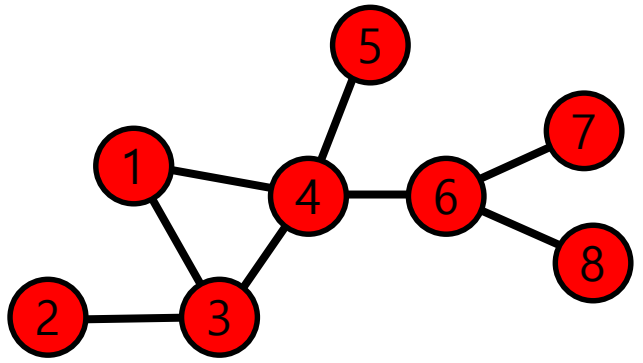
정점 1과 8 사이의 경로가 아닌 순열의 예시:
1, 6, 8
1, 3, 4, 5, 6, 8



2.1 필수 개념: 경로, 거리 및 지름

경로의 길이는 해당 경로 상에 놓이는 간선의 수로 정의됩니다

경로 1, 4, 6, 8의 길이는 3 입니다
경로 1, 3, 4, 6, 8의 길이는 4 입니다
경로 1, 4, 3, 4, 6, 8의 길이는 5 입니다

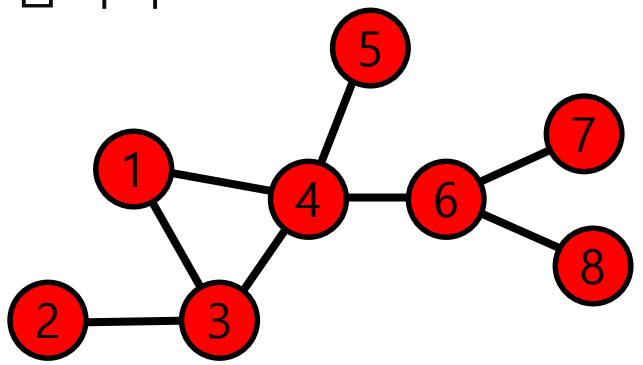


2.1 필수 개념: 경로, 거리 및 지름

정점 u와 v의 사이의 거리(Distance)는 u와 v 사이의 최단 경로의 길이입니다

예시:

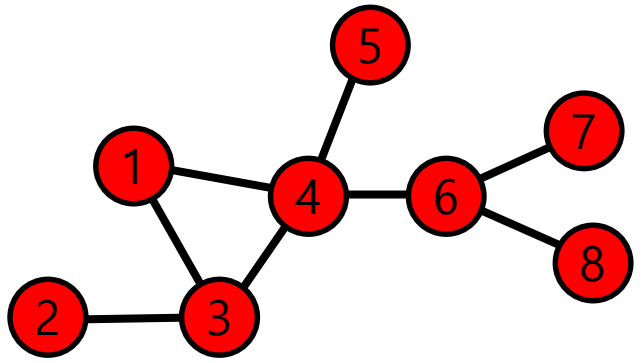
정점 1과 8 사이의 최단 경로(Shortest Path)는 1, 4, 6, 8 입니다
해당 경로의 길이는 3 입니다
따라서 정점 1과 8 사이의 거리는 3 입니다



2.1 필수 개념: 경로, 거리 및 지름

그래프의 지름(Diameter)은 정점 간 거리의 최댓값입니다

예시 그래프에서의 지름은 4 입니다
이는 정점 2와 8 사이의 최단 경로의 거리와 같습니다



2.2 작은 세상 효과

임의의 두 사람을 골랐을 때, 몇 단계의 지인을 거쳐 연결되어 있을까?

여섯 단계 분리(Six Degrees of Separataion) 실험

- 사회학자 스탠리 밀그램(Stanley Milgram)에 의해 1960년대에 수행된 실험입니다
- 오마하 (네브라스카 주)와 위치타 (켄사스 주)에서 500명의 사람을 뽑았습니다
- 그들에게 보스턴에 있는 한 사람에게 편지를 전달하게끔 하였습니다
- 단, 지인를 통해서만 전달하게끔 하였습니다

2.2 작은 세상 효과

임의의 두 사람을 골랐을 때, 몇 단계의 지인을 거쳐 연결되어 있을까?

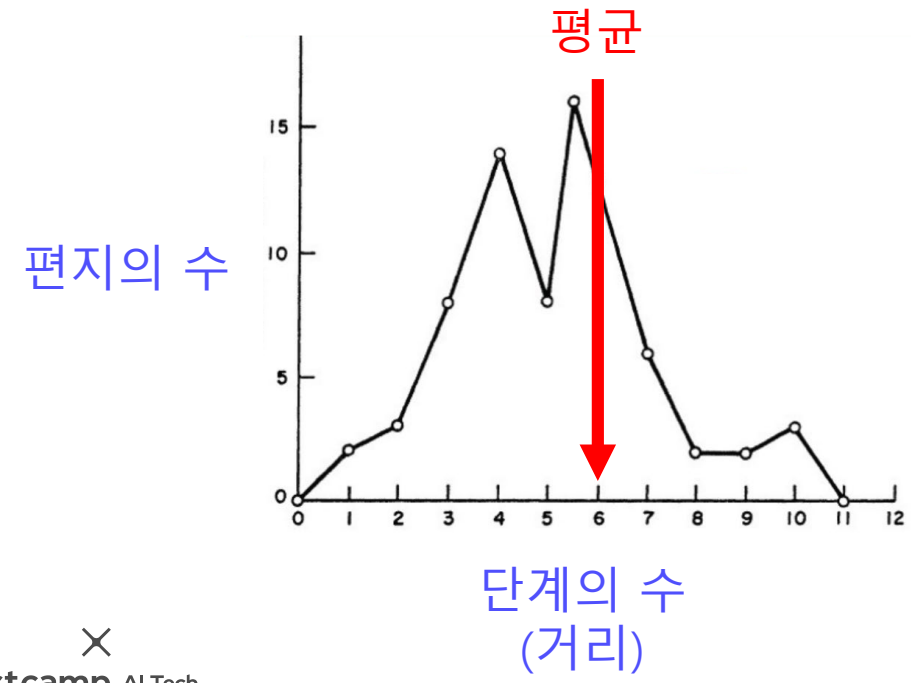
Q. 목적지에 도착하기까지 몇 단계의 지인을 거쳤을까요?

2.2 작은 세상 효과

임의의 두 사람을 골랐을 때, 몇 단계의 지인을 거쳐 연결되어 있을까?

Q. 목적지에 도착하기까지 몇 단계의 지인을 거쳤을까요?

A. 25%의 편지만 도착했지만, 평균적으로 6 단계만을 거쳤습니다



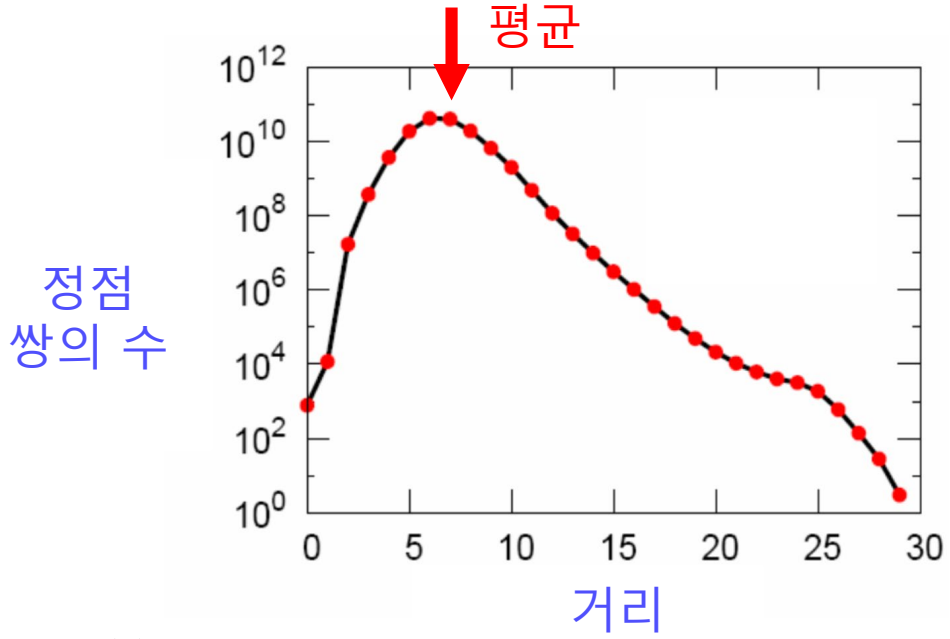
이미지 출처: https://web.stanford.edu/class/cs224w/slides/02-gnp-smallworld.pdf

2.2 작은 세상 효과

Q. MSN 메신저 그래프에서는 어떨까요?

A. 정점 간의 평균 거리는 7 정도 밖에 되지 않습니다

단, 거대 연결 구조만 고려했습니다. 거대 연결 구조는 뒤에서 설명합니다



이미지 출처: https://web.stanford.edu/class/cs224w/slides/02-gnp-smallworld.pdf

2.2 작은 세상 효과

이러한 현상을 작은 세상 효과(Small-world Effect)라고 부릅니다

한국에서는 "사돈의 팔촌"이 먼 관계를 나타내는 표현으로 사용됩니다
즉, 아무리 먼 관계도 결국은 사돈의 팔촌(10촌 관계)입니다

2.2 작은 세상 효과

작은 세상 효과는 높은 확률로 랜덤 그래프에도 존재합니다

모든 사람이 100명의 지인이 있다고 가정해봅시다
다섯 단계를 거치면 최대 100억($= 100^5$)명의 사람과 연결될 수 있습니다

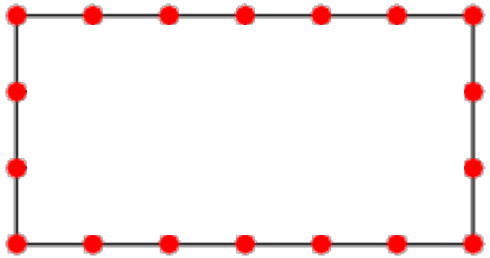
단, 실제로는 지인의 중복 때문에 100억 명보다는 적은 사람일 겁니다
하지만 여전히 많은 사람과 연결될 가능성이 높습니다

2.2 작은 세상 효과

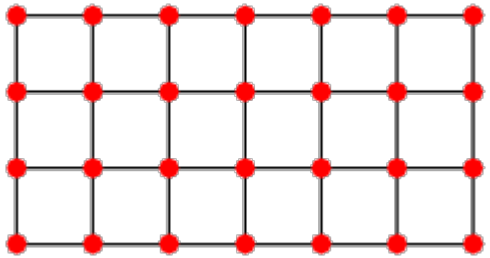
하지만 모든 그래프에서 작은 세상 효과가 존재하는 것은 아닙니다

체인(Chain), 사이클(Cycle), 격자(Grid) 그래프에서는 작은 세상 효과가 존재하지 않습니다

체인
(Chain Graph)



사이클 그래프
(Cycle Graph)



격자 그래프
(Grid Graph)

3. 연결성의 두터운 꼬리 분포

3.1 필수 개념: 연결성

3.2 연결성의 두터운 꼬리 분포

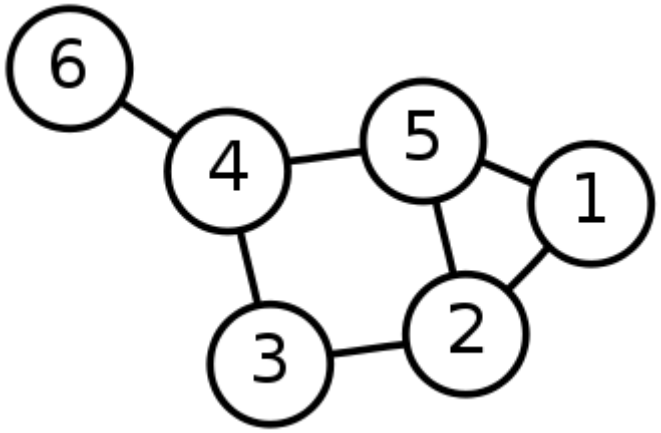
3.1 필수 개념: 연결성

정점의 연결성(Degree)은 그 정점과 연결된 간선의 수를 의미합니다

정점 v의 연결성은 해당 정점의 이웃들의 수와 같습니다
보통 정점 v의 연결성은 $\boldsymbol{d(v)}$, $\boldsymbol{d_v}$ 혹은 $\boldsymbol{|N(v)|}$ 로 적습니다

예시:

$d(1) = 2$
$d(2) = 3$
$d(3) = 2$
$d(4) = 3$
$d(5) = 3$
$d(6) = 1$



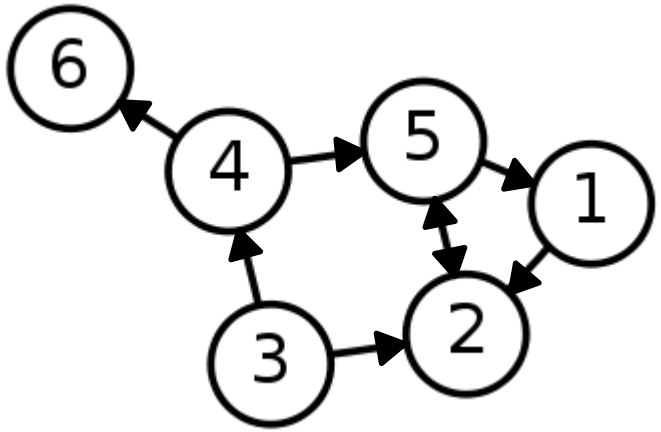
3.1 필수 개념: 연결성

정점의 나가는 연결성(Out Degree)은 그 정점에서 나가는 간선의 수를 의미합니다
보통 정점 v의 나가는 연결성은 $\boldsymbol{d_{out}(v)}$ 혹은 $\boldsymbol{|N_{out}(v)|}$으로 표시합니다

정점의 들어오는 연결성(In Degree)은 그 정점으로 들어오는 간선의 수를 의미합니다
보통 정점 v의 들어오는 연결성은 $\boldsymbol{d_{in}(v)}$ 혹은 $\boldsymbol{|N_{in}(v)|}$으로 표시합니다

예시:

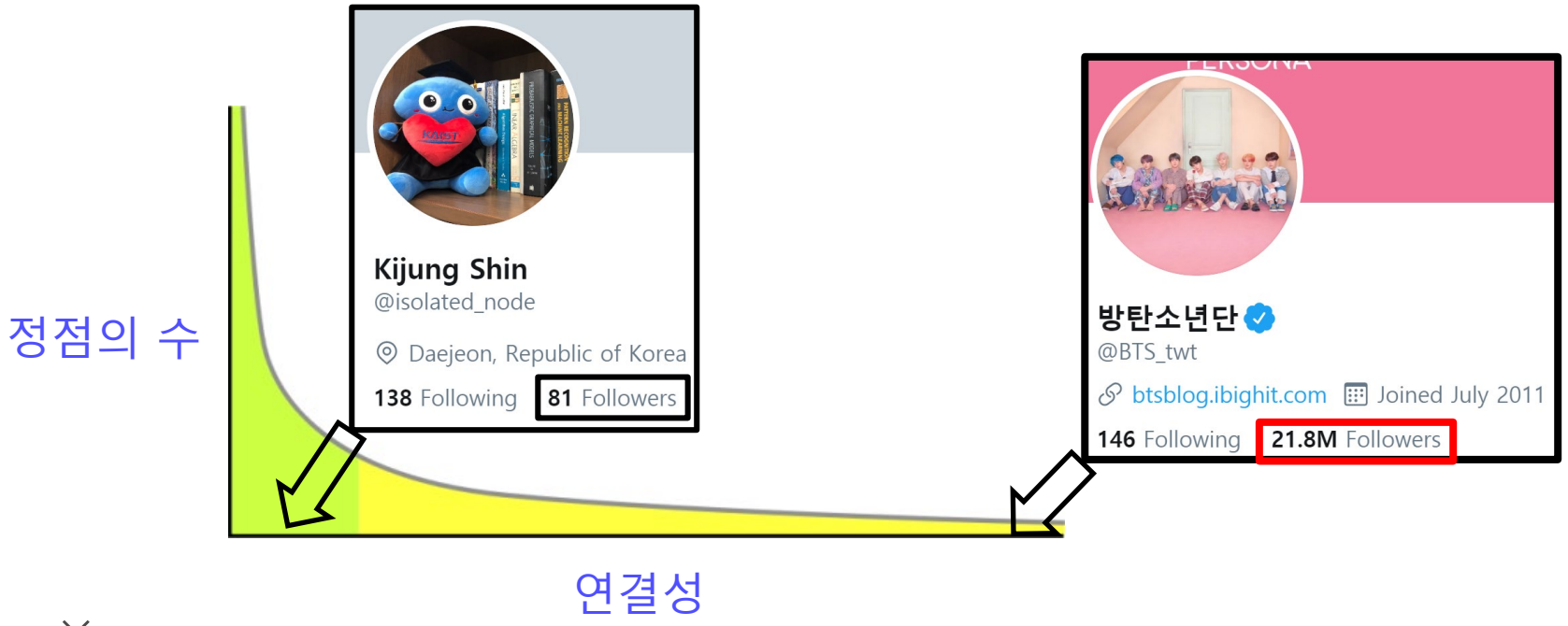
$d_{in}(1) = 1,\ d_{out}(1) = 1$
$d_{in}(2) = 3,\ d_{out}(1) = 1$
$d_{in}(3) = 0,\ d_{out}(3) = 2$
$d_{in}(4) = 1,\ d_{out}(4) = 2$
$d_{in}(5) = 2,\ d_{out}(5) = 2$
$d_{in}(6) = 4,\ d_{out}(6) = 0$



3.2 연결성의 두터운 꼬리 분포

실제 그래프의 연결성 분포는 두터운 꼬리(Heavy Tail)를 갖습니다

즉, 연결성이 매우 높은 허브(Hub) 정점이 존재함을 의미합니다

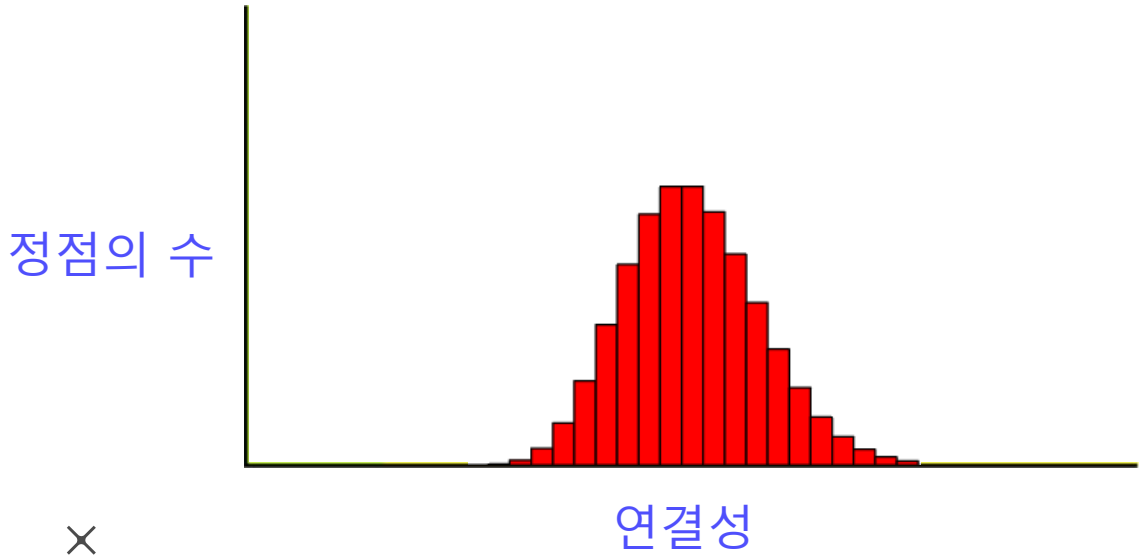


3.2 연결성의 두터운 꼬리 분포

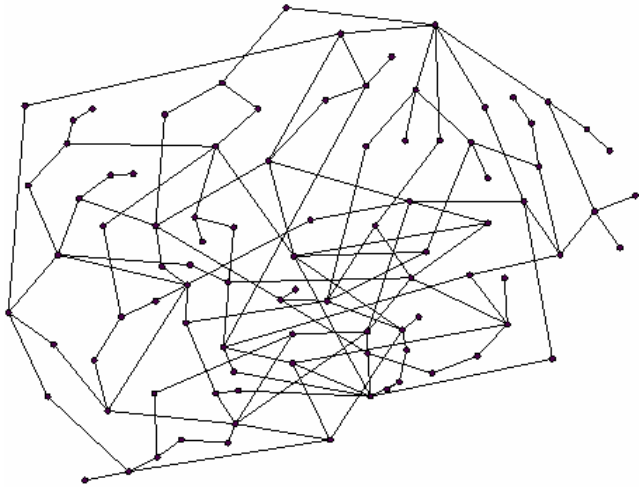
랜덤 그래프의 연결성 분포는 높은 확률로 정규 분포와 유사합니다

이 경우, 연결성이 매우 높은 허브(Hub) 정점이 존재할 가능성은 0에 가깝습니다

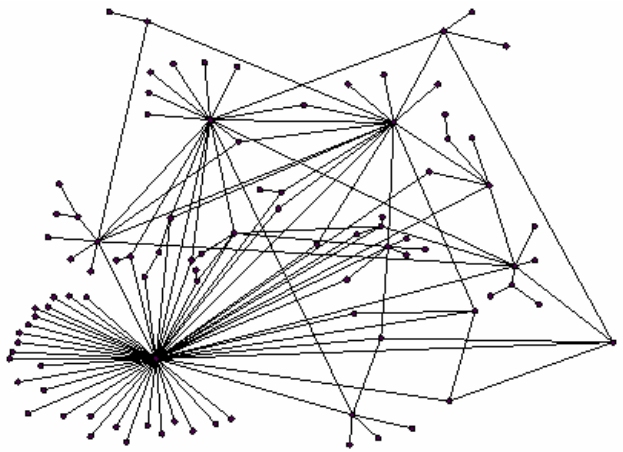
정규 분포와 유사한 예시로는 키의 분포가 있습니다
키가 10 미터를 넘는 극단적인 예외는 존재하지 않습니다



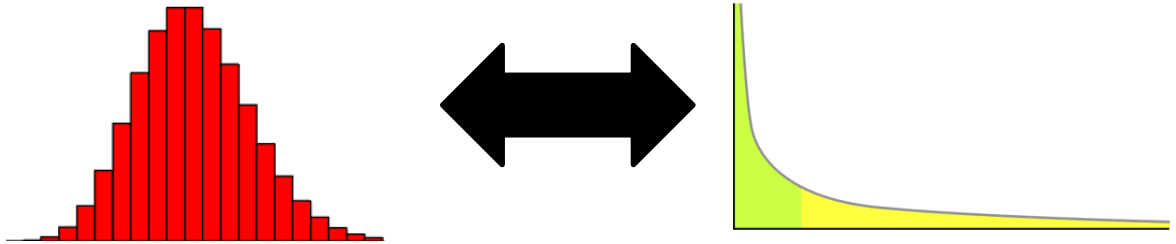
3.2 연결성의 두터운 꼬리 분포



랜덤 그래프에서의 연결성



실제 그래프에서의 연결성



이미지 출처: https://web.stanford.edu/class/cs224w/slides/02-gnp-smallworld.pdf

4. 거대 연결 요소

4.1 필수 개념: 연결 요소

4.2 거대 연결 요소

4.1 필수 개념: 연결 요소

연결 요소(Connected Component)는 다음 조건들을 만족하는 정점들의 집합을 의미합니다

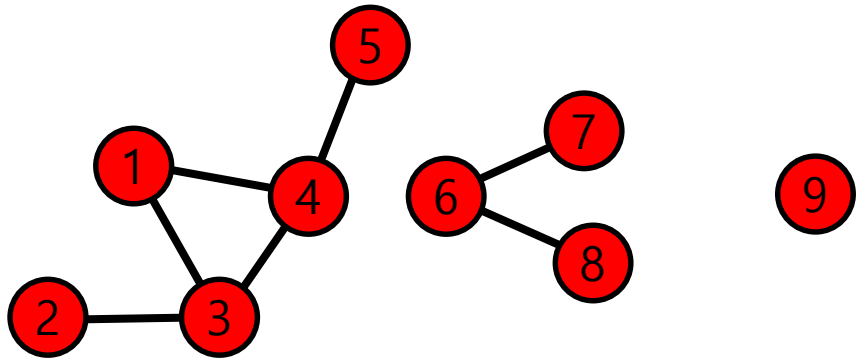
(1) 연결 요소에 속하는 정점들은 경로로 연결될 수 있습니다
(2) (1)의 조건을 만족하면서 정점을 추가할 수 없습니다

4.1 필수 개념: 연결 요소

연결 요소(Connected Component)는 다음 조건들을 만족하는 정점들의 집합을 의미합니다

(1) 연결 요소에 속하는 정점들은 경로로 연결될 수 있습니다
(2) (1)의 조건을 만족하면서 정점을 추가할 수 없습니다

예시 그래프에는 3개의 연결 요소가 존재합니다
{1,2,3,4,5}, {6,7,8}, {9}



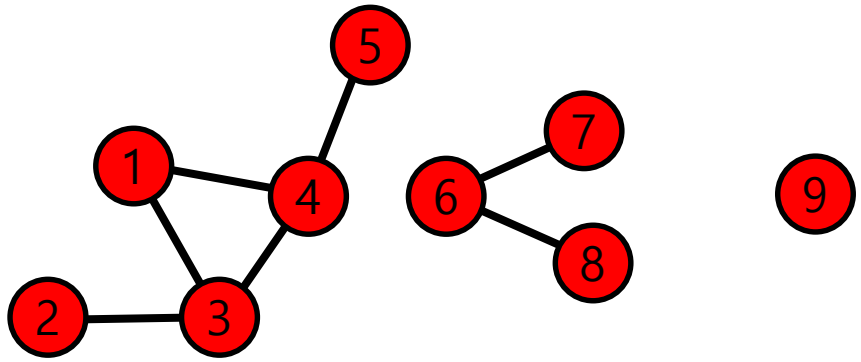
4.1 필수 개념: 연결 요소

연결 요소(Connected Component)는 다음 조건들을 만족하는 정점들의 집합을 의미합니다

(1) 연결 요소에 속하는 정점들은 경로로 연결될 수 있습니다
(2) (1)의 조건을 만족하면서 정점을 추가할 수 없습니다

예시 그래프에는 3개의 연결 요소가 존재합니다
{1,2,3,4,5}, {6,7,8}, {9}

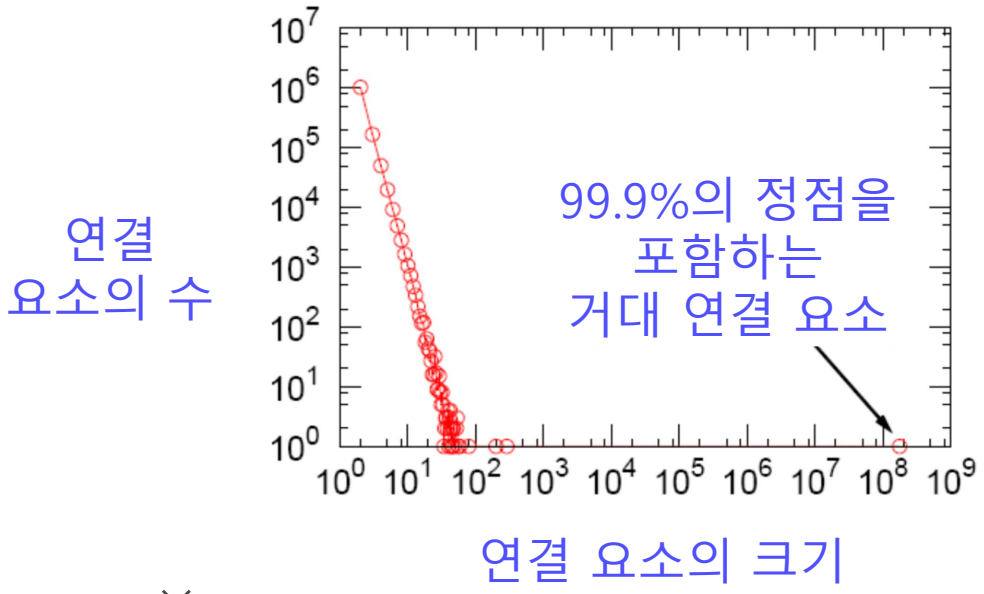
{1,2,3,4}는 조건 (2)를 위배합니다
{6,7,8,9}는 조건 (1)을 위배합니다



4.2 거대 연결 요소

실제 그래프에는 거대 연결 요소(Giant Connected Component)가 존재합니다
거대 연결 요소는 대다수의 정점을 포함합니다

MSN 메신저 그래프에는 99.9%의 정점이 하나의 거대 연결 요소에 포함됩니다



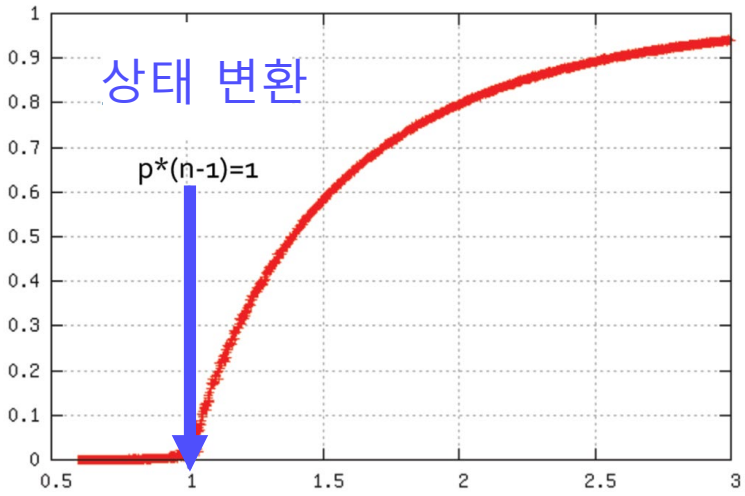
이미지 출처: https://web.stanford.edu/class/cs224w/slides/02-gnp-smallworld.pdf

4.2 거대 연결 요소

랜덤 그래프에도 높은 확률로 거대 연결 요소(Giant Connected Component)가 존재합니다

단, 정점들의 평균 연결성이 1보다 충분히 커야 합니다
자세한 이유는 Random Graph Theory를 참고하시기 바랍니다

가장 큰 연결 요소에 속하는 정점의 비율



평균 연결성

이미지 출처: https://web.stanford.edu/class/cs224w/slides/02-gnp-smallworld.pdf

5. 군집 구조

5.1 필수 개념: 군집 구조 및 군집 계수

5.2 높은 군집 계수

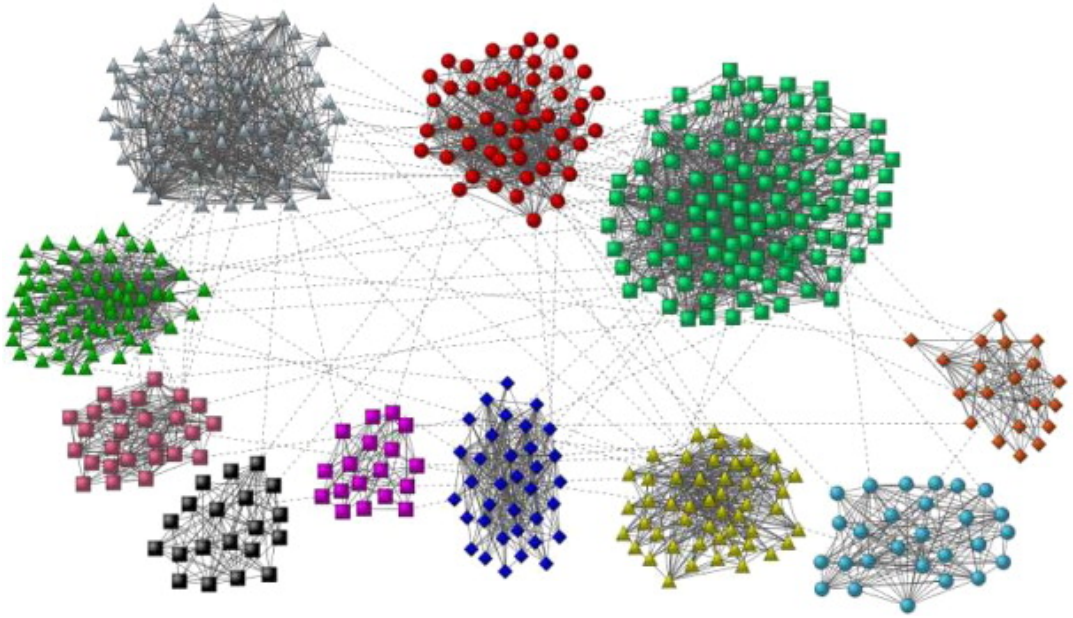
5.1 필수 개념: 군집

군집(Community)이란 다음 조건들을 만족하는 정점들의 집합입니다

(1) 집합에 속하는 정점 사이에는 많은 간선이 존재합니다
(2) 집합에 속하는 정점과 그렇지 않은 정점 사이에는 적은 수의 간선이 존재합니다

수학적으로 엄밀한 정의는 아닙니다

예시 그래프에는 11 개의 군집이
있는 것으로 보입니다



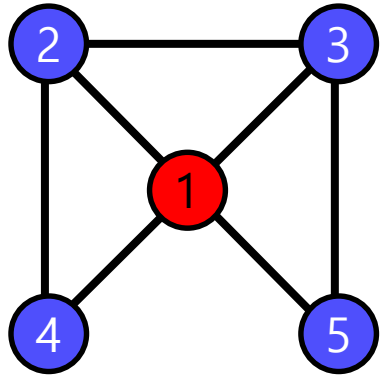
5.1 필수 개념: 지역적 군집 계수

지역적 군집 계수(Local Clustering Coefficient)는 한 정점에서 군집의 형성 정도를 측정합니다

정점 i의 지역적 군집 계수는 정점 i의 이웃 쌍 중 간선으로 직접 연결된 것의 비율을 의미합니다
정점 i의 지역적 군집 계수를 C_i로 표현합시다

예시 그래프를 살펴봅시다
정점 1의 이웃은 4개이며, 총 6개의 이웃 쌍이 존재합니다
그 중 3개의 쌍이 간선으로 직접 연결 되어 있습니다

따라서, $C_1 = \frac{3}{6} = 0.5$

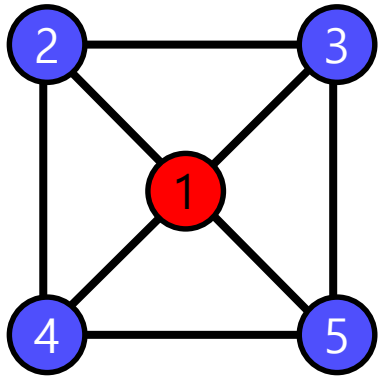


5.1 필수 개념: 지역적 군집 계수

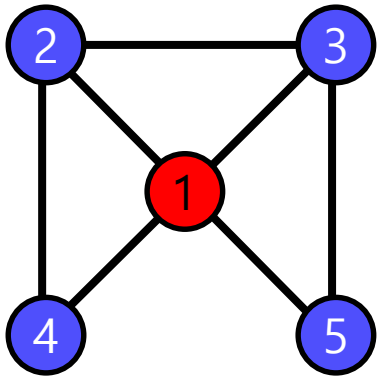
지역적 군집 계수(Local Clustering Coefficient)는 한 정점에서 군집의 형성 정도를 측정합니다

정점 i의 지역적 군집 계수는 정점 i의 이웃 쌍 중 간선으로 직접 연결된 것의 비율을 의미합니다
정점 i의 지역적 군집 계수를 C_i로 표현합시다

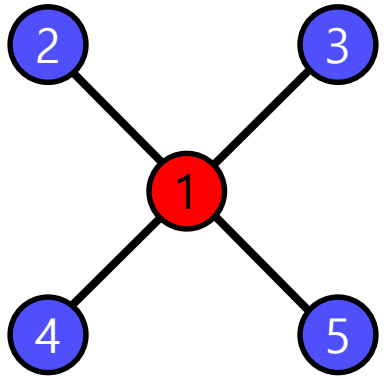
더 많은 예시를 살펴봅시다



$C_1 = 0.66$



$C_1 = 0.5$



$C_1 = 0$

5.1 필수 개념: 지역적 군집 계수

지역적 군집 계수(Local Clustering Coefficient)는 한 정점에서 군집의 형성 정도를 측정합니다

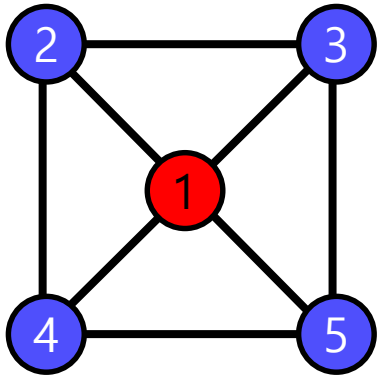
정점 i의 지역적 군집 계수는 정점 i의 이웃 쌍 중 간선으로 연결된 것의 비율을 의미합니다
정점 i의 지역적 군집 계수를 C_i로 표현합시다

참고로 연결성이 0인 정점에서는 지역적 군집 계수가 정의되지 않습니다

5.1 필수 개념: 지역적 군집 계수

잠깐, 지역적 군집 계수가 군집이랑 어떻게 연결되는 것이죠?

정점 i의 지역적 군집 계수가 매우 높다고 합시다
즉, 정점 i의 이웃들도 높은 확률로 서로 간선으로 연결되어 있습니다
정점 i와 그 이웃들은 높은 확률로 군집을 형성합니다



$$C_1 = 0.66$$

×

5.1 필수 개념: 전역 군집 계수

전역 군집 계수(Global Clustering Coefficient)는 전체 그래프에서 군집의 형성 정도를 측정합니다

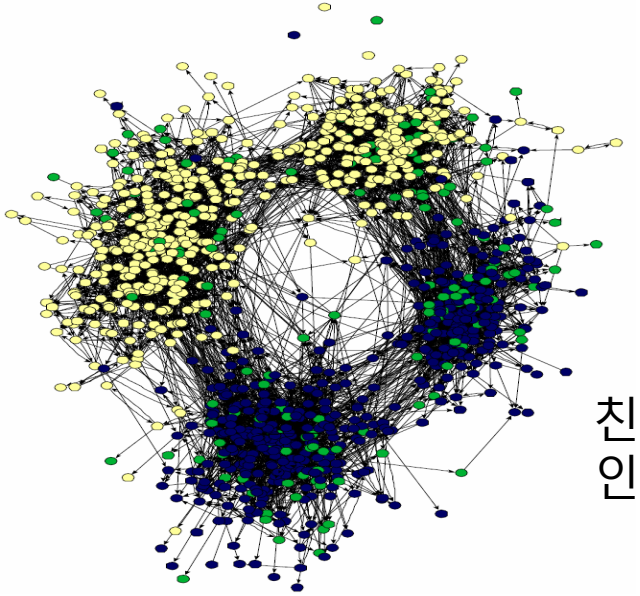
그래프 G의 전역 군집 계수는 각 정점에서의 지역적 군집 계수의 평균입니다
단, 지역적 군집 계수가 정의되지 않는 정점은 제외합니다

5.2 높은 군집 계수

실제 그래프에서는 군집 계수가 높습니다. 즉 많은 군집이 존재합니다

여러가지 이유가 있을 수 있습니다

동질성(Homophily): 서로 유사한 정점끼리 간선으로 연결될 가능성이 높습니다
같은 동네에 사는 같은 나이의 아이들이 친구가 되는 경우가 그 예시입니다



친구 관계 그래프로 정점의 색은
인종을 의미합니다

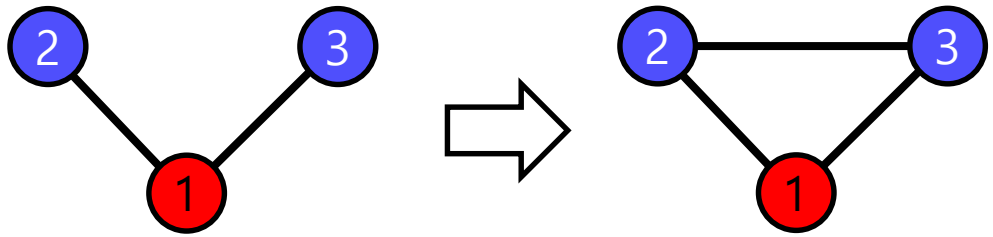
5.2 높은 군집 계수

실제 그래프에서는 군집 계수가 높습니다. 즉 많은 군집이 존재합니다

여러가지 이유가 있을 수 있습니다

동질성(Homophily): 서로 유사한 정점끼리 간선으로 연결될 가능성이 높습니다
같은 동네에 사는 같은 나이의 아이들이 친구가 되는 경우가 그 예시입니다

전이성(Transitivity): 공통 이웃이 있는 경우, 공통 이웃이 매개 역할을 해줄 수 있습니다
친구를 서로에게 소개해주는 경우가 그 예시입니다



5.2 높은 군집 계수

반면 랜덤 그래프에서는 지역적 혹은 전역 군집 계수가 높지 않습니다

구체적으로 랜덤 그래프 $G(n,p)$에서의 군집 계수는 p입니다
랜덤 그래프에서의 간선 연결이 독립적인 것을 고려하면 당연한 결과입니다
즉 공통 이웃의 존재 여부가 간선 연결 확률에 영향을 미치지 않습니다

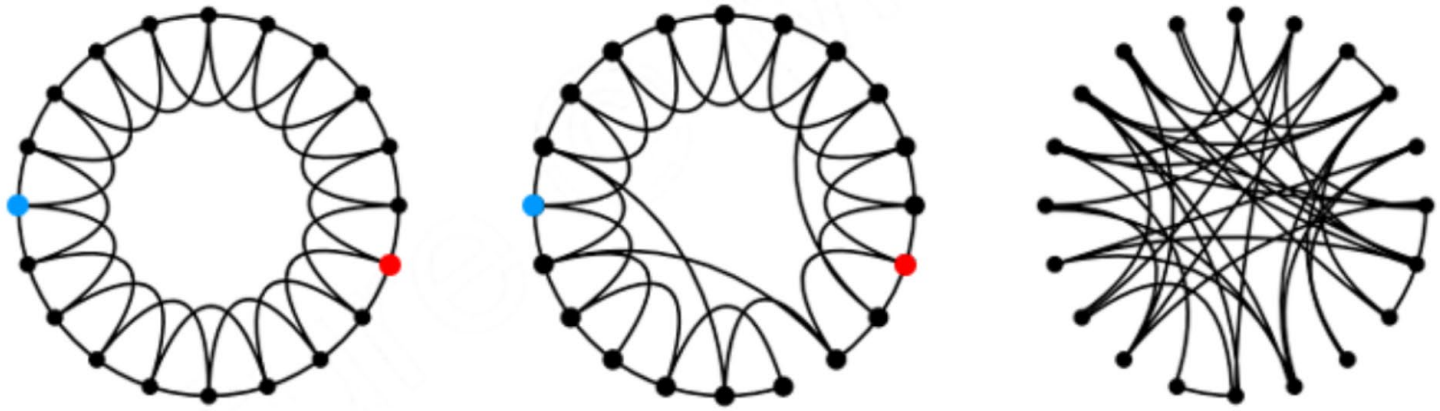
6. 실습: 군집 계수 및 지름 분석

6.1 데이터 불러오기
6.2 군집 계수 계산
6.3 지름 계산
6.4 비교

6.1 그래프 불러오기

이번 실습에서는 다음 세 종류의 그래프의 구조를 분석합니다

작은 세상 그래프는 균일 그래프의 일부 간선을 임의로 선택한 간선으로 대체한 그래프입니다



균일 그래프
(Regular Graph)

작은 세상 그래프
(Small-world Graph)

랜덤 그래프
(Random Graph)

6.1 그래프 불러오기

파일에 저장된 균일 그래프를 읽어옵니다

```
regular_graph = nx.Graph()
data = osp.abspath(osp.join(os.getcwd(), 'drive/MyDrive/data/simple/regular.txt'))
f = open(data)
for line in f:
    v1, v2 = map(int, line.split())
    regular_graph.add_edge(v1, v2)
```

6.1 그래프 불러오기

파일에 저장된 작은 세상 그래프와 랜덤 그래프도 읽어서 불러옵니다

```
small_world_graph = nx.Graph()
random_graph = nx.Graph()

data = osp.abspath(osp.join(os.getcwd(), 'drive/MyDrive/data/simple/small_world.txt'))
f = open(data)
for line in f:
    v1, v2, = map(int, line.split())
    small_world_graph.add_edge(v1, v2)

data = osp.abspath(osp.join(os.getcwd(), 'drive/MyDrive/data/simple/random.txt'))
f = open(data)
for line in f:
    v1, v2 = map(int, line.split())
    random_graph.add_edge(v1, v2)
```

6.2 군집 계수 계산

주어진 그래프의 전역 군집 계수를 계산하는 함수를 정의합니다

```
def getGraphAverageClusteringCoefficient(Graph):
    ccs = []
    for v in Graph.nodes:
        num_connected_pairs = 0
        for neighbor1 in Graph.neighbors(v):
            for neighbor2 in Graph.neighbors(v):
                if neighbor1 <= neighbor2:
                    continue
                if Graph.has_edge(neighbor1, neighbor2)
                    num_connected_pairs = num_connected_pairs + 1
        cc = num_connected_pairs / (Graph.degree(v) * (Graph.degree(v) -1) / 2)
        ccs.append(cc)
    return sum(ccs) / len(ccs)
```

6.3 지름 계산

주어진 그래프의 지름을 계산하는 함수를 정의합니다

```
def getGraphDiameter(Graph):
    diameter = 0
    for v in Graph.nodes:
        length = nx.single_source_shortest_path_length(Graph, v)
        max_length = max(length.values())
        if max_length > diameter:
            diameter = max_length
    return diameter
```

6.4 비교 분석

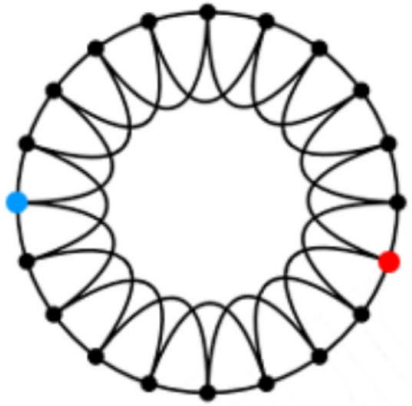
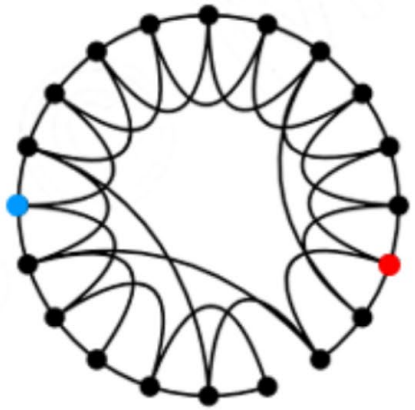
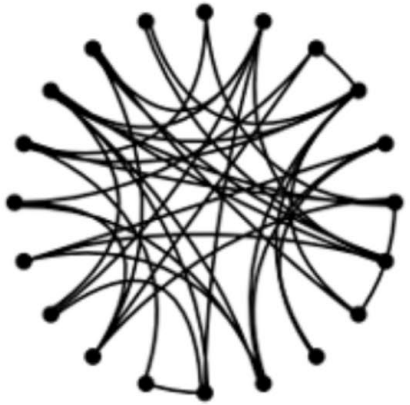
정의한 함수를 이용하여, 세 가지의 그래프를 비교합니다

```
print("1. Graph Diameter")
print("regular graph : " + str(getGraphDiameter(regular_graph)))
print("small world graph : " + str(getGraphDiameter(small_world_graph)))
print("random graph : " + str(getGraphDiameter(random_graph)) + "\n")

print("2. Average Clustering Coefficient")
print("regular graph : " + str(getGraphAverageClusteringCoefficient(regular_graph)))
print("small world graph : " + str(getGraphAverageClusteringCoefficient(small_world_graph)))
print("random graph : " + str(getGraphAverageClusteringCoefficient(random_graph)) + "\n")
```

6.4 비교 분석

비교 분석 결과를 정리하면 다음과 같습니다

	 균일 그래프 (Regular Graph)	 작은 세상 그래프 (Small-world Graph)	 랜덤 그래프 (Random Graph)
군집 계수	크다	크다	작다
지름	크다	작다	크다

2강 정리

1. **실제 그래프 vs 랜덤 그래프:** 실제 그래프는 복잡계로부터 얻어지는 반면, 랜덤 그래프는 확률적 과정을 통해 생성합니다

2. **작은 세상 효과:** 실제 그래프의 정점들은 가깝게 연결되어 있습니다

3. **연결성의 두터운 꼬리 분포:** 실제 그래프에는 연결성이 매우 높은 허브 정점이 존재합니다

4. **거대 연결 요소:** 실제 그래프에는 대부분의 정점을 포함하는 거대 연결 요소가 존재합니다

5. **군집 구조:** 실제 그래프에는 군집이 존재하며, 실제 그래프는 군집 계수가 높습니다

6. **(실습) 파이썬을 이용한 지름 및 군집 계수 분석**