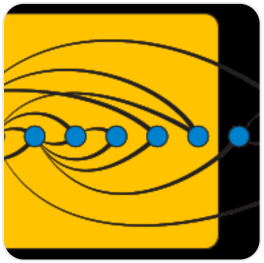
Applications: Triangles are key in network analysis. In social networks, the *triadic closure* principle says two individuals with a common friend are likely to become friends, so social graphs tend to have many triangles. The clustering coefficient of a network is based on triangle counts en.wikipedia.org . In computational biology, triangles appear as simple network motifs indicating feedback loops. Triangles are also used in graph mining (finding frequent subgraphs) and in graph embedding/clustering: communities often have high triangle density. Fast triangle-counting algorithms are used in performance-sensitive areas like large-scale social network analysis and subgraph mining.

Degeneracy in a Graph



The **degeneracy** of a graph is a measure of its sparsity. A graph is called *k*-degenerate if every subgraph has at least one vertex of degree ≤ *k*. Equivalently, one can iteratively remove the lowest-degree vertex: if the highest degree encountered during this removal process is *k*, the graph is *k*-degenerate en.wikipedia.org en.wikipedia.org . For example, the figure shows a 2-degenerate graph (each vertex has at most two neighbors to its left) and highlights its 2-core (the shaded subgraph that remains after repeatedly deleting vertices of degree <2) en.wikipedia.org .

Formally, the **degeneracy** of a graph is the smallest *k* for which it is *k*-degenerate en.wikipedia.org . It is also equal to the maximum core number in the graph: if one repeatedly removes all vertices of degree less than *k*, the remaining components are the *k*-cores, and the degeneracy is the largest *k* such that a non-empty *k*-core exists en.wikipedia.org . Degeneracy is closely related to other sparsity measures like arboricity. It satisfies facts such as: every planar graph is 5-degenerate en.wikipedia.org (by the Five Color Theorem), every outerplanar graph is 2-degenerate en.wikipedia.org , and many real-world networks (social, web, biological) have small degeneracy compared to their size. Additionally, the degeneracy of a graph plus one gives an upper bound on its chromatic number (a greedy coloring on a degeneracy ordering uses at most *k*+1 colors).

Computing Degeneracy (Core Decomposition): A classic linear-time algorithm (Matula–Beck) finds a *degeneracy ordering* of the vertices. Conceptually, repeat the following until all vertices are removed: pick a vertex *v* of minimum degree, record *d_v*, remove *v* and its edges, and update degrees of neighbors. Track the maximum degree removed; this maximum is the degeneracy *k* en.wikipedia.org en.wikipedia.org . A bucket queue or min-heap can perform these removals in $O(n+m)$ time (where m = number of edges) en.wikipedia.org . Pseudocode for this process:

vbnet Copy

```
initialize all vertices in a bucket structure keyed by their degree
degeneracy = 0
for i from 1 to n:
    let u = a vertex of minimum degree (remove it from bucket)
    degeneracy = max(degeneracy, degree[u])
    output u in the degeneracy ordering
    for each neighbor w of u (not yet removed):
        decrement degree[w] by 1 (and update bucket)
return degeneracy
```

After this algorithm, the list of removed vertices in reverse order is a *core decomposition*: vertices removed in rounds 1 remain in the 0-core, those surviving until degree ≥1 removal form the 1-core, and so on. In code, one might implement a simple version using a heap or even a sorted structure:

python Copy

```python
# Example Python: compute degeneracy of an undirected graph G (adj list)
def compute_degeneracy(G):
    import heapq
    degree = {u: len(G[u]) for u in G}
    heap = [(deg, u) for u, deg in degree.items()]
    heapq.heapify(heap)
    removed = set()
    degeneracy = 0
    while heap:
        deg, u = heapq.heappop(heap)
        if u in removed:
            continue
        degeneracy = max(degeneracy, deg)
        removed.add(u)
        for w in G[u]:
            if w not in removed:
```

```
                degree[w] -= 1
                heapq.heappush(heap, (degree[w], w))
    return degeneracy
```

This returns the graph's degeneracy in worst-case $O((n+m)\log n)$ time. In practice, specialized bucket queues achieve true $O(n+m)$ performance en.wikipedia.org .

Applications: Degeneracy and core decomposition have many applications. In social networks, high k-cores often correspond to cohesive subcommunities or influential users: peeling off low-degree "peripheral" nodes reveals a core of well-connected individuals. In biology, k-core analysis of protein interaction or gene-regulatory networks can identify central modules or functional groups. Degeneracy ordering is also used as a heuristic in algorithms for maximum clique or graph coloring, since any clique has size at most $k+1$. More broadly, knowing a graph's small degeneracy ($k \ll n$) allows faster algorithms (for example, triangle listing in $O(m\,k)$ time via Chiba–Nishizeki's method, or faster subgraph enumeration).

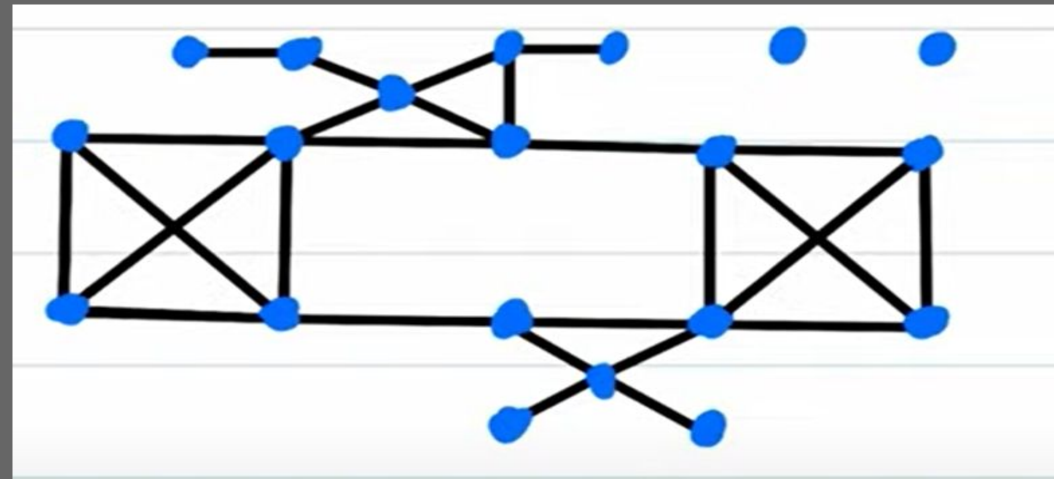
Relevant facts include: every planar graph is at most 5-degenerate en.wikipedia.org ; trees and forests are 1-degenerate (since they always have a leaf); and an m-edge graph has degeneracy at most $\sqrt{2m}$ (since average degree is $2m/n$, some subgraph has low-degree). Computing degeneracy (via core decomposition) is thus a practical way to analyze the "densest" parts of a network while maintaining efficient runtimes.

References: Standard graph theory sources and algorithms texts discuss these concepts. For definitions and linear-time algorithms see en.wikipedia.org en.wikipedia.org ; for applications in network analysis see recent surveys such as "core decomposition" in large-scale networks en.wikipedia.org en.wikipedia.org .

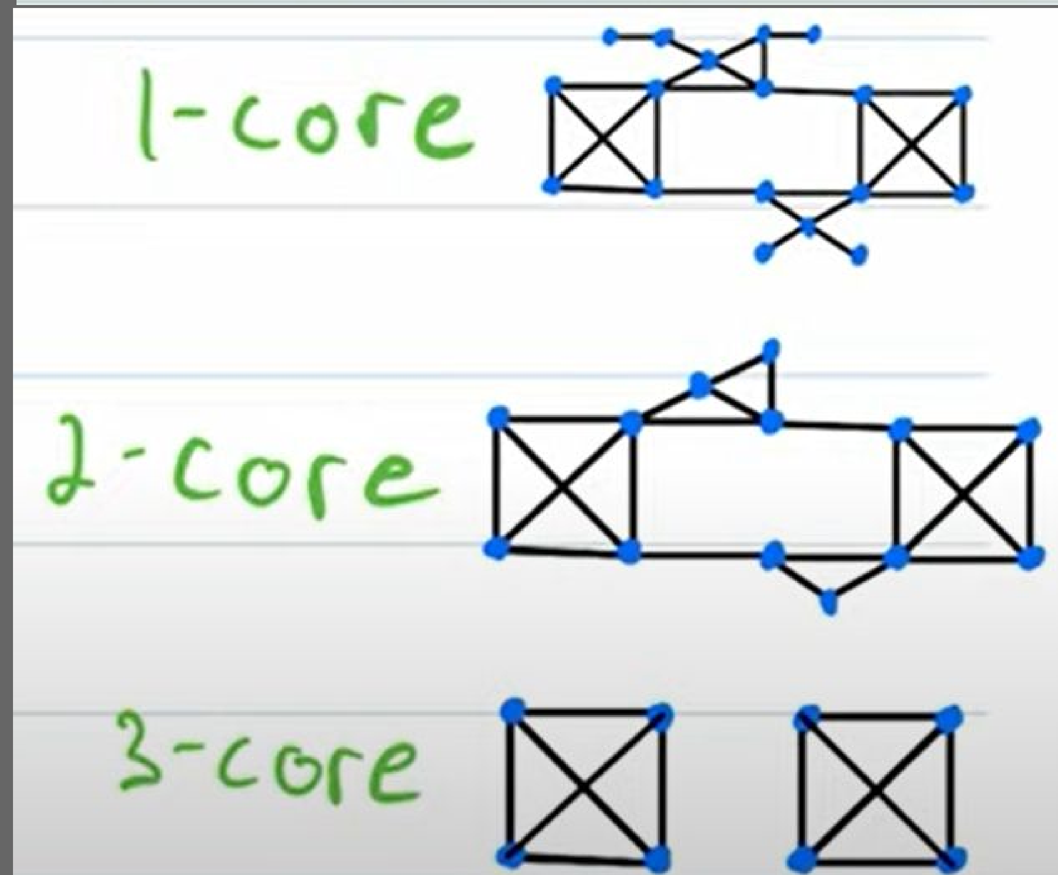
Citations

W **Triangle graph - Wikipedia**

the K-Core of a graph G is the maximal subgraph H (subset G) that minimum degree of H, or Delta(H) >= K.



for example this graph can be decomposed into different K-Cors, But the 4-Core is Null Graph.



Formally, the degeneracy of a graph is the smallest k for which it is k-degenerate .

It is also equal to the maximum core number in the graph: if one repeatedly removes all vertices of degree less than k, the remaining components are the k-cores.

the degeneracy is the largest k such that a non-empty k-core exists. which aligns with the other definition.

The degeneracy of a graph is a measure of its sparsity. A graph is called k-degenerate if every subgraph has at least one vertex of degree $\leq$ k

for example this graph is 3-degenerate !!