Cheat Sheet: Degeneracy for Triangle Counting in Graph Streams

1. Context & Motivation

- **Why triangles?**
  - Fundamental in network science (clustering, anomaly detection) and database join-size estimation.
  - Streaming setting: edges arrive one by one, memory very limited.
- **Prior space bounds:** Worst-case for arbitrary-order, constant-pass streams is

$$\Theta\big(\min\{m^{3/2}/T,\ m/\sqrt{T}\}\big) \quad \text{(McGregor et al. 2016; Bera et al. 2017)} : contentReference[oaicite:0]index =$$

2. Problem Statement

- **Input:** stream of m edges over n vertices.
- **Goal:** $(1 \pm \varepsilon)$-approximation $\hat{T}$ to true triangle count T.
- **Constraints:** constant passes, sublinear memory in m.

3. Graph Degeneracy & Core Number

- $\kappa(G) = \max_{G' \subseteq G} \big\{\min\text{-degree}(G')\big\}$: largest minimum degree over all subgraphs :contentReference[oaicite:1]index=1.
- Equivalently, vertices can be ordered so each has $\leq \kappa$ later-neighbors.
- Real-world graphs (planar, minor-closed, preferential-attachment) often have constant κ :contentReference[oaicite:2]index=2.

4. High-Level Algorithm (Ideal Estimator)

- **3-pass procedure (no degree oracle):** :contentReference[oaicite:3]index=3
  (1) *Pass 1:* sample an edge e with probability $\frac{d_e}{d_E}$.
  (2) *Pass 2:* pick a neighbor $w \in N(e)$ uniformly.
  (3) *Pass 3:* check if $\{e, w\}$ closes a triangle; if so, run `IsAssigned` to ensure unique assignment; set $Y = 1$ if assigned, else 0.
  (4) Return estimator $X = d_E \cdot Y$.
- Run $\tilde{O}\big(d_E/T\big)$ independent copies in parallel and take median-of-means.

5. Key Lemma & Space Bound

- $d_E = \sum_{e \in E} d_e \leq 2m\kappa$ (Chiba–Nishizeki) :contentReference[oaicite:4]index=4.
- Hence estimator variance $[X] = d_E \cdot T = O(m\kappa T)$.
- **Space:** $\tilde{O}\big(d_E/T\big) = \tilde{O}(m\kappa/T)$.
- **Accuracy:** $(1 \pm \varepsilon)$-approx. w.h.p. via Chernoff/Chebyshev bounds.

6. Lower Bound

- Any constant-pass streaming algo achieving constant-factor approx. on graphs with degeneracy κ requires $\Omega(m\kappa/T)$ space :contentReference[oaicite:5]index=5.

7. Experimental Evaluation

No empirical results reported in this theoretical work.

8. Key Takeaways

- Leverage *low degeneracy* to circumvent worst-case space lower bounds.
- For $\kappa = O(1)$, space becomes $\tilde{O}(m/T)$, beating $m^{3/2}/T$ and $m/\sqrt{T}$.
- Technique extends to other subgraph-counting problems under bounded degeneracy.

9. Future Directions

- Multi-pass refinements to further reduce dependence on T.
- Extensions to dynamic/weighted streams and larger motifs (cliques of size $\ell > 3$).
- Validate practical performance on real large-scale graph streams.