Graph streams

$G = \langle (e_1, f_1), \ldots, (e_m, f_m) \rangle$ where each $e_i \in \mathbb{E}^\times \times \mathbb{E}$

Define a multi-graph $G = (V, E)$ in the natural way.

Essentially, every non-trivial graph problem requires $\Omega(n)$ bits of space to solve in the stream-only streaming model. However, some can be solved in space $O(n \log n)$.

Example: Connectivity in span only streams.

Algorithm: Maintain a spanning forest, i.e., initialize $T = \emptyset$.

While processing update $e_j$:

If $T U e_j$ does not contain a cycle:

Add $e_j$ to $T$

Output "connected" if and only if $|T| = n-1$.

What about turnstile streams?

Recall the following view of our turnstile streaming algorithm was presented:

Linear sketches:

Sketch: $\mathcal{S}$

Input:

Sketch main A

N rows
In particular, we gave a linear sketch for computing $F_2$ where the estimate for $\|f_2\|_2$ was precisely $\|f_2\|_2$ (i.e., we just used an exact algorithm for $\|f_2\|_2$ in sketch space instead of in the input space $\mathbb{R}^n$).

The idea for graph problems will be the same. We will run a connectivity algorithm in sketch space.

**Connectivity Algorithm (compute & spanning forest)**

* For each node, select an incident edge
* Contract selected edges, repeat until no edges remain
* Output "connected" if only a single supernode remains.

**Lemma:** Takes $O(n^2)$ steps to halt.

**Proof:** Difference between number of connected components in $G$ and number of supernodes monitored by the algorithm at each halve.

Define a signed node-edge adjacency matrix $A$ as follows:

\[
A_{ij}(E_{ik}) = 1 \text{ if } E_{ik} \in E
\]

\[
= 1 \text{ if } i = k \text{ and } E_{ik} \in E
\]

\[
= 0 \text{ otherwise}
\]

Example:

\[
a_1 = \begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

\[
a_2 = \begin{pmatrix}
-1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]
Lemma 4.4: For any subset of nodes \( S \subseteq V \), support \( \left( \sum_{i \in S} q_i \right) = E(S, V \setminus S) \).

E.g. In example, support \( \left( q_{14} \right) = \{ 31.33, 32.33 \} \).

Final Algorithm: For each node \( i \in [n] \), compute \( T_i \) (\( O(\log n) \)) Lo-sampling sketches of the vector \( q_i \). Call these sketches \( T_{i,1}, \ldots, T_{i,t} \).

* Run connectivity algorithm in sketch space. Specifically,
  * In round 1, use one Lo-sampling sketch from each node \( i \) to sample a random edge incident to \( i \). Merge all weight \( w \) into supernodes.
  * In round \( j \geq 2 \), for each supernode find an edge incident on the supernode as follows.
    * If \( S \subseteq V \) is a supernode, get an Lo-sampling sketch of the edges leaving supernode \( S \).

\[
\sum_{i \in S} T_{i,j} \quad \text{by linearity of the sketching algorithm, this is the same as an Lo-sketch for } \sum_{i \in S} q_i \quad \text{and by Lemma 4.4}
\]

the non-zero entries of \( \sum_{i \in S} q_i \) are precisely edges leaving supernode \( S \).

Total space usage is \( O(n \cdot \log^3 n) \).

Each node needs \( O(\log n) \) Lo-samplers

each with failure prob \( 6^{-1} \leq \frac{1}{n^3} \).
Definition: An $\alpha$-spanner of $G$ is a subgraph $H$ such that for all nodes $u, v$,
\[ d_G(u, v) \leq d_H(u, v) \leq \alpha \cdot d_G(u, v), \]
where $d_G$ and $d_H$ are shortest path distances in $G$ and $H$, respectively.

Algorithm in insert-only streams:
\[ 1. H \leftarrow \emptyset \]
\[ 2. \text{For each edge update } (u, v); \text{ if } d_H(u, v) \geq 2t, H \leftarrow H \cup \{(u, v)\} \]

Analysis of error:
- Distance increase by a factor of $2^t - 1$ since an edge $(u, v)$ is only for when it has already a degree of length at most $2^t - 1$.

Analysis of space usage:
Claim: A tuple of stream $H$ contains $O(n^2 \log n)$ edges.

Proof: First, observe that all cycles have length $2^t + 1$.

Case 1: $(u, v)$ already connected in $H$ then only keep $(u, v)$ if $(u, v)$ length $2^t + 1$

Case 2: $(u, v)$ not connected in $H$ then $(u, v)$ always keep but this doesn't create cycle.
Lemma. Let $t$ be an integer. A graph $H$ with no cycles of length $\leq 2t$ has $O(n^{1+1/2})$ edges.

Note. The complete bipartite graph has no triangles (cycles of length 3) and $\frac{n^2}{4}$ edges. So the requirement that $t$ be an integer is in some sense necessary (at least $t=2$ is necessary).

Proof. Let $d=\frac{2m}{n}$ be the average degree in $H$.

- Let $J$ be the core of $H$ (i.e., the graph formed from $H$ by removing nodes with degree at most $\frac{d}{2} - 1$ and all their incident edges).

- $J$ is not empty. To see this note,
  - every time you peel away a node and its incident edges, you remove $\leq \frac{d}{2}$ edges,
  - so if you remove in edges, you'd have to remove more than $\frac{m}{d} = n$ nodes! This is impossible.

- Grow a BFS of depth $t$ from an arbitrary node in $J$.

  - Because there are no cycles of length $\leq 2t+1$, we never encounter the same node twice in the BFS.
  - Because all degrees in $J$ are at least $\frac{d}{2}$, the number of nodes encountered in step $t$ of the BFS is at least $(\frac{d}{2} - 1)^t = \left(\frac{m}{h} - 1\right)^t$.

  - This $(\frac{m}{h} - 1)^t \leq n$ since we can't encounter more than $n$ distinct nodes, \(m \leq nh^{1/2}\).