Graph streams

$\sigma = \langle (e_1, \delta_1), \ldots (e_m, \delta_m) \rangle$ where each $e_i \in [n] \times [n]$

σ defines 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 insert-only streaming model. However, some can be solved in space $O(n \operatorname{polylog} n)$.

Example: Connectivity in insert-only streams.

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

While processing update e_j:

If $T \cup e_j$ does not contain a cycle:

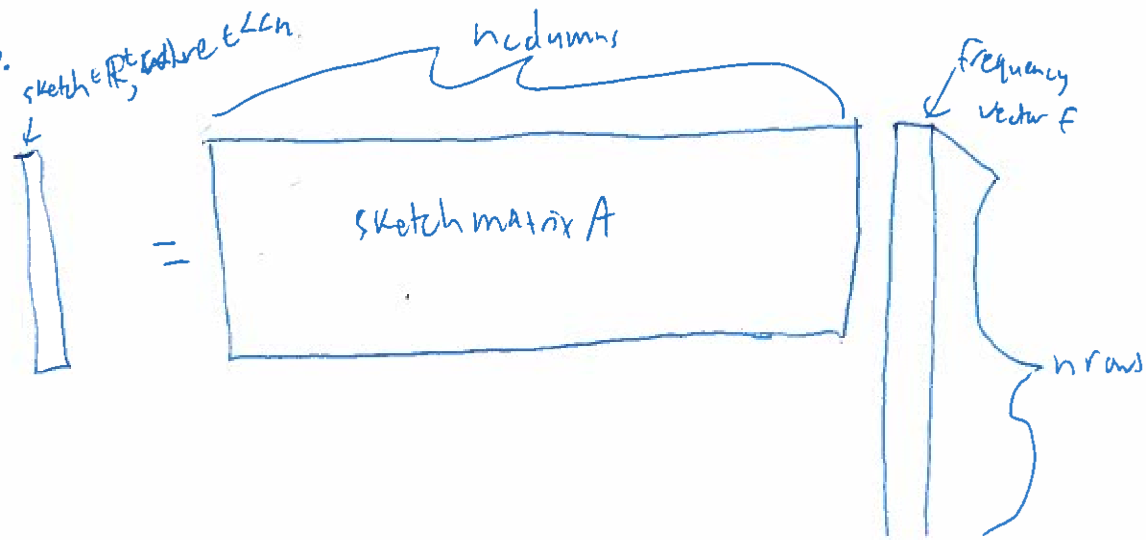
Add e_j to T

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

What about turnstile streams?

Recall two lectures ago the following view of our turnstile streaming algorithms was presented

Linear sketches.



In particular, we gave a linear sketch for computing F_2 where the estimate for $\|f\|_2^2$ was precisely $\|Sf\|_2^2$ (i.e. we just "ran 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.

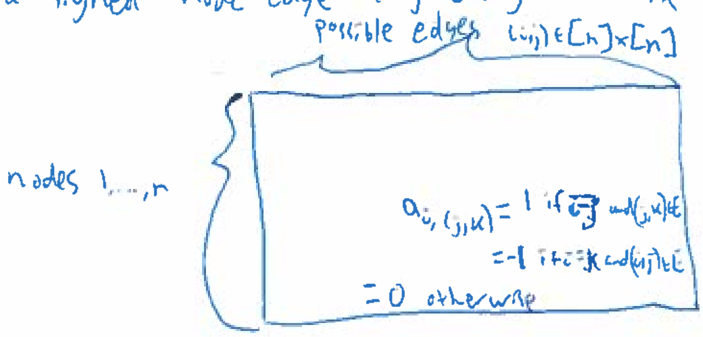
Connectivity Algorithm (computes spanning forest)

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

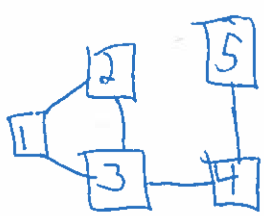
Lemma: Takes $O(\log n)$ steps to halt.

Proof: Difference between number of connected components in G and number of supernodes maintained by the algorithm at least halves.

Define a signed node-edge adjacency matrix as follows:



Example



$$\begin{array}{ccccccccccc} & \{1,2\} & \{1,3\} & \{1,4\} & \{1,5\} & \{2,3\} & \{2,4\} & \{2,5\} & \{3,4\} & \{3,5\} & \{4,5\} \\ a_1 = (& 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \;) \\ a_2 = (& -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \;) \end{array}$$

Lemma (*): For any subset of nodes $S \subseteq V$, $\text{support}\left(\sum_{i \in S} a_i\right) = E(S, V \setminus S)$

e.g. in example, $\text{support}(a_1 + a_2) = \{\{1,3\}, \{2,3\}\}$.

Final Algorithm: • For each node $i \in [n]$, compute $t = O(\log n)$ L_0-sampling sketches of the vector a_i. Call these sketches $T_{i,1}, \ldots T_{i,t}$

• Run connectivity algorithm in sketch space. Specifically

• In round 1, use one L_0-sampling ~~sk~~ sketch from each node i to sample a random edge incident to i. Merge all neighbors 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 L_0-sampling sketch of the edges leaving supernode S ~~4/~~ as

$\sum_{i \in S} T_{i,j}$ ← by ~~[illegible]~~ linearity of the sketching algorithm, this is the same as an L_0-sketch for $\underbrace{\sum_{i \in S} a_i}$, and by Lemma (*) the non-zero entries of ↑ are precisely edges leaving supernode S.

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

↑ each node needs $O(\log n)$ L_0-samplers each w/ failure probability $\leq \frac{1}{n^2}$.

Spanners

Definition: An α-spanner of G is a subgraph H such that for all nodes u, v,

$$d_G(u, v) \le d_H(u, v) \le \alpha \cdot d_G(u, v),$$

where d_G and d_H are shortest path distances in G and H respectively.

Algorithm in insert-only streams:

- $H \leftarrow \emptyset$
- For each edge update (u,v): if $d_H(u,v) \geq 2t$, $H \leftarrow H \cup \{(u,v)\}$

Analysis of error:

- Distances increase by at most a factor of $2t-1$ since an edge (u,v) is only forgotten if there's already a detour of length at most $2t-1$.

Analysis of space usage:

Claim: At end of stream H contains $O(n^{1+1/t})$ edges.

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

Cases:

- if (u,v) already connected in H then only keep (u,v) if cycle has length $\geq 2t+1$

- if (u,v) not already connected in H, then (u,v) always kept but this doesn't create a cycle.

Lemma: Let t be an integer. A graph H with no cycles of length $\leq 2t$ has $O(n^{1+1/t})$ 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, $t \geq 2$ is necessary).

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

\# of neighbors of a node

- Let J be the $\lfloor \frac{d}{2} \rfloor$-core of H (i.e., the graph formed from H by removing nodes with degree at most $\frac{d}{2} - 1$ & 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 $< \frac{d}{2}$ edges.
  - So if you remove m edges, ← i.e., J is empty, you'd have to remove ~~at least~~ more than $\frac{m}{\frac{d}{2}} = n$ nodes! This is impossible.

- Grow a BFS of depth t from an arbitrary node in J.
  - Because there are no cycles of length $< 2t+1$, 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

$$\left(\frac{d}{2} - 1\right)^t = \left(\frac{m}{n} - 1\right)^t.$$

- So $\left(\frac{m}{n} - 1\right)^t \leq n$ since we can't encounter more than n distinct nodes. $\implies m \leq n + n^{1+1/t}$.