Motivating Problem

- **Problem:** There are $n$ machines and each has the row of an adjacency matrix of a graph with $n$ nodes. A single message is communicated from each machine to a central machine. How many bits do these messages need to be such that the central machine can determine whether the graph is connected?

- **Answer:** $O(\text{polylog } n)$ bits suffice such that the connectivity can be determined with high probability.

- **Corollary:** $O(n \text{ polylog } n)$ bits suffice to determine whether a graph defined by a stream of edge insertions/deletions is connected.
First Ingredient: Sketching for $\ell_0$ Sampling

Lemma
There exists random matrix $A \in \mathbb{R}^{O(\log^2 N) \times N}$ such that for any $x \in \mathbb{R}^N$, with probability at least $1 - 1/poly(n)$, we can learn $(i, x_i)$ for some $x_i \neq 0$ from $Ax$.

Useful properties:

- **Union Bound:** Suppose we have multiple vectors $x^1, x^2, \ldots, x^t$, then, if the failure probability of the sketch is $\delta$, we can determine a non-zero element from everyone of them from

  $$Ax^1, Ax^2, \ldots, Ax^t$$

  with probability at least $1 - \delta t$.

- **Linearity:** Given $Ax$ and $Ay$, we can find a non-zero entry from $z = x + y$ since

  $$Az = A(x + y) = Ax + Ay$$
Consider the following (non-streaming) algorithm for connectivity:

- For each node, select an incident edge.
- For each connected component, select an incident edge.
- Repeat above line until process terminates.

Analysis:

- There are at most $\log n$ rounds since in each round, the size of every connected component either stops growing or doubles size.
- The set of all edges selected includes a spanning forest of the graph.
Third Ingredient: Signed Vertex-Edge Vectors

With each vertex $i$ of the graph, associate a length $\binom{n}{2}$ vector that is indexed by pairs on nodes. The only non-zero entries correspond to incident edges $\{i, j\} \in E$ and this entry is 1 if $j > i$ and $-1$ if $j < i$. E.g.,

\[
\begin{array}{cccccccccc}
& \{1,2\} & \{1,3\} & \{1,4\} & \{1,5\} & \{2,3\} & \{2,4\} & \{2,5\} & \{3,4\} & \{3,5\} & \{4,5\} \\
x_1 = ( & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 ) \\
x_2 = ( & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 ) \\
x_3 = ( & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 ) \\
x_4 = ( & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 ) \\
x_5 = ( & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 ) \\
\end{array}
\]

corresponds to a graph with edges \{1, 2\}, \{1, 3\}, \{2, 3\}, \{3, 4\}, and \{4, 5\}.

Lemma

Non-zero entries of $\sum_{i \in S} x_i$ correspond to edges between $S$ and $V \setminus S$.

Proof.

$j, k$th entry of $\sum_{i \in S} x_i$ is 0 iff $j, k \in S$ or $j, k \notin S$ or $(j, k) \notin E$. \qed
The Final Recipe

- **What players send:** Player with node $i$ sends $A_1 x_i, A_2 x_i, \ldots, A_{\log n} x_i$ where $A_1, A_2, \ldots$ are independent random matrices for $\ell_0$ sampling.

- **Central player emulates Boruvka’s algorithm:**
  - Can identify an incident edge from each node $i$ using $A_1 x_i$ since can find a non-zero entry of $x_i$ and such entries of $x_i$ are incident edges.
  - In round $t$, suppose we need to find an incident edge from a connected component $S$. Then, we can such an edge since

$$
\sum_{i \in S} A_t x_i = A_t \sum_{i \in S} x_i
$$

and we can therefore identify of non-zero elements of $\sum_{i \in S} x_i$ which gives a suitable edge.
How $\ell_0$ sketching works

- Let $S_0, S_1, \ldots, S_{\log N}$ be random subsets of $[N]$ where each element is in $S_i$ with probability $1/2^i$.

- To sketch the vector $x$, for each $S \in \{S_0, S_1, \ldots, S_{\log N}\}$ compute:

$$
a = \sum_{j \in S} jx_j \\
b = \sum_{j \in S} x_j \\
c = \sum_{j \in S} x_j r^j \pmod{p}
$$

where $r$ is a random value in range $1, \ldots, p - 1$ and $p = \text{poly}(N)$.

- Intuition:
  - If there's a unique $j \in S$ such that $x_j \neq 0$ then $a = jx_j$ and $b = x_j$ will allow us to determine a non-zero element and its location.
  - $c$ will allow us to test if there's a unique $j \in S$ such that $x_j \neq 0$

- We say $S$ passes the test if $a/b \in [N]$ and $c = br^{a/b} \pmod{p}$.
  - If all $S$ do not pass the test, output “fail”
  - Otherwise, pick a passing $S$. Claim that $(a/b)$th entry of $x$ is $b > 0$
**Analysis: Part 1**

**Lemma**

Let $A = \{ i \in N : x_i \neq 0 \}$ be the positions of non-zero entries.

- If $|A \cap S| = 1$, then $S$ passes the test and $x_{a/b} = b$.
- If $|A \cap S| \neq 1$, then $S$ doesn’t pass the test with high probability.

**Proof.**

- If $A \cap S = \{ j \}$ then $a = jx_j$, $b = x_j$, and $c = br^j \mod p$.
- If $|A \cap S| > 1$ then

$$f(z) = \sum_{j \in S} x_jz^j - bz^{a/b} \mod p$$

is a non-zero polynomial of degree at most $N$. Hence, it evaluates to 0 at a random $r$ with probability at most $N/(p - 1) < 1/\text{poly}(N)$. 

\[\square\]
Analysis: Part 2

Lemma
\[ \mathbb{P}[|A \cap S| = 1] \geq 1/8 \text{ for some } S. \]

Proof.
Pick \( i \) such that \( 2^{i-2} \leq |A| < 2^{i-1} \). Then,

\[
\mathbb{P}[|A \cap S_i| = 1] = \sum_{j \in A} \mathbb{P}[j \in S_i, k \not\in S_i \text{ for all } k \in A \setminus \{j\}]
\]

\[
= \sum_{j \in A} \frac{1}{2^i} \left( 1 - \frac{1}{2^i} \right)^{|A|-1}
\]

\[
= \frac{|A|}{2^i} \left( 1 - \frac{1}{2^i} \right)^{|A|-1}
\]

\[
> \frac{|A|}{2^i} \left( 1 - \frac{|A|}{2^i} \right) > 1/8
\]

Can boost the probability from 1/8 to \( 1 - 1/\text{poly}(n) \) by repeating the process \( O(\log n) \) times in parallel.
How to do it with hash functions: Part 1

For some applications of sketches, e.g., data stream processing, it is important that we don't use so much randomness.

Definition

A family $\mathcal{H}$ of functions from $A \rightarrow B$ is $k$-wise independent if for any distinct $x_1, \ldots, x_k \in A$ and $i_1, i_2, \ldots, i_k \in B$,

$$\mathbb{P}_{h \in \mathcal{R} \mathcal{H}}[h(x_1) = i_1, h(x_2) = i_2, \ldots, h(x_k) = i_k] = \frac{1}{|B|^k}$$

For example, if $A \subset \{0, 1, 2, \ldots, p - 1\}$ and $B = \{0, 1, 2, \ldots, p - 1\}$. Then,

$$\mathcal{H} = \{h(x) = \sum_{i=0}^{k-1} a_i x^i \mod p : 0 \leq a_0, a_1, \ldots, a_{k-1} \leq p - 1\}$$

is a $k$-wise independent family of hash functions. Note that we can store random $h$ using $O(k \log p)$ bits.
Hash Functions

Definition
A family $\mathcal{H}$ of functions from $A \rightarrow B$ is $k$-wise independent if for any distinct $x_1, \ldots, x_k \in A$ and $i_1, i_2, \ldots, i_k \in B$,

$$\mathbb{P}_{h \in \mathcal{H}}[h(x_1) = i_1, h(x_2) = i_2, \ldots, h(x_k) = i_k] = \frac{1}{|B|^k}$$

Example
Suppose $A \subset \{0, 1, 2, \ldots, p - 1\}$ and $B = \{0, 1, 2, \ldots, p - 1\}$. Then,

$$\mathcal{H} = \{ h(x) = \sum_{i=0}^{k-1} a_i x^i \mod p : 0 \leq a_0, a_1, \ldots, a_{k-1} \leq p - 1 \}$$

is a $k$-wise independent family of hash functions.

Note. If $|B|$ is not prime or $|A| > |B|$ more ideas are required.
How to do it with hash functions: Part 2

- To define $S_0, S_1, S_2, \ldots$, pick $h$ from a 2-wise independent family of hash functions.
- Let $S_i = \{x \in [N] : h(x) \text{ is divisible by } 2^i \}$ and so
  \[
  \gamma_i = \mathbb{P} [j \in S_i] = (\lfloor (p - 1)/2^i \rfloor + 1)/p \approx 1/2^i
  \]
- If $i$ satisfies that $2^{i-2} \leq |A| < 2^{i-1}$ then,
  \[
  \mathbb{P} [|A \cap S_i| = 1] = \sum_{j \in A} \mathbb{P} [j \in S_i, k \notin S_i \text{ for all } k \in A \setminus \{j\}]
  \]
  \[
  = \sum_{j \in A} \mathbb{P} [j \in S_i] \mathbb{P} [k \notin S_i \text{ for all } k \in A \setminus \{j\}|j \in S_i]
  \]
  \[
  = \sum_{j \in A} \gamma_i (1 - \mathbb{P} [\bigcup_{k \in A \setminus \{j\}} \{k \in S_i\}|j \in S_i])
  \]
  \[
  \geq \sum_{j \in A} \gamma_i (1 - \sum_{k \in A \setminus \{j\}} \mathbb{P} [k \in S_i|j \in S_i])
  \]
  \[
  > \sum_{j \in A} \gamma_i (1 - \gamma_i |A|) = |A| \gamma_i (1 - \gamma_i |A|) > 1/8
  \]