Let $\sigma = \langle a_1, a_2, ..., a_m \rangle$ be a stream; each $a_i$ is a pair $(j, c)$, where $j \in [n]$ and $c$ is an integer—meaning of $a_i$ is: update $f_j \leftarrow f_j + c$, where $i \in [1..m]$.

**Algorithm 1 Sketch Algorithm**

1. **Initialize**: $C[0..k] \leftarrow [0..0]$ //count vector
2. Choose random hash function $h : [n] \rightarrow [k]$ from a 2-universal process
3. Choose random hash function $g : [n] \rightarrow \{-1, +1\}$ from a 2-universal process
4. **Process** $a_i = (j, c')$
   - $C[h(j)] \leftarrow C[h(j)] + c' \ast g(j)$
5. **Output**: on query $a$, report $f_a = g(a) \ast C[h(a)]$

### 3.0.1 Analysis

Let $e_j$ be the $k$-vector with 1 in $h(j)$ co-ordinate, and 0 otherwise. For stream $\sigma$,

$$\sigma \rightarrow f = (f_0, f_1, ..., f_{n-1}) \rightarrow C[\sigma]$$

$$\sigma \rightarrow f_0 g(0)e_0 + f_1 g(1)e_1 + ... + f_{n-1} g(n-1)e_{n-1}$$

$$\sigma \rightarrow ||M|| \begin{pmatrix} f_0 \\ f_1 \\ . \\ . \\ f_{n-1} \end{pmatrix}$$

**Definition 3.1** Fix $\sigma \rightarrow C[\sigma]$. $C$ is a sketch if, given 2 streams $\sigma_1$ and $\sigma_2$, the concatenation of the two streams $C[\sigma_1 \cdot \sigma_2]$ can be obtained from $C[\sigma_1]$ and $C[\sigma_2]$.

If $C[\sigma_1] = M \ast f^{\sigma_1}$, $C[\sigma_2] = M \ast f^{\sigma_2}$,

$$C[\sigma_1 \cdot \sigma_2] = M \ast f^{\sigma_1 \cdot \sigma_2} = M \ast (f^{\sigma_1} + f^{\sigma_2}) = C[\sigma_1] + C[\sigma_2]$$

Fix $a \in [n]$. Let $X = \hat{f}_a$. Define random variable $Y_j$,

$$Y_j = \begin{cases} 1 & \text{if } h(j) = h(a); //a \text{ and } j \text{ maps to the same bin in } C[ ] \\ 0 & \text{otherwise.} \end{cases}$$
Now we compute the variance.

\[ E[X] = f_a + \sum_{j \in [n] \setminus \{a\}} f_j E[g(a)g(j)Y_j] \]

\[ = f_a + \sum_{j \in [n] \setminus \{a\}} f_j E[g(a)g(j)Y_j] \quad // g() \text{ and } h() \text{ are independent} \]
\[ \text{note that } E[g(a)] = E[g(j)] = 0 \]
\[ = f_a \]

Now we compute the variance.

\[ \text{Var}[x] = 0 + \text{Var}[\sum_{j \in [n] \setminus \{a\}} f_j g(a)g(j)Y_j] \]
\[ = E[\sum_{j \in [n] \setminus \{a\}} f_j g(a)g(j)Y_j^2] - E[\sum_{j \in [n] \setminus \{a\}} f_j g(a)g(j)Y_j]^2 \]
\[ = E[\sum_{j \in [n] \setminus \{a\}} f_j g(a)^2g(j)^2Y_j^2 + \sum_{i,j \in [n] \setminus \{a\}, i \neq j} f_i f_j g(i)g(j)Y_iY_j] \quad // \text{by pairwise independence} \]
\[ \text{note that } g(i)^2 = (+1)^2 = (-1)^2 = 1 \]
\[ = E[\sum_{j \in [n] \setminus \{a\}} f_j g(a)^2Y_j^2] + \sum_{i,j \in [n] \setminus \{a\}, i \neq j} f_i f_j g(i)g(j)Y_iY_j \]
\[ = \sum_{j \in [n] \setminus \{a\}} E[f_j^2 Y_j^2] + \sum_{i,j \in [n] \setminus \{a\}, i \neq j} f_i f_j E[g(i)g(j)]E[Y_iY_j] \]
\[ = \sum_{j \in [n] \setminus \{a\}} f_j^2 E[Y_j^2] + \sum_{i,j \in [n] \setminus \{a\}, i \neq j} f_i f_j E[g(i)]E[g(j)]E[Y_iY_j] \]
\[ = \sum_{j \in [n] \setminus \{a\}} f_j^2 E[Y_j^2] + 0 \]
\[ = \sum_{j \in [n] \setminus \{a\}} f_j^2 E[Y_j^2] \quad //Y_j = 0 \text{ or } 1; Y_j^2 = Y_j \]
\[ = \frac{1}{k} \sum_{j \in [n] \setminus \{a\}} f_j^2 \quad // \text{Pr}[h(j) = h(a)] = \frac{1}{k} \]
\[ = \frac{1}{k}(\|f\|_2^2 - f_a^2) \]

We now compute the error probability. By Chebyshev’s inequality,

\[ \Pr[|X - E[X]| \geq \epsilon \sqrt{(\|f\|_2^2 - f_a^2)}] \leq \frac{\text{Var}[X]}{\epsilon^2 (\|f\|_2^2 - f_a^2)} \leq \frac{1}{k\epsilon^2} \]

if \( k \geq \frac{3}{\epsilon^2} \),

\[ \Pr[|X - E[X]| \geq \epsilon \sqrt{(\|f\|_2^2 - f_a^2)}] \leq \frac{1}{3} \]

Also,

\[ \Pr[|\hat{f}_a - f_a| \geq \epsilon \sum_{j \in [n]} f_j] \leq \Pr[|X - E[X]| \geq \epsilon \sqrt{(\|f\|_2^2 - f_a^2)}] \leq \frac{1}{3} \]
3.1 The Tug-of-War Sketch

Problem: We have a stream \(a_1, a_2, ..., a_m\), where each \(a_i\) has the form \((j, c)\), where \(j \in [n]\) and \(c\) is an integer. The frequency of element \(j\) in the stream is calculated when \((j, c)\) appears in the stream as follows:

\[ f_j \leftarrow f_j + c \]

Estimate:

\[ F_2 = \sum_{j \in [n]} f_j^2 = \|f\|^2_2 \]

where \(f = (f_0, f_1, ..., f_n - 1)\) is the frequency vector of elements appearing in the stream.

The above formula can be generalized for \(k \geq 0\) as follows:

\[ F_k = \sum_{j \in [n]} f_j^k \]

Algorithm 2 Tug-of-War Sketch Algorithm

1. Initialize:
   \(x \leftarrow 0\)
   Choose random hash function \(h : [n] \rightarrow \{-1, +1\}\) from a 4-universal process

3. Process \(a_i = (j, c)\)
   \(x \leftarrow x + h(j) \cdot c\)

5. Output: \(x^2\)

3.1.1 Analysis

Let \(X\) denote \(x\) at the end of the stream. Let \(Y_j = h(j)\). So, \(X = \sum_{j \in [n]} f_j Y_j\).

\[ E[X^2] = \sum_{j \in [n]} f_j^2 E[Y_j^2] + \sum_{i,j \in [n], i \neq j} f_i^2 f_j^2 E[Y_i Y_j] \]

note that \(E[Y_j^2] = 1\), and by pairwise independence \(E[Y_i Y_j] = 0\), hence,

\[ E[X^2] = \sum_{j \in [n]} f_j^2 + 0 = F_2 \]

\[ \Rightarrow \text{var}[X^2] \leq 2F_2^2 \]

To reduce the error gap, do:

- Run \(t\) parallel, independent copies of Tug-of-War sketch algorithm.
- Return \(Z\), which is the average of the outputs of the \(t\) copies.

For \(Z\), \(E[Z] = F_2\), which leads to \(\text{var}[Z] \leq \frac{2F_2^2}{t}\).
\[
\Rightarrow \Pr[|Z - F_2| \geq \epsilon F_2] \leq \frac{\text{var}[Z]}{(\epsilon F_2)^2}
\]

\[
\Pr[|Z - F_2| \geq \epsilon F_2] \leq \frac{2F_2^2}{t\epsilon F_2^2} = \frac{2}{t\epsilon^2}
\]

for \( t \geq \frac{6}{\epsilon^2} \),

\[
\Pr[ |Z - F_2| \geq \epsilon F_2 ] \leq 1/3
\]

For \( t \) copies of the algorithm, with 5 items for example,

\[
t \ast \left( \begin{array}{c}
1,\ 1,\ -1,\ 1,\ -1 \\
. \\
. \\
. \\
M
\end{array} \right) \ast \left( \begin{array}{c}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5
\end{array} \right) = \frac{||Mf||_2^2}{t}
\]

where

\[
\Rightarrow Z = \frac{||Mf||_2^2}{t} \in [(1 - \epsilon) F_2, (1 + \epsilon) F_2]
\]

by taking square root,

\[
\frac{||Mf||_2}{\sqrt{t}} \in [\sqrt{(1 - \epsilon)} \ ||f||_2, \ \sqrt{(1 + \epsilon)} \ ||f||_2]
\]

**Note:** The above operation is called *dimension reduction.* Johnson-Lindenstrauss lemma states that a small set of points in a high-dimensional space can be embedded into a space of much lower dimension in such a way that distances between the points are nearly preserved. When \( t = \frac{\log n}{\epsilon^2} \), the distance is preserved with high probability.