Recap

**Thm.** The max-loaded bin $O \left( \frac{\log N}{\log \log N} \right)$ balls with prob. at least $1 - \frac{1}{N}$.

**Proof:** High level steps:

1. prob. of any bin receiving $\frac{1}{N}$ (for union bounding over bins)
   - want: $\frac{1}{N}$
2. prs. of there being at least one bin with at least
   - there many balls.
   - want: $\frac{1}{N}$

union bound:
$$p(A \cup B \cup C \ldots) \leq p(A) + p(B) + \ldots$$
Recap

\[ p(\text{bin } i \text{ has at least } k \text{ balls}) \leq \binom{N}{k} \left( \frac{1}{N} \right)^k \]

\[ = \frac{n!}{(N-k)!k!} \cdot \frac{k!}{N^k} \cdot \frac{1}{N^k} \]

\[ \leq \frac{N!}{k!(N-k)!} \cdot \frac{1}{N^k} \]

\[ = \frac{1}{k!} \]

Stirling's approx. \[ k! \approx \sqrt{2\pi k} \left( \frac{k}{e} \right)^k \]

Choose \( k = O \left( \frac{\log N}{\log \log N} \right) \) give desired result
Clarification:

Sets $S_1, S_2, \ldots, S^{(N)}_{(k)}$

Events $E_1, \ldots, E^{(N)}_{(k)}$

$E_j := \text{Set } S_j \text{ landing in bin } i$

$E = \text{bin } i \text{ getting at least } k \text{ balls}$

$= E_1 \cup E_2 \cup \ldots \cup E^{(N)}_{(k)}$

$P(E) = P(E_1 \cup \ldots \cup E^{(N)}_{(k)})$

$\leq \sum P(E_i)$
power - 2 - choice :

$O \left( \log \log N \right)$

proof (intuition/sketch):

height (b) = num of balls in its bin after placing b

Prob. of an incoming ball getting height 3 is at most?

Fraction of bins that can have $\geq 2$ balls?

- at most $\frac{1}{2}$

$\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$

Expected # of bin with 3 balls is at most $= \frac{N}{4}$
Same case for height 4:

\[
\frac{1}{4} \cdot \frac{1}{4} = \frac{1}{2^{4-2}}
\]

Prob. of an incoming ball getting height \( h \) is at most

\[
\frac{1}{h-2}
\]

\[
\frac{2}{2}
\]

Choose \( h = O(\log \log N) + 2 \) \( \Rightarrow \) prob. \( \leq \frac{1}{N} \)
power-of-d-choice

Then, for any \( d \geq 2 \), for d-choice process, max load

\[
\frac{\log \log N}{\log d} = O(1)
\]

with \( p \) at least \( 1 - O(\sqrt{n}) \)

Diminishing returns.
Data streaming model:

- Elements coming as a "stream"
- Limited storage (cannot store all elements)

Notation:
- Elements in the stream $a_1, a_2, \ldots$ from alphabet $U$

Example functions:
1. Sum (easy)
2. Max (easy)
3. Median (tricky)
4. Heavy hitters i.e. 'most appeared'
5. Num. $S$ distinct elements
Natural option: Sampling

Eq.: Unique element

Sample 10% \to\text{count uniques} \to \text{multiply by 10}.

Can lead to incorrect answer

Counterex:
Stream of len. \( n \)

\( \frac{n}{2} \) of unique. \( \frac{n}{4} \) appear twice

10% Sampling:
\[ \# \text{uniques} = 0.1 \times \frac{n}{2} + \frac{n}{4} (2 \times 0.1 - 0.1^2) \]

\[ \leq \frac{n}{10} \]

So actually \( \# \text{uniques} = \frac{n}{10} = \frac{n}{10} \]
Abstraction:
Stream at time $t$ as a vector $x \in \mathbb{Z}^{t+1}$

$$x = (x_1, x_2, \ldots, x_{t+1})$$

number of time element $i$ of $U$ has been seen until time $t$

Generalization: $- (\text{add } e) (\text{del } e)$

Ex: $U = \{A, B, c\}$

- add $A$
- add $B$
- del $A$

$(0, 0, 0), (1, 0, 0), (1, 1, 0), (0, 1, 0), \ldots$

Assumption: $- \# \text{del} \leq \# \text{add}$
Heavy hitters:

\[ \epsilon \text{-heavy hitters: indices } i \text{ s.t. } x_i > \epsilon \| x \|_1 \]

Count-query:

At time \( t \), for any index \( i \)

output an estimate

\[ y_i \in x_i + \epsilon \| x \|_1 \]

For \( \epsilon \)-heavy hitter: can look at \( i \) s.t. \( y_i > 0 \)
Hashing-based Solution: Count-Min Sketch

Step 1: $h: \mathbb{U} \rightarrow [M]$

Array $A[1..M]$ can store non-negative integers

\[\text{add } i:\]

\[A[h(i)] ++\]

else (del $i$)

\[A[h(i)] --\]

Estimate for $x_i$: $y_i = A[h(i)]$
\[ A[h(i)] = \sum_{j \in \mathcal{U}} x_j^t \cdot 1\{h(i) = h(j)\} \]

\[ = x_i^t + \sum_{j \neq i} x_j^t \cdot 1\{h(i) = h(j)\} \]

Error in estimate

Assume \[\mathcal{H}\] universal hash family.

\[ p(h(x_i) = h(x_j)) \leq \frac{1}{m} \]
\[ E \left[ \sum_{j \neq i} x_j^t \mathbf{1} \{ h(j) = h(i) \} \right] = \sum_{j \neq i} x_j^t \Pr(h(j) = h(i)) \leq \sum_{j \neq i} x_j^t \frac{1}{M} \leq \frac{1}{M} \left( \| x_i \|_1 - x_i^t \right) \leq \frac{1}{M} \left( \| x_i \|_1 - \left\| x_i^+ \right\|_1 \right) \leq \frac{1}{M} \left( \| x_i^+ \|_1 \right) \leq \frac{1}{M} \]
Step 2:

Boost success probability

Idea: Repeat & take the best outcome

1. Hash functions $h_1, \ldots, h_k$
2. Arrays $A_1, \ldots, A_k$

Same approach as earlier.

$y_i = \min_k A_k[h_k(i)]$

$(\ddagger$: overestimate)