Load balancing.
Balls in Bins.
Power of two choices.
Cuckoo hashing.

Balls in bins.
For each of \( n \) balls, choose random bin: \( X_i \) balls in bin \( i \).

From Union Bound: \( Pr[\bigcup_i A_i] \leq \sum_i Pr[A_i] \)

\[ Pr[|X_i| \geq k] \leq \binom{n}{k} \left( \frac{1}{n} \right)^k \leq \frac{n^k}{k^k} \left( \frac{1}{n} \right)^k = \frac{1}{k!} \]

Choose \( k \), so that \( Pr[|X_i| \geq k] \leq \frac{1}{n^k} \).

\[ Pr[\text{any } X_i \geq k] \leq n \times \frac{1}{n^k} = \frac{1}{k^k} \rightarrow \text{max load } \leq k \text{ w.p. } \geq 1 - \frac{1}{n} \]

\( k! \geq \left( \frac{k}{2} \right)^k \)

**Lemma:** Max load is \( \Theta(\log n) \) with probability \( \geq 1 - \frac{1}{n} \).

Much better than \( n \).

Actually Max load is \( \Theta(\log n/\log \log n) \) w.h.p.

(W.h.p. - means with probability at least \( 1 - O(1/n^2) \) for today.)

\[
\left( \frac{n}{k} \right)^k \leq \frac{n^k}{k^k} \leq \left( \frac{ne}{k} \right)^k
\]

\[ \left( \frac{n}{k} \right)^k = \frac{n(n-1)(n-k+1)}{k(k-1)\cdots 1} \leq \frac{k^k}{k!} \]

\[ n(n-1)\cdots(n-k+1) \leq n^k \]

\[ k! \geq \left( \frac{k}{2} \right)^k \]

\[ \delta = \frac{1}{m^2} \text{ for today. } c \text{ is 1 or 2.} \]

**Analysis.**

\( n/8 \) balls in \( n \) bins.

Each ball chooses two bins at random. Picks least loaded.

View as graph.
Bin is vertex.
Each ball is edge.

Analysis Intuition:
Add edge, add one to lower endpoint's "count."

Max load is max vertices count.
If max count is \( k \), neighbors with counts \( \geq k-1, k-2, k-3, \ldots \) and so on!
No cycles and max-load \( k \rightarrow \geq 2^{k/2} \) nodes in tree.
No connected component of size \( X \) and no cycles \( \Rightarrow \) max load \( O(\log^2 X) \).

Will show:
- Max conn. comp is \( O(\log n) \) w.h.p.
- Average induced degree is small. (E.g.: cycle degree 2)
- Extend tree intuition.

---

**Simplest.**

Load balance: \( m \) balls in \( n \) bins.

For simplicity: \( n \) balls in \( n \) bins.

Round robin: load 1!

Centralized! Not so good.

Uniformly at random? Average load 1.

Max load?

\( n \). Uh Oh!

Max load with probability \( \geq 1 - \delta \)?

\[ \delta = \frac{1}{m^2} \text{ for today. } c \text{ is 1 or 2.} \]
Connected Component.

Claim: Component size in $n$ vertex, $\theta$ edge random graph is $O(\log n)$ w/ prob. $\geq 1 - \frac{1}{m}$.

Proof: Size $k$ component, $C$, contains $\geq k - 1$ edges.

\[
\Pr[|C| \geq k] \leq \left( \binom{n}{k} \frac{n/8}{k} \right) \left( \frac{k}{n} \right)^{2(k-1)}
\]

Possible $C$. Which edges. Prob. both endpoints inside $C$.

\[
\Pr[|C| \geq k] \leq \frac{n}{k} \left( \frac{n/8}{k} \right) \left( \frac{k}{n} \right)^{2k} \leq \frac{n}{k} \left( \frac{e^2}{8} \right)^k \leq \frac{n}{k} (0.93)^k (2)
\]

Choose $k = -(c + 1) \log 0.93 n$ make probability $\leq 1/n^2$.

Not dense.

Induced degree of node on subset, $S$, is degree of internal edges.

Induced degree of nodes in blue subset is $2$, not $5!$

Claim: Average induced degree on any subset of nodes is $\leq 8$ with probability $\geq 1 - O(1/2^k)$.

Proof: Induced degree $\geq 8$

$\rightarrow 4k$ internal edges for subset of size $k$.

\[
\Pr[\text{dense } S] \leq \frac{n}{k} \left( \frac{n/8}{4k} \right) \left( \frac{k}{n} \right)^{4k} \leq \left( \frac{e^2}{32} \right)^k \left( \frac{k}{n} \right)^{3k} \leq \left( \frac{k}{n} \right)^{3k}
\]

Starts at $1/n^2$, decreasing till $k \leq n/8$ (at least)

$\rightarrow$ Total $O(1/n^2)$.

Cuckoo hashing.

Hashing with two choices: max load $O(\log\log n)$.

Cuckoo hashing:

Array. Two hash functions $h_1, h_2$.

Insert $x$: place in $h_1(x)$ or $h_2(x)$ if space.

Else bump elt $y$ in $h_i(x)$ u.a.r.

Bump $y$. $x$: place $y$ in $h_i(y)$. $h_i(x)$ if space.

Else bump $y'$ in $h_i(y)$.

If go too long. Fail. Rehash entire hash table.

Fails if cycle.

$C_l$ - event of cycle of length $l$.

\[
\Pr[C] \leq \frac{m}{m + 1} \left( \binom{n}{k} \left( \frac{n}{2k} \right)^{2(k-1)} \right) \leq \left( \frac{e^2}{8} \right)^l
\]

Probability that an insert makes a cycle of length $l \leq \frac{1}{n} \left( \frac{e^2}{8} \right)^l$.

Rehash every $O(n)$ inserts (if $\leq n/8$ items in table.)

$O(1)$ time on average.

Removal Process!

Random Graph: Component size is $\log n$ and max-induced degree is $8$ w.h.p.

Process: Remove degree $\leq 16$ nodes and incident edges. Repeat.

Claim: $O(\log X)$ iterations where $X$ is max component size.

For any connected component:

- Average induced degree $8 \rightarrow$ half nodes removed in each iteration.
- $\log X$ iterations to remove all nodes.

Claim: Max load is $O(\log\log n)$ w.h.p.

Recall edge corresponds to ball.

Height of ball, $h_i$, is load of bin when it is placed in bin.

Corresponding edge removed in iteration $r_i$.

Property: $h_i \leq 16r_i$.

Case $r_i = 1$: only $16$ balls incident to bin $\rightarrow h_i \leq 16$.

Induction: Previous removed edges(ball) induce load $\leq 16(r_i - 1)$.

$+16$ edges/balls this iteration.

$\rightarrow h_i \leq 16r_i$.

Power of two choices.

Max load: $\log X$ where $X$ is max component size.

$X$ is $O(\log n)$ with high probability.

Max load is $O(\log\log n)$.

Sum up

Balls in bins: $\Theta(\log n / \log\log n)$ load.

Power of two: $\Theta(\log\log n)$.

Cuckoo hashing.
See you on Thursday...