1 Overview

In the last lecture we talked about “The Power of Two Choices”.

In this lecture we will continue, and talk about cuckoo hashing.

2 The Power of Two Choices

- one choice: $\Theta(\frac{\log n}{\log \log n})$ max load
- two choices: $\Theta(\log \log n)$

The proof is by induction.

\begin{align*}
V_i(t) &= \text{number of bins at height } \geq i \text{ after } t \text{ balls } \\
V_i(t) &\leq \beta_i n \text{ w.h.p. } \\
\beta_4 &= \frac{1}{4} \\
\beta_{i+1} &= 2\beta_i^2 \\
Y_t &= 1 \text{ if ball was placed at height } i+1 \text{ and } V_i(t+1) \leq \beta_i n
\end{align*}

$\Rightarrow Y_t$ is stochastically dominated by

$Z_t \sim \{0, 1\}$ i.i.d. pr. $\beta_i^2$

\begin{align*}
\sum Y_t &\leq \sum Z_t \\
\mathbb{E}[\sum Z_t] &= \beta_i^2 n = \frac{\beta_{i+1} n}{2}
\end{align*}

If $\beta_{i+1} \geq C \log n$ for sufficiently large $C$, we have

$\mathbb{P}[\sum Z_t \geq \beta_{i+1} n] \leq e^{-\Omega(\beta_{i+1} n)} \leq O\left(\frac{1}{n^c}\right)$
\( E_i \) = event that \( \sum Y_t \geq \beta_{i+1}n \) \hfill (10)

\( P[E_i] < n^{-10} \) \hfill (11)

\( Q_i \) = event that \( V_i(t) \leq \beta n \) \hfill (12)

\( P[Q_i] = 1 \) \hfill (13)

\( P[Q_i | Q_i] \leq P[E_i | Q_i] \) \hfill (14)

\( \Rightarrow P[Q_i] \leq n^{-9} \) \hfill (15)

\( \Rightarrow P[\text{any } Q_i] \leq n^{-8} \) \hfill (16)

The above analysis works until \( \beta n \leq C \log n \), which corresponds to \( i^* = \Theta(\log \log n) \). We will analyze this case in the next lecture.

## 3 Cuckoo hashing

“Hash each element to two points”:

- \( n \) vertices (bins)
- \( m \) edges (balls)

The analysis uses Erdos-Renyi graphs.

- store each element in one of the locations
- each location stores at most 1 element \( \Rightarrow O(1) \) lookup, insertion is \( O(1) \) expected

\[
\mathbb{P}[	ext{given length } k \text{ cycle exists} \cdots \rightarrow i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow i_k \rightarrow i_1 \rightarrow \cdots] \leq \left(\frac{O(m)}{n^2}\right)^k \quad (17)
\]

\[
\mathbb{P}[\text{edge } e \text{ exists}] \leq \frac{m}{\binom{n}{2}} = O\left(\frac{m}{n^2}\right) = O\left(\frac{1}{n}\right) \quad (18)
\]

\[
\mathbb{P}[\text{any length } k \text{ cycle exists}] \leq n^k \left(\frac{O(m)}{n^2}\right)^k = (O\left(\frac{m}{n}\right))^k \leq \frac{1}{100^k} \text{ if } n \geq 100m \quad (19)
\]

\( \Rightarrow \mathbb{P}[\text{any cycle exists}] \leq \frac{1}{99} \cdot \frac{98}{99} \) probability that no cycle exists for \( n = O(m) \).

If a cycle is encountered during insertion, re-hash, rebuild the hash table. \( \mathbb{E}[\text{number of times we rebuild}] = O(1) \).
\[ \mathbb{E}[\text{time to build}] = \sum_{i=1}^{m} \mathbb{E}[\text{time to insert } i^{th} \text{ element}] \]

\[ \leq m \cdot \mathbb{E}[\text{size of component of any element}] \]

\[ \leq 2m \cdot \mathbb{E}[\text{size of component of a vertex}] \]

\[ = O(m) \]

This follows from a bound on the expected size of a component in an Erdos-Renyi graph \( G(n, p) \) with \( n \) vertices and probability \( p \).

\[ f(n, p) = \mathbb{E}[\text{size of component in } G(n, p)] \]

\[ \leq 1 + p \cdot (n - 1) \cdot f(n - 1, p) \]

\[ \leq 1 + np + (np)^2 + \cdots \]

\[ \leq \frac{1}{1 - np} \]

References
