CS 388R: Randomized Algorithms Fall 2015

Lecture 9 - Sep. 28, 2015

Prof. Eric Price *Scribe: Tianyang Li*

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.

$$V_i(t) = \text{number of bins at height} \geq i \text{ after } t \text{ balls} \tag{1}$$
$$V_i(t) \leq \beta_i n \text{ w.h.p.} \tag{2}$$
$$\beta_4 = \frac{1}{4} \tag{3}$$
$$\beta_{i+1} = 2\beta_i^2 \tag{4}$$
$$Y_t = 1 \text{ if ball was placed at height } i+1 \text{ and } V_i(t+1) \leq \beta_i n \tag{5}$$

$\Rightarrow Y_t$ is stochastically dominated by

$$Z_t \sim \{0,1\} \text{ i.i.d. pr. } \beta_i^2 \tag{6}$$

$$\sum Y_t \leq \sum Z_t \tag{7}$$
$$\mathbb{E}[\sum Z_t] = \beta_i^2 n = \frac{\beta_{i+1} n}{2} \tag{8}$$

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

$$\mathbb{P}[\sum Z_t \geq \beta_{i+1} n] \leq e^{-\Omega(\beta_{i+1} n)} \leq O(\frac{1}{n^c}) \tag{9}$$

$$E_i = \text{event that } \sum Y_t \geq \beta_{i+1} n \tag{10}$$
$$\mathbb{P}[E_i] < n^{-10} \tag{11}$$
$$Q_i = \text{event that } V_i(t) \leq \beta_i n \tag{12}$$
$$\mathbb{P}[Q_4] = 1 \tag{13}$$
$$\mathbb{P}[\bar{Q}_{i+1}|Q_i] \leq \mathbb{P}[E_i|Q_i] \tag{14}$$
$$\Rightarrow \mathbb{P}[\bar{Q}_i] \leq n^{-9} \tag{15}$$
$$\Rightarrow \mathbb{P}[\text{any } \bar{Q}_i] \leq n^{-8} \tag{16}$$

The above analysis works until $\beta_i n \leq C \log n$, 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

$$\mathbb{P}[\text{given length } k \text{ cycle exists} \cdots \to i_1 \to i_2 \to \cdots \to i_k \to i_1 \to \cdots] \leq (\frac{O(m)}{n^2})^k \tag{17}$$
$$\mathbb{P}[\text{edge } e \text{ exists}] \leq \frac{m}{\binom{n}{2}} = O(\frac{m}{n^2}) = O(\frac{1}{n}) \tag{18}$$
$$\mathbb{P}[\text{any length } k \text{ cycle exists}] \leq n^k (\frac{O(m)}{n^2})^k = (O(\frac{m}{n}))^k \leq \frac{1}{100^k} \text{ if } n \geq 100m \tag{19}$$

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

If a cycle is encountered during insertion, re-hash, rebuild the hash table. $\mathbb{E}[\text{number of times we rebuild}] = O(1)$.

$$
\begin{aligned}
\mathbb{E}[\text{time to build}] &= \sum_{i=1}^{m} \mathbb{E}[\text{time to insert } i^{\text{th}} \text{ element}] \\
&\leq m \cdot \mathbb{E}[\text{size of component of any element}] \\
&\leq 2m \cdot \mathbb{E}[\text{size of component of a vertex}] \\
&= O(m)
\end{aligned}
$$

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.

$$
\begin{aligned}
f(n, p) &= \mathbb{E}[\text{size of component in } G(n, p)] \\
&\leq 1 + p \cdot (n-1) \cdot f(n-1, p) \\
&\leq 1 + np + (np)^2 + \cdots \\
&\leq \frac{1}{1 - np}
\end{aligned}
$$

References

[MU05] Michael Mitzenmacher, Eli Upfal. Probability and Computing: Randomized Algorithms and Probabilistic Analysis *Cambridge University Press*, 2005.

[PR01] Rasmus Pagh, Flemming Friche Rodler. Cuckoo hashing *Journal of Algorithms*, 51 (2004) 122-144.