Sample Solution for Problem Set 7

Data Structures and Algorithms, Fall 2021

November 22, 2021

Contents

1 Problem 1

1.1 a

$$
\begin{aligned}
Q_k &= \binom{n}{k}\left(\frac{1}{n}\right)^k\left(1-\frac{1}{n}\right)^{n-k} \\
&\le \binom{n}{k}\frac{1}{n^k} \\
&= \frac{n!}{(n-k)!n^k}\frac{1}{k!} \\
&= \frac{n(n-1)\cdots(n-k+1)}{n^k}\frac{1}{k!} \\
&\le \frac{1}{k!} \\
&\le \frac{e^k}{k^k} \quad \left(\text{Stirling's approximation:} k! \ge \left(\frac{k}{e}\right)^k\right)
\end{aligned}
$$

1.2 b

Let $q_k^{(i)}$ denote the event that exactly k keys hash to slot i and no other slot has more keys than it. Obviously, $\Pr[q_k^{(i)}] \le Q_k$

$$
\begin{aligned}
P_k &= \Pr[\bigcup_{i=1}^{n} q_k^{(i)}] \\
&\le \sum_{i=1}^{n} \Pr[q_k^{(i)}] \quad \text{(Union bound)} \\
&\le nQ_k
\end{aligned}
$$

1.3 c

From (a) and (b), we only need

$$\left(\frac{e}{k}\right)^k < \frac{1}{n^3}$$

that is,

$$k(\lg e - \lg k) < -3\lg n$$

substitute $k \ge (c\lg n)/\lg\lg n$,

$$3 < c\left(\frac{\lg\lg n - \lg\lg\lg n + \lg c - \lg e}{\lg\lg n}\right)$$

for sufficient large n, $\left(\frac{\lg\lg n - \lg\lg\lg n + \lg c - \lg e}{\lg\lg n}\right) \to 1$. So $c = 3$ is enough.

1.4 d

Let $k_0 = c \lg n / \lg \lg n$

$$
\begin{aligned}
E[M] &= \sum_{k=0}^{n} k \times \Pr\{M = k\} \\
&= \sum_{k=0}^{k_0} k \times \Pr\{M = k\} + \sum_{k=k_0+1}^{n} k \times \Pr\{M = k\} \\
&\leq \sum_{k=0}^{k_0} k_0 \times \Pr\{M = k\} + \sum_{k=k_0+1}^{n} n \times \Pr\{M = k\} \\
&= k_0 \sum_{k=0}^{k_0} \Pr\{M = k\} + n \sum_{k=k_0+1}^{n} \Pr\{M = k\} \\
&\leq k_0 \times 1 + n \times \sum_{k=k_0+1}^{n} \frac{1}{n^2} \\
&\leq k_0 + 1 \\
&= O(\frac{\lg n}{\lg \lg n})
\end{aligned}
$$

2 Problem 2

(a) We will show that each string hashes to the sum of it's digits $\mod 2^p - 1$. We will do this by induction on the length of the string. As a base case, suppose the string is a single character, then the value of that character is the value of k which is then taken $\mod m$. For an inductive step, let $w = w_1 w_2$ where $|w_1| \geq 1$ and $|w_2| = 1$. Suppose $h(w_1) = k_1$. Then, $h(w) = h(w_1)2^p + h(w_2) \mod 2^p - 1 = h(w_1) + h(w_2) \mod 2^p - 1$. So, since $h(w_1)$ was the sum of all but the last digit $\mod m$, and we are adding the last digit $\mod m$, we have the desired conclusion.

(b) let $h_2(k) = x \cdot d$, $m = y \cdot d$, and x, y are co-prime. Let we prove that y is cycle period for $h(k) \pmod m$.

$$
\begin{aligned}
h(k, y) &= h_1(k) + y \cdot h_2(k) \pmod m \\
&= h_1(k) + y \cdot d \cdot x \pmod m \\
&\equiv h_1(k) \pmod{m = y \cdot d} \\
&= h(k, 0) \pmod m
\end{aligned}
$$

Because x, y are co-prime, so $x \cdot y \cdot d$ is the least common multiple of $h_2(k) = x \cdot d$ and $m = y \cdot d$, so the smaller cycle period is impossible.

So, we should search for $y = \frac{m}{d}$ steps, and $d = 1$ is a special case that we need to search the whole hash table.

3 Problem 3

Assume $m \geq n$, as otherwise, finding a perfect hash function is impossible.

(a)

$\mathbb{E}[\text{\# collisions}] = \binom{n}{2} \cdot \frac{1}{m} = \frac{n(n-1)}{2m}$

(b)

The exact probability that a random hash function is perfect is $\frac{\binom{m}{n} \cdot n!}{m^n}$. For convenience, we define $p = \frac{\binom{m}{n} \cdot n!}{m^n}$.

(c)

Let r.v. X be the number of random hash functions we have to test until we find a perfect hash function. We have $X \sim G(p)$. Then, $\mathbb{E}[X] = \frac{1}{p}$. The exact expected number of random hash functions we have to test before we find a perfect hash function is $\frac{1}{p} - 1$.

(d)

The exact probability that none of the first N random hash functions we try is perfect is $(1-p)^N$.

(e)

Suppose we have to test N ideal random hash functions to find a perfect hash function with high probability. We have $1-(1-p)^N \geq 1-\frac{1}{n}$. We thus have $N \geq \log_{1-p} \frac{1}{n}$. We have to test (at least) $\lceil \log_{1-p} \frac{1}{n} \rceil$ ideal random hash functions to find a perfect hash function with high probability.

4 Problem 4

Note that, the total cost is bounded by

$$\sum_{i=0}^{\lceil \log_2 n \rceil} 2^i + n \leq 5n.$$

5 Problem 5

Let $\Phi : \mathbb{N}^2 \to \mathbb{R}_{\geq 0}$ be the potential function defined as follows:

$$\Phi(\{cur, size\}) = 100 \left| cur - \frac{3 \times size}{8} \right|,$$

where cur denotes the elements in the hash table, and $size$ denotes the size of hash table.

We conclude our proof by checking the following cases.

Case 1: insert x without resizing after insertion

In this case,

$$T_{\textbf{amortized}}(op) = T_{\textbf{actual}}(op) + \Phi(\{cur + 1, size\}) - \Phi(\{cur, size\}) \leq 101.$$

Case 2: insert x with resizing after insertion

In this case,

$$T_{\textbf{amortized}}(op) = T_{\textbf{actual}}(op) + \Phi(\{cur + 1, 2 \times size\}) - \Phi(\{cur, size\}) \leq 101.$$

The last inequality follows from the fact that $\frac{3 \times size}{4} - 1 < cur \leq \frac{3 \times size}{4}$.

Case 3: remove x without resizing after removal

In this case,

$$T_{\textbf{amortized}}(op) = T_{\textbf{actual}}(op) + \Phi(\{cur - 1, size\}) - \Phi(\{cur, size\}) \leq 101.$$

Case 4: remove x with resizing after removal

In this case,

$$\begin{aligned} T_{\textbf{amortized}}(op) &= T_{\textbf{actual}}(op) + \Phi(\{cur - 1, \frac{size}{2}\}) - \Phi(\{cur, size\}) \\ &\leq 1 + 100 \left(\left| cur - 1 - \frac{3 \times size}{16} \right| - \left| cur - \frac{3 \times size}{8} \right| \right) \\ &\overset{(\star)}{\leq} 1 + 100 \left(\frac{size}{16} - \frac{size}{8} + 1 \right) \\ &\leq 101, \end{aligned}$$

where $(\star)$ follows from the fact that $\frac{size}{4} \leq cur < \frac{size}{4} + 1$.