HASH TABLES

1. Introduction

2. Dictionary data type

Definition 2.1. Let $\mathcal{X}$ be a set of *items* and let $\mathcal{U}$ be a set of *keys*. Consider an abstract data type D which dynamically stores a collection of pairs (k, x) where $k \in \mathcal{U}$ and $x \in \mathcal{X}$ in such a way that D does not store two pairs having the same key at the same time. Moreover, we assume that D supports the following operations.

INSERT$\big((k, x)\big)$
Adds pair (k, x) into D if there is no other pair stored in D with k as a first entry.

DELETE$\big(k\big)$
Removes a pair with k as a first entry from D if such pair is stored in D.

SEARCH$\big(k\big)$
Returns x if a pair (k, x) is stored in D. Otherwise returns *nil*.

An abstract data type with these properties and interface is called *an associative array* or *a dictionary*.

Definition 2.2. Let $\mathcal{X}$ and $\mathcal{U}$ be sets. *Dictionary problem for $\mathcal{X}$ and $\mathcal{U}$* is the task of designing a dictionary with $\mathcal{X}$ as the set of items and $\mathcal{U}$ as the set of keys.

3. Hash functions and hash tables with chaining

In this section we introduce the important notion of a hash function and we use it to solve dictionary problem.

Definition 3.1. Let $\mathcal{U}$ be a set. *A hash function* is a mapping $h : \mathcal{U} \to \{0, 1, ..., m-1\}$ where $m \in \mathbb{N}_+$.

Definition 3.2. Let $h : \mathcal{U} \to \{0, 1, ..., m-1\}$ be a hash function. *A collision* is a pair of keys $k_1, k_2 \in \mathcal{U}$ such that $k(k_1) = h(k_2)$.

Using hash functions one can solve dictionary problem. We introduce the notion which describes this solution.

Definition 3.3. Let $\mathcal{U}$ and $\mathcal{X}$ be sets. Let $h : \mathcal{U} \to \{0, 1, ..., m-1\}$ be a hash function for some $m \in \mathbb{N}_+$. We consider an m-element array D_h such that $D_h[l]$ is a linked list storing values from $\mathcal{U} \times \mathcal{X}$ for every $l \in \{0, 1, ..., m-1\}$. We describe dictionary operations.

INSERT$_h\big((k, x)\big)$
Searches for a pair with the first entry k in the list $D_h[h(k)]$. If such pair is found, then replaces its second entry with x. If such pair is not found, then inserts pair (k, x) to the linked list $D_h[h(k)]$ as its new head.

DELETE$_h\big(k\big)$
Deletes a pair with first entry k from the linked list $D_h[h(k)]$.

SEARCH$_h\big(k\big)$
Searches for the pair with the first entry k in the list $D_h[h(k)]$. If such pair is found, then returns its second entry. Otherwise returns *nil*.

Then D_h together with these operations is a solution of dictionary problem for $\mathcal{U}$ and $\mathcal{X}$. We call it *the hash table with collisions resolved by chaining for h.*

4. Analysis of hash tables with chaining for simple uniform hashing

We start by introducing important stochastic property of hash functions.

Definition 4.1. Let $\mathcal{U}$ be a measurable space and let $h : \mathcal{U} \to \{0, 1, ..., m-1\}$ be a measurable hash function for some $m \in \mathbb{N}_+$. Suppose that μ is a probability distribution on $\mathcal{U}$. Fix a probability space $(\Omega, \mathcal{F}, P)$ and a sequence of independent random variables $K_1, ..., K_n : \Omega \to \mathcal{U}$ with distribution μ for some $n \in \mathbb{N}_+$. Consider the following assertions.

(1) Event
$$\mathcal{K} = \left\{\forall_{i,j\in\{1,...,n\},i\neq j}\, K_i \neq K_j\right\}$$
is of positive probability.

(2) Let $K : \Omega \to \mathcal{U}$ be a random variable with distribution μ and independent of $K_1, ..., K_n$. Then
$$P\big(h(K) = h(K_i) \,\big|\, \mathcal{K}\big) = \frac{1}{m}$$
for every $i \in \{1, ..., n\}$.

If assertions above hold for every probability space $(\Omega, \mathcal{F}, P)$, every $n \in \mathbb{N}_+$ and every sequence $K_1, ..., K_n : \Omega \to \mathcal{U}$ of independent random variables with distribution μ, then h is *a simple uniform hashing with respect to μ.*

Now let us give two examples of hash functions satisfying simple uniform hashing with respect to canonical probability distributions on their spaces of keys.

Example 4.2. Let $\mathcal{U} = [0, m]$ for some $m \in \mathbb{N}_+$. Then $\mathcal{U}$ is a measurable space with respect to Borel algebra $\mathcal{B}\big([0, m]\big)$. We define a hash function $h : \mathcal{U} \to \{0, 1, ..., m-1\}$ by formula
$$h(x) = \lfloor x \rfloor$$
Then h is a simple uniform hashing with respect to the normalization of Lebesgue measure on $[0, m]$.

Example 4.3. Let $\mathcal{U} = \{0, 1, ..., m^2 - 1\}$ for some $m \in \mathbb{N}_+$. Then $\mathcal{U}$ is a measurable space with respect to the power algebra $\mathcal{P}\big(\{0, 1, ..., m^2 - 1\}\big)$. Consider $\mathcal{U}$ as a probability space with respect to the uniform distribution μ. We define a hash function $h : \mathcal{U} \to \{0, 1, ..., m-1\}$ by formula
$$h(x) = x \bmod m$$
We verify that h is a simple uniform hashing with respect to μ. Fix a probability space $(\Omega, \mathcal{F}, P)$ and $n \in \mathbb{N}_+$. Suppose first that $K_1, ..., K_n : \Omega \to \mathcal{U}$ are independent random variables with distribution μ. If
$$\mathcal{K} = \left\{\forall_{i,j\in\{1,...,n\},i\neq j}\, K_i \neq K_j\right\}$$
then
$$P(\mathcal{K}) = \frac{m^2 \cdot (m^2 - 1) \cdot ... \cdot (m^2 - n + 1)}{m^{2n}} > 0$$
Next suppose that $K : \Omega \to \mathcal{U}$ is a random variable with distribution μ which is independent of $K_1, ..., K_n$. Then for fixed $i \in \{1, ..., n\}$ we have
$$P\big(h(K) = h(K_i) \,\big|\, \mathcal{K}\big) = \frac{P\big(\{h(K) = h(K_i)\} \cap \mathcal{K}\big)}{P(\mathcal{K})} =$$
$$= m \cdot \frac{m^2 \cdot (m^2 - 1) \cdot ... \cdot (m^2 - n + 1)}{m^{2n+2}} \cdot \left(\frac{m^2 \cdot (m^2 - 1) \cdot ... \cdot (m^2 - n + 1)}{m^{2n}}\right)^{-1} = \frac{1}{m}$$
This completes the verification that h is a simple uniform hashing with respect to μ.

In order to analyze expected costs of dictionary operations for hash tables with chaining we introduce natural probabilistic model.

Setup 4.4 (Probabilistic model for hash tables with chaining)**.** We fix a measurable space of keys $\mathcal{U}$ and a set $\mathcal{X}$ of items. We consider a measurable hash function $h : \mathcal{U} \to \{0, 1, ..., m-1\}$ and a probability distribution μ on $\mathcal{U}$. We also fix a probability space $(\Omega, \mathcal{F}, P)$ and $n \in \mathbb{N}_+$. Let $K_1, ..., K_n : \Omega \to \mathcal{U}$ be independent random variables with distribution μ. Write

$$\mathcal{K} = \left\{\forall_{i,j\in\{1,...,n\},i\neq j}\, K_i \neq K_j\right\}$$

and suppose that $K : \Omega \to \mathcal{U}$ is a random variable with distribution μ and independent from $K_1, ..., K_n$. We assume that pairs with keys $K_1(\omega), ..., K_n(\omega)$ for $\omega \in \mathcal{K}$ were consecutively inserted into initially empty D_h. Under this assumption we denote by $\mathbf{search}_h(K)$ the function $\mathcal{K} \to \mathbb{N}$ which for every $\omega \in \mathcal{K}$ returns the cost (in terms of the number of basic operations) of operation

$$\mathrm{SEARCH}_h(K(\omega))$$

Similarly for

$$\mathrm{DELETE}_h(K(\omega))$$

and (for fixed $x \in \mathcal{X}$)

$$\mathrm{INSERT}_h\left((K(\omega), x)\right)$$

we define functions $\mathbf{delete}_h(K) : \mathcal{K} \to \mathbb{N}$ and $\mathbf{insert}_h\big((K, x)\big) : \mathcal{K} \to \mathbb{N}$.

In the remaining part of this section we work under probabilistic model described in Setup 4.4. We have the following fundamental result.

Theorem 4.5. *Let h be a simple uniform hashing with respect to μ. Then $\mathbf{search}_h(K) : \mathcal{K} \to \mathbb{N}$ is measurable and*

$$\mathbb{E}\,\mathbf{search}_h(K) = \int_{\mathcal{K}} \mathbf{search}_h(K)\, dP_{\mathcal{K}} \leq 1 + \frac{n}{m}$$

Proof. First we introduce certain notation. We consider events

$$W_i = \left\{h\left(K_i\right) = h\left(K\right)\right\},\ Z_i = \left\{K = K_i\right\} \cap \mathcal{K},\ Z = \bigcup_{i=1}^{n} Z_i$$

for $i \in \{1, ..., n\}$. Fix $\omega \in \mathcal{K}$. For

$$\mathrm{SEARCH}_h(K(\omega))$$

we first calculate $h(K(\omega))$. This is a single basic operation. Next if $\omega \in \mathcal{K} \setminus Z$, then we run through the list $D_h[h(K(\omega))]$ with length equal to the number of keys in $\left\{K_1(\omega), ..., K_n(\omega)\right\}$ mapped by h to $h(K(\omega))$. This is the case of the unsuccessful search. Otherwise, if $\omega \in Z_i$ then we run through the initial segment of the list $D_h[h(K(\omega))]$ which consists of elements from the set $\left\{K_i(\omega), K_{i+1}(\omega), ..., K_n(\omega)\right\}$ mapped by h to $h(K(\omega))$. This is the case when the search is successful. Thus

$$\mathbf{search}_h\left(K\right) = \underbrace{1}_{\text{computation of the hash}} + \underbrace{\chi_{\mathcal{K}\setminus Z} \cdot \sum_{i=1}^{n} \chi_{W_i}}_{\text{unsuccessful search}} + \underbrace{\sum_{i=1}^{n} \chi_{Z_i} \cdot \sum_{j=i}^{n} \chi_{W_j}}_{\text{succesfull search}}$$

Hence $\mathbf{search}_h\left(K\right) : \mathcal{K} \to \mathbb{N}$ is measurable. Moreover, note that

$$\mathbf{search}_h\left(K\right) = 1 + \chi_{\mathcal{K}\setminus Z} \cdot \sum_{i=1}^{n} \chi_{W_i} + \sum_{i=1}^{n} \chi_{Z_i} \cdot \sum_{j=i}^{n} \chi_{W_j} \leq 1 + \chi_{\mathcal{K}\setminus Z} \cdot \sum_{i=1}^{n} \chi_{W_i} + \sum_{i=1}^{n} \chi_{Z_i} \cdot \sum_{j=1}^{n} \chi_{W_j} = 1 + \sum_{i=1}^{n} \chi_{W_i}$$

and hence

$$\mathbb{E}\,\mathbf{search}_h\left(K\right) \leq 1 + \sum_{i=1}^{n} \mathbb{E}\,\chi_{W_i} = 1 + \sum_{i=1}^{n} P_{\mathcal{K}}(W_i) = 1 + \sum_{i=1}^{n} \frac{P(W_i \cap \mathcal{K})}{P(\mathcal{K})} = 1 + \frac{n}{m}$$

The last inequality is a consequence of the fact that h is a simple uniform hashing with respect to μ. □

Using essentially the same method (we omit the proof) one derives the following results.

Theorem 4.6. *Let h be a simple uniform hashing with respect to μ. Fix x in $\mathcal{X}$. Then both functions $\mathbf{delete}_h(K), \mathbf{insert}_k((K,x)) : \mathcal{K} \to \mathbb{N}$ are measurable and*

$$\mathbb{E}\,\mathbf{delete}_h(K), \mathbb{E}\,\mathbf{insert}_h((K,x)) \leq 1 + \frac{n}{m}$$

These results have the following consequence.

Corollary 4.7. *Let h be a simple uniform hashing with respect to μ. Suppose that there exists a constant $c \in \mathbb{R}_+$ such that $n \leq c \cdot m$. Then the expected costs of all dictionary operations for D_h are $\mathcal{O}(1)$.*

Proof. The assertion follows from Theorems 4.5 and 4.6 and the inequality $1 + \frac{n}{m} \leq 1 + c$. □

5. Analysis of hash tables with chaining for universal hashing families

6. Universal hash function families

In this section we introduce and study important notion of universal hash function family.

Definition 6.1. Let $\mathcal{U}$ be a set and let $m \in \mathbb{N}_+$. Suppose that $\mathcal{H}$ is a finite family of functions $\mathcal{U} \to \{0, 1, ..., m-1\}$ and let P be a uniform probability distribution on $\mathcal{H}$. Suppose that

$$P\big(h(k_1) = h(k_2)\big) = \frac{1}{m}$$

for every pair of distinct keys $k_1, k_2 \in \mathcal{U}$. Then $\mathcal{H}$ is *a universal family of hash function*.

The remaining part of this section is devoted to giving examples of universal families of hash functions.

Example 6.2. Let p be a prime number. Consider $\mathcal{U} = \mathbb{Z}_p^n$ for some $n \in \mathbb{N}_+$. Fix k in $\mathcal{U}$. Expand

$$k = a_{n-1}(k) \cdot p^{n-1} + ... + a_1(k) \cdot p + a_0(k)$$

where $a_0(k), a_1(k), ..., a_{n-1}(k) \in \{0, 1, ..., p-1\}$. Then the map

$$\mathcal{U} \ni k \mapsto (a_0(k), a_1(k), ..., a_{n-1}(k)) \in \mathbb{Z}_p^n$$

is bijective. Now for fixed $v = (v_0, ..., v_n) \in \mathbb{Z}_p^n$ we define $h_v : \mathcal{U} \to \{0, 1, ..., p-1\}$ by formula

$$h_v(k) = \sum_{i=0}^{n-1} a_i(k) \cdot v_i \bmod p$$