Data struktures II

NTIN067

Jirka Fink

https://ktiml.mff.cuni.cz/~fink/

Katedra teoretické informatiky a matematické logiky
Matematicko-fyzikální fakulta
Univerzita Karlova v Praze

Summer semester 2016/17

Last change February 21, 2017

Content

General information

E-mail	fink@ktiml.mff.cuni.cz
Homepage	http://ktiml.mff.cuni.cz/˜fink/
Consultations	Individual schedule

Content

Computational model Word-RAM

Description

- A word is a w-bit integer
- A memory an array of words indexed by words
- The size of memory is 2^w, so we assume that $w = \Omega(\log n)$
- Operations of words in constant time:
  - Arithmetical operations are $+, -, \star, /, \mod$
  - Bit-wise operations &, $|$,^, $>>$, $<<$
  - Comparisons $=, <, \leq, >, \geq$
- Other operations in constant time: (un)conditional jumps, assignments, memory accesses, etc.
- Inputs and outputs are stored in memory

Static dictionaries

Notations

- Universe U of all elements (words)
- Store $S \subseteq U$ of size n in a data structure
- Using hashing, we store S in a table $M = [m] = \{0, \ldots, m-1\}$ of size m

Goal

Create a data structure determining whether a given element of U belongs to S.

Methods

	Build	Member	
Search tree	$n \log n$	$\log n$	optimal in the comparison model
Cuckoo	n (exp.)	1	$\log n$-independent
FKS	n (exp.)	1	2-independent
	$n \log n$	1	deterministic

Independent repeated trials

Markov inequality

If X is an independent non-negative random variable and $c > 1$, then $P[X < cE[X]] > \frac{c-1}{c}$.

Expected number of trial using probability

Let V be an event that occurs in a trial with probability p. The expected number of trials to first occurrence of V in a sequence of independent trials is $\frac{1}{p}$.

Expected number of trial using mean

If X is an independent non-negative random variable and $c > 1$. The expected number of trials to first occurrence of $X \leq cE[X]$ in a sequence of independent trials is $\frac{c}{c-1}$.

Example

If the expected number of collisions of a randomly chosen hashing function h is k, then the expected number of independent trials to the first occurrence of a hashing function h with at most $2k$ collisions is 2.

Universal hashing systems

c-universal hashing system

A hashing system $\mathcal{H}$ of functions $h : U \to M$ is c-universal for $c > 1$ if a uniformly chosen h from $\mathcal{H}$ satifies $P[h(x) = h(y)] \leq \frac{c}{m}$ for every $x, y \in U$ and $x \neq y$.

k-independent hashing system

A hashing system $\mathcal{H}$ of functions $h : U \to M$ is k-independent for $k \in \mathbb{N}$ if a uniformly chosen h from $\mathcal{H}$ satifies $P[h(x_i) = z_i \text{ for all } i = 1, \ldots, k] = \mathcal{O}\left(\frac{1}{m^k}\right)$ for all pairwise different $x_1, \ldots, x_k \in U$ and all $z_1, \ldots, z_k \in M$.

Example: System Multiply-mod-prime

- Let p be a prime greater than u
- $h_{a,b}(x) = (ax + b \bmod p) \bmod m$
- $\mathcal{H} = \{h_{a,b};\ a, b \in [p], a \neq 0\}$
- System $\mathcal{H}$ is 1-universal and 2-independent but it is not 3-independent

Hagerup, Miltersen, Pagh, 2001 [1]

Goal

A static dictionary for n w-bit keys with constant lookup time and a space consumption of $\mathcal{O}(n)$ words can be constructed in $\mathcal{O}(n \log n)$ time on w-word-RAM.
The algorithm is weakly non-uniform, i.e. requires certain precomputed constants dependent on w.

Overview

1. Create a function $f_1 : [2^w] \to [2^{4w}]$ which is an error-correcting code of relative minimum distance $\delta > 0$.
2. Create a function $f_2 : [2^{4w}] \to [\mathcal{O}(n^k)]$ which is an injection on $f_1(S)$
3. Create a function $f_3 : [\mathcal{O}(n^k)] \to [\mathcal{O}(n^2)]$ which is an injection on $f_2(f_1(S))$
4. Create a function $f_4 : [\mathcal{O}(n^2)] \to [\mathcal{O}(n)]$ which is an injection on $f_3(f_2(f_1(S)))$

- $f_4 \circ f_3 \circ f_2 \circ f_1$ can be computed in constant time
- f_2, f_3, f_4 can be fount in time $\mathcal{O}(n \log n)$
- f_1 can be precomputed in time $\mathcal{O}(w)$

Static dictionaries: $[\mathcal{O}(n^2)] \to [\mathcal{O}(n)]$

Goal

Find a function $h : U \to [2^r]$ with $U = [\mathcal{O}(n^2)]$ and $r = \max\left\{\frac{w}{2}, 3 + \log n\right\}$ s.t.

- h is perfect on $S \subseteq U$ of size n and
- h can be computed in constant time and
- space consumption $\mathcal{O}(n)$ for finding and storing h and
- h can be fount in $\mathcal{O}(n \log n)$ worst case time.
  - First, expected $\mathcal{O}(n)$ time,
  - then derandomize to $\mathcal{O}(n \log n)$ worst case time.

$x \in U$ is a point $(f(x), g(x))$ in a $(\mathcal{O}(n) \times \mathcal{O}(n))$-table

For $x \in U$, let $f(x)$ denote the first r bits of x and $g(x)$ denotes the remaining bits. ①
Then $x \mapsto (f(x), g(x))$ is an injection (e.i. perfect on U). ②

1. The number of remaining bits is at most r.
2. Since $r \geq \frac{w}{2}$.

Static dictionaries: $[\mathcal{O}(n^2)] \to [\mathcal{O}(n)]$

Definition

For $q \geq 0$ and functions $f, g : U \to [2^r]$, the pair (f, g) is q-good if

- f has at most q collisions and
- $x \mapsto (f(x), g(x))$ is perfect on S.

The number of collisions is the number of pairs $\{x, y\} \subseteq S$ such that $f(x) = f(y)$.

Lemma

Suppose that (f, g) is q-good and $r \geq 3 + \log n$. Then, for every $v \in [2^r]$ there exists $a_v \in [2^r]$ such that $(x \mapsto g(x) \oplus a_{f(x)}, f)$ is q'-good where $q' = \begin{cases} 0 & \text{if } q \leq n \\ n & \text{otherwise.} \end{cases}$

All values a_v can be computed in expected time $\mathcal{O}(n)$ and space $\mathcal{O}(n)$ worst case.

Application: Randomized construction of a mapping $[\mathcal{O}(n^2)] \to [\mathcal{O}(n)]$

0. (f, g) is $\binom{n}{2}$-good
1. $(x \mapsto g(x) \oplus a_{f(x)}, f) = (f', g')$ is n-good
2. $(x \mapsto g'(x) \oplus a'_{f'(x)}, f') = (f'', g'')$ is 0-good, so f'' is perfect

Static dictionaries: $[\mathcal{O}(n^2)] \to [\mathcal{O}(n)]$

Lemma

Suppose that (f, g) is q-good and $r \geq 3 + \log n$. Then, for every $v \in [2^r]$ there exists $a_v \in [2^r]$ such that $(x \mapsto g(x) \oplus a_{f(x)}, f)$ is q'-good where $q' = \begin{cases} 0 & \text{if } q \leq n \\ n & \text{otherwise.} \end{cases}$

All values a_v can be computed in expected time $\mathcal{O}(n)$ and space $\mathcal{O}(n)$ worst case.

Proof ($q' \leq n$)

1. Let $h(x) = g(x) \oplus a_{f(x)}$
2. If $x, y \in S$ and $x \neq y$ and $f(x) = f(y)$, then $g(x) \neq g(y)$ and $h(x) \neq h(y)$ ①
3. If $f(x) \neq f(y)$, then $P[h(x) = h(y)] = \frac{1}{2^r}$ where $a_v \sim U[2^r]$ independently for all $v \in [2^r]$ ②
4. $E[|\{\{x, y\} \subseteq S;\ h(x) = h(y)\}|] \leq \binom{n}{2}/2^r < \frac{n}{16}$ ③
5. The expected number of trials to generate h with at most n collisions is $\mathcal{O}(1)$.

1. Since $x \mapsto (f(x), g(x))$ is perfect on S, $g(x) \neq g(y)$. From $g(x) \oplus a_{f(x)} = g(x) \oplus a_{f(y)} \neq g(y) \oplus a_{f(y)}$ it follows that $h(x) \neq h(y)$.
2. For every $v \in [2^r]$ we randomly and independently choose a_v from the uniform distribution on $[2^r]$. Then,

$$\begin{aligned} h(x) &= h(y) \\ g(x) \oplus a_{f(x)} &= g(y) \oplus a_{f(y)} \\ a_{f(x)} &= g(x) \oplus g(y) \oplus a_{f(y)} \end{aligned}$$

   Since $([2^r], \oplus)$ is an Abelian group, $b \mapsto b \oplus c$ is a bijection on $[2^r]$ for every $c \in [2^r]$ and so $a_{f(y)} \sim U[2^r]$, it follows that $g(x) \oplus g(y) \oplus a_{f(y)} \sim U[2^r]$. Since $a_{f(x)}$ and $a_{f(y)}$ are independent, also $a_{f(x)}$ and $g(x) \oplus g(y) \oplus a_{f(y)}$ are independent. Hence, $P[h(x) = h(y)] = \frac{1}{2^r}$.
3. Use the linearity of expectation and substitute r.

Lemma

Suppose that (f, g) is q-good and $r \geq 3 + \log n$. Then, for every $v \in [2^r]$ there exists $a_v \in [2^r]$ such that $(x \mapsto g(x) \oplus a_{f(x)}, f)$ is q'-good where $q' = \begin{cases} 0 & \text{if } q \leq n \\ n & \text{otherwise.} \end{cases}$

All values a_v can be computed in expected time $\mathcal{O}(n)$ and space $\mathcal{O}(n)$ worst case.

Proof ($q \leq n$ implies $q' = 0$)

1. Let $S_v = \{x \in S;\ f(x) = v\}$
2. Order S_v by non-increasing size, i.e. $|S_{v_1}| \geq |S_{v_2}| \geq \ldots \geq |S_{v_{2^r}}|$
3. For $j = 1, \ldots, 2^r$ we find a_{v_j} such that h is perfect ①
4. For $a_{v_j} \sim U[2^r]$ it holds $E[|\left\{(x, y) \in S_{v_j} \times S_{<j};\ h(x) = h(y)\right\}|]$
   $\leq |S_{v_j}||S_{<j}|P[h(x) = h(y)]$ ②
   $\leq \sum_{i=1}^{j-1} |S_{v_j}||S_{v_i}|/2^r$ ③
   $\leq \sum_{i=1}^{j-1} |S^2_{v_i}|/2^r$ ④
   $\leq \sum_{i=1}^{j-1} \binom{S_{v_i}}{2}/2^{r-2}$ ⑤
   $\leq q/2^{r-2} \leq \frac{1}{2}$
5. The expected number of trials to generate a_{v_j} such that h has no collision is $\mathcal{O}(1)$.
   ⑥ ⑦

1. Note that we must find h without collisions. To be precise, we iteratively find a_{v_j} for j from 1 to 2^r such that it holds $h(x) \neq h(y)$ for every $x \in S_{v_j}$ and $y \in S_{<j}$ where $S_{<j} = \bigcup_{i=1}^{j} S_{v_i}$.
2. Linearity of expectation
3. Definition of $S_{<j}$
4. $|S_{v_i}| \geq |S_{v_j}|$
5. From this point, assume that $|S_{v_j}| \geq 2$.
6. In order to verify that h has no collision, we use a counter $m_v = |\{y \in S_{<j};\ h(y) = v\}|$. For every j we can count the collisions and update m_v in time $\mathcal{O}(|S_j|)$. The expected time to find all a_{v_j} is $\sum_j \mathcal{O}(|S_j|) = \mathcal{O}(n)$.
7. For $S_{v_j} = \{x\}$ we can find v with $m_v = 0$ and set $a_{v_j} = v \oplus g(x)$.

Static dictionaries: $[\mathcal{O}(n^2)] \to [\mathcal{O}(n)]$

Derandomization

Let $L_k(a) = |\{(x,y) \in S_{v_j} \times S_{<j};\ (g(x) \oplus a)_{[k]} = (h(y))_{[k]}\}|$ ①

$(a)_i$ denotes the i-th bit of a and $(a)_M$ denotes the vector of all bits $(a)_i$ for $i \in M \subseteq [r]$ ②

for $j \leftarrow 1$ **to** 2^r **do**
    $a_{v_j} \leftarrow 0$
    for $k \leftarrow 0$ **to** $r-1$ **do**
        if $L_k(a_{v_j}) > L_k(a_{v_j} + 2^k)$ **then**
            $a_{v_j} \leftarrow a_{v_j} + 2^k$

Proof (goodness of (h, f))

- $L_k(a) + L_k(a \oplus 2^k) = L_{k-1}(a)$ for every $a \in [2^k]$ and $k \in [r]$
- $L_k(a_{v_j}) \le \frac{L_{k-1}(a_{v_j})}{2} \le \frac{L_0(a_{v_j})}{2^k} = \frac{|S_{v_j}||S_{<j}|}{2^k}$
- The total number of collision is at most $\sum_j L_r(a_{v_j}) \le \sum_j 2^{-r}|S_{v_j}||S_{<j}| \le \sum_{i<j} 2^{-r}|S_{v_j}||S_{v_i}| \le 2^{-r-1}\left(\sum_i S_{v_i}\right)^2 < \frac{n}{16}$
- If $q \le n$, then the number of collision with S_{v_j} is $L_r(a_{v_j}) \le \sum_{i<j} 2^{-r}|S_{v_j}||S_{v_i}| \le \sum_{i<j} 2^{2-r}\binom{S_{v_j}}{2} \le 2^{2-r}q \le \frac{1}{2}$

1. Where $k \in [r]$ and $a \in [2^k]$
2. Our goal is to iteratively and deterministically compute a_{v_j} for j from 1 to 2^r. The value of a_{v_k} is computed by bits from the least significant to the most significant bit. $L_k(a)$ determines the number of collision between S_{v_j} and $S_{<j}$ if we consider only last k bits.

Static dictionaries: $[\mathcal{O}(n^2)] \to [\mathcal{O}(n)]$

Derandomization

Let $L_k(a) = \left|\left\{(x,y) \in S_{v_j} \times S_{<j};\ (g(x) \oplus a)_{[k]} = (h(y))_{[k]}\right\}\right|$
$(a)_i$ denotes the i-th bit of a and $(a)_M$ denotes the vector of all bits $(a)_i$ for $i \in M \subseteq [r]$

for $j \leftarrow 1$ **to** 2^r **do**
  $a_{v_j} \leftarrow 0$
  for $k \leftarrow 0$ **to** $r-1$ **do**
    if $L_k(a_{v_j}) > L_k(a_{v_j+2^k})$ **then**
      $a_{v_j} \leftarrow a_{v_j} + 2^k$

Proof (Complexity)

- In order to compute $L_k(a)$, we build a binary tree (trie)
- Every vertex $a \in [2^k]$ of the k-th level has a counter $c_k(a) = \left|\left\{y \in S_{<j};\ (h(y))_{[k]} = (a)_{[k]}\right\}\right|$
- $L_k(a) = \sum_{x \in S_{v_j}} c_k(g(x) \oplus a)$ can be computed in $\mathcal{O}(|S_{v_j}|)$ time
- After the j-th step, counters can be updated in $\mathcal{O}(|S_{v_j}|r)$ time
- Total time is $\sum_j |S_{v_j}|r = \mathcal{O}(n \log n)$

Static dictionaries: $[\mathcal{O}(n^k)] \to [\mathcal{O}(n^2)]$

Approach

- Every $x \in U = [\mathcal{O}(n^k)]$ can be regarded as constant-length string over an alphabet of size n
- Build n-way branching compressed trie of string S
- The number of leaves is $|S| = n$, so the total number of vertices is at most kn
- Build static $[\mathcal{O}(n^2)] \to [\mathcal{O}(n)]$ dictionary for pairs (vertex of the trie, letter) which returns a child of the vertex
- One polynomial-size-universe lookup is evaluated using a constant number of quadratic-size-universe lookups
- Space complexity is $\mathcal{O}(n)$ and this dictionary is constructed in $\mathcal{O}(n \log n)$ time

Static dictionaries: Error-correcting code

Definition

- The Hamming distance between $x \in [2^w]$ and $y \in [2^w]$ is the number of bits in which x and y differ.
- $\psi : [2^w] \to [2^{4w}]$ is an error correcting code of relative minimum distance $\delta > 0$ if the Hamming distance between $\psi(x)$ and $\psi(y)$ is at least $4w\delta$ for every distinct $x, y \in [2^w]$.

Lemma

Let $\mathcal{H}$ be a 2-universal hashing system of function $[2^w] \to [2^{4w}]$. For every δ with $1/4w < \delta \leq 1/2$, the probability that $h \sim U(\mathcal{H})$ is an error correcting code of relative minimum distance $\delta > 0$ is at least $1 - \left(\left(\frac{e}{\delta}\right)^{4\delta}/4\right)^w$. ①

Proof

- For $x \in [2^{4w}]$ the number of y within Hamming distance k is at most $\left(\frac{4ew}{k}\right)^k$. ②
- For $x \neq y$, $P(\text{Hamming distance between } x \text{ and } y \ \leq k) \leq 2^{1-4w}\left(\frac{4ew}{k}\right)^k$
- The probability that this happens for any of the $\binom{2^w}{2} < 2^{2w-1}$ pairs is at most $\left(\left(\frac{e}{\delta}\right)^{4\delta}/4\right)^w$ ③

1. For $\delta < \frac{1}{4w}$ it holds that $4w\delta < 1$ and the identity is an error correcting code of relative minimum distance δ.
2. The number of $y \in [2^{4w}]$ within Hamming distance $k \geq 1$ from a fixed $x \in [2^{4w}]$ is $\sum_{i=0}^{k} \binom{4w}{i} \leq (\frac{4w}{k})^k \sum_{i=0}^{k} \binom{4w}{i} (\frac{k}{4w})^i \leq (\frac{4w}{k})^k (1 + \frac{k}{4w})^{4w} \leq (\frac{4w}{k})^k e^k \leq (\frac{4ew}{k})^k$ using the binomial theorem.
3. By setting $k = \lfloor 4w\delta \rfloor$ we obtain $2^{2w-1} 2^{1-4w} (\frac{4ew}{k})^k \leq 2^{-2w} (\frac{4ew}{4w\delta})^{4w\delta} = (2^{-2} (\frac{e}{\delta})^{4\delta})^w$

Static dictionaries: $[2^w] \to [\mathcal{O}(n^k)]$

Lemma

Let $\psi : [2^w] \to [2^{4w}]$ be an error correcting code of relative minimum distance $\delta > 0$ and $S \subseteq U = [2^w]$ of size n. There exists a set $D \subseteq [4w]$ with $|D| \leq 2 \log n / \log \frac{1}{1-\delta}$ such that for every pair x, y of distinct elements of S it holds $(\psi(x))_D \neq (\psi(y))_D$.

Proof

- For $D \subseteq [4w]$ and $v \in [2^{|D|}]$ let $C(D, v) = \{x \in S;\ (\psi(x))_D = v\}$ ①
- The set of colliding pairs of D is $B(D) = \bigcup_{v \in [2^{|D|}]} \binom{C(D,v)}{2}$
- We construct $D_0 \subseteq D_1 \subseteq \ldots \subseteq D_k$ such that $|D_i| = i$ and $|B(D_i)| < (1-\delta)^i n^2/2$ ②
- Let $I(d) = \{\{x, y\} \in B(D_i);\ (\psi(x))_d = (\psi(y))_d\}$ be the colliding pairs indistinguishable by $d \in [4w] \setminus D_i$
- Let $I = \sum_{d \in [4w] \setminus D_i} |I(d)|$
- Every pair $\{x, y\} \in B(D_i)$ contributes to I by at most $4w - i - 4w\delta < 4w(1-\delta)$, so $I \leq 4w(1-\delta)|B(D_i)|$
- By averaging, there exists $d \in [4w] \setminus D_i$ such that $|I(d)| \leq \frac{I}{4w-i} \leq (1-\delta)|B(D_i)|$ ③
- Let $D_{i+1} = D_i \cup \{d\}$. Hence, $|B(D_{i+1})| = |I(d)| \leq (1-\delta)|B(D_i)|$
- By setting $k = \left\lfloor 2 \log n / \log \frac{1}{1-\delta} \right\rfloor$ we obtain $|B(D_k)| < 1$.

1. Note that for every $D \subseteq [4w]$ the set S is split into $2^{|D|}$ disjoint clusters $C(D, v)$ for $v \in [2^{|D|}]$.
2. For $i = 0$ it holds that $D_0 = \emptyset$ and $B(D_0) = \binom{n}{2} < \frac{n^2}{2}$.
3. A bit $d \in [4w] \setminus D_i$ with $|I(d)| \leq (1 - \delta)|B(D_i)|$ can be found in $\mathcal{O}(wn)$ time as follows. We a list of all clusters $C(D_i, v)$ of size at least two. Every cluster has a list of all elements. So, $I(d)$ for one $d \in [4w] \setminus D_i$ can be determined in $\mathcal{O}(n)$ time and we can process all d in $\mathcal{O}(wn)$ time. Then, all lists can be updated in $\mathcal{O}(n)$ time. Using word-level parallelism, the time complexity can be improved to $\mathcal{O}(n)$.

Content

[1] Torben Hagerup, Peter Bro Miltersen, and Rasmus Pagh.
Deterministic dictionaries.
Journal of Algorithms, 41(1):69–85, 2001.