Extended Algorithms courses COMP3821/9801

Aleks Ignjatović

School of Computer Science and Engineering
University of New South Wales

Introduction to Randomized Algorithms: Randomized Hashing

Hash Functions

Scenario:

- You are given an assignment to implement hashing;
- You will self-grade in pairs, testing and grading your partners implementation;
- Your partner plays dirty:
  - he analyses your hash function;
  - picks a sequence of the worst-case keys, causing your implementation to take O(n) time to search.

- **What would you do?**

Hash functions: randomised hashing

Solution:

- Randomise your hashing;
- Pick a hash function randomly in a way that is independent of the keys that are actually going to be stored.
- In this way no single input always evokes worst case performance!
- Guarantees good performance **on average** over many runs of your program, no matter what keys adversary chooses.

Towards randomised hashing: universal families of hash functions

- Let H be a (finite) collection of hash functions that map a given universe U of keys into the (much smaller) range $\{0, 1, , m - 1\}$;
- H is said to be **universal** if:
  - for each pair of distinct keys $x, y \in U$, the number of hash functions $h \in H$ for which $h(x) = h(y)$ is $|H|/m$.
- In other words: any two keys x and y, if you randomly pick a hash function from H, the chance of a collision between x and y is the same, equal to $1/m$.

Universal hashing

- Assume a family of hash functions H is universal.
  - Let $y, z \in U$ be arbitrary keys. For a randomly chosen $h \in H$ let the random variable c_{yz} be defined by $c_{yz} = 1$ if the keys y and z collide under h, i.e., $h(y) = h(z)$, and $c_{yz} = 0$ otherwise.
  - Fix x; then, by definition of a universal family, the expected value $E[c_{yx}]$ satisfies

$$\begin{aligned} E[c_{yx}] &= P\left(h(y) = h(x)\right) \cdot 1 + P\left(h(y) \neq h(x)\right) \cdot 0 \\ &= \frac{1}{m} \cdot 1 + \left(1 - \frac{1}{m}\right) \cdot 0 \\ &= \frac{1}{m} \end{aligned}$$

Universal hashing

- Assuming that family H is universal, and assuming that we are hashing n keys into a hash table of size m,
- let C_x be total number of collisions involving key x; then

$$C_x = \sum_{y \neq x} c_{yx}$$

- Then the expected value $E[C_x]$ satisfies

$$E[C_x] = \sum_{y \neq x} E[c_{yx}] = \frac{n-1}{m} \tag{1}$$

- Thus, if $n \leq m$ then the expected total number of collisions involving any particular key x is less than 1!

Universal hashing family main property:

If we:

- choose randomly a hash function h from a universal family of hash functions H;
- hash n keys into a hash table of size m,

then:

- the expected number of keys in each slot is $\alpha = n/m$;
- thus, if $n \leq m$ then the expected total number of collisions involving any particular key x is $\frac{n-1}{m} < 1$.

Designing a universal family of hash Functions

- Choose the size m of the hash table to be a prime number;
- let r be such that the size $|U|$ of the universe U of all keys satisfies $m^r \leq |U| < m^{r+1}$ (i.e. $r = \lfloor \log_m |U| \rfloor$);
- represent each key x in base m, i.e., let $x_0, x_1, \ldots, x_r$ be such that $0 \leq x_i < m$ for all i such that $0 \leq i \leq r$ and such that

$$x = \sum_{i=0}^{r} x_i\, m^i$$

- let $\vec{a} = \langle a_0, a_1, \ldots, a_r \rangle$ be a sequence of $r + 1$ **randomly chosen** elements from the set $\{0, 1, \ldots, m - 1\}$;
- define corresponding hash function $h_{\vec{a}}(x) = \left(\sum_{i=0}^{r} x_i a_i \right) \pmod{m};$

Proving universality of family of hash functions $h_{\vec{a}}$

- Assume x, y are two distinct keys;
- let the corresponding sequences be $\langle x_0, x_1, \ldots, x_r \rangle$ and $\langle y_0, y_1, \ldots, y_r \rangle$;
- then

$$\begin{aligned} h_{\vec{a}}(x) = h_{\vec{a}}(y) \quad & \Leftrightarrow \quad \sum_{i=0}^{r} x_i a_i = \sum_{i=0}^{r} y_i a_i \pmod{m} \\ & \Leftrightarrow \quad \sum_{i=0}^{r} (x_i - y_i) a_i = 0 \pmod{m} \end{aligned}$$

- since $x \neq y$ there exists $k \leq r$ such that $x_k \neq y_k$;
- let us assume that $x_0 \neq y_0$;
- then $(x_0 - y_0) a_0 = -\sum_{i=1}^{r} (x_i - y_i) a_i \pmod{m}$

Universality of family of hash functions $h_{\vec{a}}$ (continued)

- Since m is a prime, every non-zero element $z \in \{0, 1, \ldots, m-1\}$ has a multiplicative inverse z^{-1}, such that $z \cdot z^{-1} = 1 \pmod{m}$;
- since $x_0 - y_0 \neq 0$ we have that

$$(x_0 - y_0)a_0 = -\sum_{i=1}^{r}(x_i - y_i)a_i \pmod{m}$$

implies

$$a_0 = \left(-\sum_{i=1}^{r}(x_i - y_i)a_i\right)(x_0 - y_0)^{-1} \pmod{m}$$

Universality of family of hash functions $h_{\vec{a}}$ (continued)

- However, $a_0 = \left(-\sum_{i=1}^{r} (x_i - y_i) a_i \right) (x_0 - y_0)^{-1} \pmod{m}$

  implies that

  - for any two keys x, y such that $x_0 \neq y_0$ and
  - for any randomly chosen r numbers $a_{\mathbf{1}}, a_2, \ldots, a_r$

  there exists **exactly one** a_0 (the one given by the above equation) such that for $\vec{a} = \langle a_0, a_1, \ldots, a_r \rangle$ we have

$$h_{\vec{a}}(x) = h_{\vec{a}}(y)$$

- Since there are:
  - m^r sequences of the form $\langle a_{\mathbf{1}}, \ldots, a_r \rangle$, each of which can uniquely be extended to a sequence $\vec{a} = \langle a_0, a_1, \ldots, a_r \rangle$ such that $h_{\vec{a}}(x) = h_{\vec{a}}(y)$
  - and m^{r+1} sequences of the form $\vec{a} = \langle a_0, a_1, \ldots, a_r \rangle$ in total,

  we conclude that the probability to randomly chose a sequence $\vec{a} = \langle a_0, a_1, \ldots, a_r \rangle$ such that $h_{\vec{a}}(x) = h_{\vec{a}}(y)$, i.e., such that x and y collide, is equal to

$$\frac{m^r}{m^{r+1}} = \frac{1}{m}$$

- Thus, the family H is a universal collection of hash functions.

Using universal family of hash functions $h_{\vec{a}}$:

- Pick $r = \lfloor \log_m |U| \rfloor$, so that $m^r \leq |U| < m^{r+1}$;
- For each run, pick a hash function by randomly picking the vector $\vec{a} = \langle a_0, a_1, \ldots, a_r \rangle$ such that $0 \leq a_i < m$ for all i s.t. $0 \leq i \leq r$;
- during each run, use that function on all keys

Note that

$$h_{\vec{a}}(x) = \left(\sum_{i=0}^{r} x_i a_i \right) \pmod{m} = \langle x, y \rangle \pmod{m};$$

Scalar product $\langle x, y \rangle$ can be computed very fast on modern hardware.