Extended Algorithms Courses COMP3821/9801

Aleks Ignjatović

School of Computer Science and Engineering
University of New South Wales

Introduction to Randomized Algorithms:
Perfect Hashing

Perfect Hashing for Static Tables

Problem:

- Assume that you need a **static hash table** to store n keys (i.e., a look-up table; no insertions or deletions, just search);
- the size of the table should be linear in n;
- the table should be completely collision free.
- the corresponding hash function should be very efficient to compute.

To get such a table we will employ a randomised design method; however, the resulting hash function will be completely deterministic.

Important design tool: Markov Inequality

- Assume $X > 0$ is a non-negative random variable;
- assume also that $t > 0$ is any positive real number;

- then:

$$P\{X \geq t\} \leq \frac{E[X]}{t}$$

- **Proof:** Essentially, we take into account only events when $X \geq t$ and ignore events when $X < t$:
- If X is discrete, then

$$E[X] = \sum_{v} P\{X = v\} \cdot v \geq \sum_{v \geq t} P\{X = v\} \cdot v \qquad (1)$$

$$\geq \sum_{v \geq t} P\{X = v\} \cdot t = t\, P\{X \geq t\} \qquad (2)$$

- Divide now both sides by $t > 0$ to obtain the Markov Inequality.
- The case when X is continuous is essentially identical.

Designing a Perfect Hash table

Method: trial and error procedure with very low probability of many consecutive failures.

First step:

- given n keys we will be constructing tables of size $< 2n^2$ using universal hashing;
- probability that such a table is collision free will be $> 1/2$.
- How do we accomplish this? We use a randomised design procedure:
    - we pick the least prime m such that $m \geq n^2$; then $m < 2n^2$ (for every $x > 1$ there exists a prime m such that $x \leq m < 2x$)
    - we pick a random vector $\vec{a}$ and hash all keys using the corresponding hash function $h_{\vec{a}}$ from the universal family;

Designing a Perfect Hash table (continued)

- Given n keys, there will be $\binom{n}{2}$ pairs of keys.
- By universality of the family of hash functions used, for each par of keys probability of a collision is $\frac{1}{m}$.
- Since $m \geq n^2$ we have $\frac{1}{m} \leq \frac{1}{n^2}$.
- Thus, the expected total number of collisions in the table is at most

$$\binom{n}{2}\frac{1}{m} \leq \frac{n(n-1)}{2}\frac{1}{n^2} < \frac{1}{2}$$

Designing a Perfect Hash table (continued)

- For the given n keys, we have constructed a table of size $m < 2n^2$, such that the expected total number of collisions is $< 1/2$.
- By the Markov Inequality with $t = 1$ we now get that

$$P\{X \geq 1\} \leq \frac{E[X]}{1} < \frac{1}{2}$$

- Thus, if we keep picking hash functions at random from a universal family, the the probability that there will be at least one collision in each of k consecutive attempts (i.e., that $X \geq 1$ in each attempt) is smaller than $(1/2)^k$, which rapidly tends to 0.
- Consequently, after a few random trial-and-error attempts we will obtain a collision free hash table of size $< 2n^2$.

Designing a Perfect Hash table (continued)

- How many trials N do we expect to have to make before we hit a collision free hash table of size $< 2n^2$?
- Let the probability of failure be denoted by p; then $p < 1/2$, and the probability of a success is $1 - p$; then

$$\begin{aligned} E[N] =& 1 \cdot (1-p) + 2 \cdot p(1-p) + 3 \cdot p^2(1-p) + 4 \cdot p^3(1-p) + \dots \\ =& (1-p)(1 + 2p + 3p^2 + 4p^3 + \dots) \end{aligned} \tag{3}$$

- Let us set $x = 1 + 2p + 3p^2 + 4p^3 + \dots$;
- then multiplying both sides by p we get

$$p\,x = p + 2p^2 + 3p^3 + 4p^4 + \dots$$

- We also have

$$\frac{1}{1-p} = 1 + p + p^2 + p^3 + \dots$$

- Summing the corresponding sides of the two equations we get

$$p\,x + \frac{1}{1-p} = 1 + 2p + 3p^2 + 4p^3 + \dots = x$$

Designing a Perfect Hash table (continued)

- This yields
$$(1-p)x = \frac{1}{1-p}$$
- But $(1-p)x$ is just $E[N]$; see(3).
- Thus, $E[N] = \frac{1}{1-p}$
- Since $p < 1/2$ we get that on average, less than two trials will be enough to obtain a collision free table of size $< 2n^2$.

- Recall that we aim to produce a collision free hash table of size linear in n for storing n keys; thus we proceed with the

- **Second step:** Choose M to be the smallest prime larger than n;
- Thus $n \leq M < 2n$; we now produce a hash table of size M again by choosing randomly from a universal family of hash functions.

Designing a Perfect Hash table (continued)

Assume that a slot i of this table has n_i many elements;

We now know how to design a secondary hash table of size a prime $m_i < 2n_i^2$ which stores all of n_i elements collision free.

We also have to guarantee that the sum total of sizes of all secondary hash tables, i.e., $\sum_{i=1}^{M} m_i$ is linear in n.

Note that

$$\binom{n_i}{2} = \frac{n_i(n_i - 1)}{2} = \frac{n_i^2}{2} - \frac{n_i}{2}$$

Thus, since n_i is the number of elements in the i^{th} slot, we get

$$\sum_{i=1}^{M} n_i^2 \;=\; 2\sum_{i=1}^{M}\binom{n_i}{2} + \sum_{i=1}^{M} n_i \;=\; 2\sum_{i=1}^{M}\binom{n_i}{2} + n \qquad (4)$$

Designing a Perfect Hash table (continued)

- However, $\binom{n_i}{2}$ is the total number of collisions in slot i;
- thus, $\sum_{i=1}^{M} \binom{n_i}{2}$ is the total number of collisions in the hash table.
- Since there are $\binom{n}{2}$ pairs of keys and for each pair of keys the probability of a collision with universal hashing is $1/M$, we obtain that the expected total number of collisions is $\binom{n}{2}\frac{1}{M}$.

  Thus,

$$E\left[\sum_{i=1}^{M} \binom{n_i}{2}\right] = \binom{n}{2}\frac{1}{M} = \frac{n(n-1)}{2M} \tag{5}$$

Designing a Perfect Hash table (continued)

Equations (4) and (5):

$$\sum_{i=1}^{M} n_i^2 \;=\; 2\sum_{i=1}^{M} \binom{n_i}{2} + n;$$

$$E\left[\sum_{i=1}^{M} \binom{n_i}{2}\right] = \frac{n(n-1)}{2M};$$

plus the fact that $M \geq n$ imply

$$E\left[\sum_{i=1}^{M} n_i^2\right] = \frac{n(n-1)}{M} + n \leq \frac{n(n-1)}{n} + n = 2n - 1 < 2n$$

Designing a Perfect Hash table (continued)

- Applying the Markov Inequality once again we obtain

$$P\left\{\sum_{i=1}^{M} n_i^2 > 4n\right\} \leq \frac{E\left[\sum_{i=1}^{M} n_i^2\right]}{4n} < \frac{2n}{4n} = \frac{1}{2} \tag{6}$$

- Thus, after a few attempts we will produce a hash table of size $M < 2n$ for which $\sum_{i=1}^{M} n_i^2 < 4n$.
- If we choose primes $m_i < 2n_i^2$ then $\sum_{i=1}^{M} m_i < 8n$.
- In this way the size of the primary hash table plus the sum of sizes of all secondary hash tables satisfies

$$M + \sum_{i=1}^{M} m_i < 2n + 8n = 10n.$$

Designing a Perfect Hash table (continued)

We now describe the entire randomised construction.

1. Choose M such that $n \le M < 2n$ a prime number.
2. Pick randomly a hash function with hash table size M from a universal hash functions family;
3. use it to hash all n elements into such a table;
4. check if the numbers of elements n_i in slots $i = 1, 2 \dots M$ satisfy $\sum_{i=1}^{M} n_i^2 < 4n$;
5. if not, pick randomly another hash function and try again, repeating until the above condition is satisfied;
6. equation (6) guarantees that you will succeed fast (on average after only two trials);
7. for each slot i of the table containing n_i elements use the first described randomised construction to obtain hash functions h_i which produce no collisions at all and such that the size of the corresponding hash tables are prime numbers m_i such that $m_i < 2n_i^2$.

Designing a Perfect Hash table (continued)

- Note that our procedure eventually produces a **fully deterministic** hash function;
- it is only that our search for such a function by "trial and error" was a randomised procedure.
- How does the resulting hash function operate?
  1. For a given key x compute $h(x) = i_x$ which is an index $1 \leq i_x \leq M$ for the primary hash table T of size $M < 2n$;
  2. in the slot i_x of T find the secondary hash function h_{i_x} and compute $h_{i_x}(x) = j_x$ which is an index in the secondary table t_{i_x};
  3. x (with the associated record R_x for x) is stored in slot j_x of the secondary table t_{i_x}.

Designing a Perfect Hash table (continued)