TD 11 : Universal Hashing, Rabin-Karp algorithm

1 ε-Universal hashing

Define a family $\mathscr{H}$ of hash functions from a finite set U to a finite set B to be *ε-universal* if for all pairs of distinct elements k and ℓ in U,

$$\mathbb{P}(h(k) = h(\ell)) \leq \varepsilon,$$

where the probability is over the choice of the hash function h drawn uniformly at random from the family $\mathscr{H}$.

Problem 1.1. Show that an ε-universal family of hash functions must have

$$\varepsilon \geq \frac{1}{|B|} - \frac{1}{|U|}.$$

Solution. Let $m = |U|$ and $n = |B|$. Assume for the sake of contradiction that $\varepsilon \leq \frac{1}{n} - \frac{1}{m}$. Then for all pairs of $k \neq \ell$ in U, we have

$$\mathbb{P}(h(k) = h(\ell)) < \frac{1}{n} - \frac{1}{m}.$$

We use a double counting argument. Consider a table whose rows are represented by the elements of $\mathscr{H}$ and columns consist of the pairs (k, ℓ) such that $1 \leq k < \ell \leq m$. The cell $(h, (k, \ell))$ of the table is set to 1 if $h(k) = h(\ell)$ and 0 otherwise. So each column sum is at most $\left(\frac{1}{n} - \frac{1}{m}\right)|\mathscr{H}|$, implying that the sum of all cells is at most $\binom{m}{2}\left(\frac{1}{n} - \frac{1}{m}\right)|\mathscr{H}| = \frac{(m-n)(m-1)}{2n}|\mathscr{H}|$.

We now compute the number of 1s in each row, i.e., for every $h \in \mathscr{H}$, how many pairs (k, ℓ) satisfy $h(k) = h(\ell)$. Let x_i be the number of elements in U that are mapped to the same element i in B (we assume that $U = \{1, \ldots, m\}$ and $B = \{1, \ldots, n\}$). Then the number of such pairs is $\binom{x_1}{2} + \cdots + \binom{x_n}{2}$. We would like to minimize this quantity subject to $x_1 + \cdots + x_n = m$. Since the function $f : [0, \infty) \to \mathbb{R}$ defined by $f(x) = \binom{x}{2} = \frac{x(x-1)}{2}$ is convex, we can use Jensen's inequality and get that

$$\frac{\binom{x_1}{2} + \ldots \binom{x_n}{2}}{n} \geq \binom{\frac{x_1 + \cdots + x_n}{n}}{2} = \binom{m/n}{2}.$$

Thus the sum of all cells in the table is at least $n\binom{m/n}{2}|\mathscr{H}| = \frac{m(m-n)}{2n}|\mathscr{H}|$. We have shown that the sum of all row sums cannot be equal to the sum of all column sums, which is a contradiction. □

Problem 1.2. Let U be the set of n-tuples of values drawn from $\mathbb{Z}_p$, and let $B = \mathbb{Z}_p$, where p is prime. Define the hash function $h_b : U \to B$ for $b \in \mathbb{Z}_p$ on an input n-tuple $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ from U as

$$h_b(\langle a_0, a_1, \ldots, a_{n-1} \rangle) = \left(\sum_{j=0}^{n-1} a_j b^j \right) \mod p,$$

and let $\mathscr{H} = \{h_b : b \in \mathbb{Z}_p\}$. Argue that $\mathscr{H}$ is $((n-1)/p)$-universal.

Solution. We use the fact that if p is a prime, then any nonzero polynomial $f \in \mathbb{Z}_p[x]$ of degree t has at most t roots in $\mathbb{Z}_p$. Thus for $k \neq \ell$ in $\mathbb{Z}_p^n$, there can be at most $n - 1$ values of b for which $h_b(k) = h_b(\ell)$. Since there are p possible values of b, we clearly have

$$\mathbb{P}(h_b(k) = h_b(\ell)) \leq \frac{n-1}{p}.$$

□

2 k-Universal hashing

Let $\mathscr{H}$ be a class of hash functions in which each hash function $h \in \mathscr{H}$ maps the universe U of keys to $\{0, 1, \ldots, m-1\}$. We say that $\mathscr{H}$ is *k-universal* if, for every fixed sequence of k distinct keys $\langle x^{(1)}, \ldots, x^{(k)} \rangle$ and for any h chosen at random from $\mathscr{H}$, the sequence $\langle h(x^{(1)}), \ldots, h(x^{(k)}) \rangle$ is equally likely to be any of the m^k sequences of length k with elements drawn from $\{0, 1, \ldots, m-1\}$.

Problem 2.1. Show that if the family $\mathscr{H}$ of hash functions is 2-universal, then it is universal.

Solution.

$$\mathbb{P}(h(k) = h(\ell)) = \sum_{i=0}^{m-1} \mathbb{P}\left(\langle h(k), h(\ell) \rangle = \langle i, i \rangle\right) = \frac{m}{m^2} = \frac{1}{m}.$$

□

Problem 2.2. Suppose that the universe U is the set of n-tuples of values drawn from $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$, where p is prime. Consider an element $x = \langle x_0, x_1, \ldots, x_{n-1} \rangle \in U$. For any n-tuple $a = \langle a_0, a_1, \ldots, a_{n-1} \rangle \in U$, define the hash function h_a by

$$h_a(x) = \left(\sum_{j=0}^{n-1} a_j x_j \right) \mod p.$$

Let $\mathscr{H} = \{h_a : a \in \mathbb{Z}_p\}$. Show that $\mathscr{H}$ is universal, but not 2-universal.

Solution. Let $\langle x_0, \ldots, x_{n-1} \rangle \neq \langle y_0, \ldots, y_{n-1} \rangle$ and define $z_i = x_i - y_i$. Note that there exists i such that $z_i \neq 0$. For every choice of $\langle a_0, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{n-1} \rangle \in \mathbb{Z}_p^{n-1}$, it can be easily proved that $\{\sum_i a_i z_i \pmod p \mid a_i \in \mathbb{Z}_p\} = \{0, 1, \ldots, p-1\}$. It follows that for every choice of $\langle a_0, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{n-1} \rangle \in \mathbb{Z}_p^{n-1}$, there is exactly one value of a_i such that $\sum_i a_i z_i \equiv 0 \pmod p$. Thus the number of $\langle a_0, \ldots, a_{n-1} \rangle$ such that $a_0 z_0 + \cdots + a_{n-1} z_{n-1} \equiv 0 \pmod p$ is p^{n-1}. We conclude that $\mathscr{H}$ is universal since

$$\mathbb{P}(h_a(\langle x_0, \ldots, x_{n-1} \rangle) = h_a(\langle y_0, \ldots, y_{n-1} \rangle)) = \frac{1}{p}.$$

To show that $\mathscr{H}$ cannot be 2-universal, consider the n-tuples $\mathbf{0} = \langle 0, \ldots, 0 \rangle$ and some non-zero tuple $\boldsymbol{x}$. There can only be at most p distinct values for the pair $\langle h_a(\mathbf{0}), h_a(\boldsymbol{x}) \rangle = \langle 0, h_a(\boldsymbol{x}) \rangle$, so it cannot ever attain $p^2 - p$ values in $\mathbb{Z}_p^2$. □

Problem 2.3. Suppose that we modify $\mathscr{H}$ slightly from the previous question : for any $a \in U$ and for any $b \in \mathbb{Z}_p$, define

$$h'_{ab}(x) = \left(\sum_{j=0}^{n-1} a_j x_j + b \right) \mod p$$

and $\mathscr{H}' = \{h'_{ab} : a, b \in \mathbb{Z}_p\}$. Argue that $\mathscr{H}'$ is 2-universal.

Solution. We will show for every $\boldsymbol{x} = \langle x_0, \ldots, x_{n-1} \rangle \in \mathbb{Z}_p^n$ and $m \in \mathbb{Z}_p$ that

$$\mathbb{P}\left(\sum_{i=0}^{n-1} a_i x_i + b = m \right) = \frac{1}{p}.$$

We split the proof into two cases.

1. We have $\boldsymbol{x} = \mathbf{0}$. Then

$$\mathbb{P}(h_{ab}(\boldsymbol{x}) = m) = \mathbb{P}(b = m) = \frac{p^n}{p^{n+1}} = \frac{1}{p}.$$

2. We have $x_i \neq 0$ for some i. For all $\langle a_0, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{n-1}, b\rangle \in \mathbb{Z}_p^n$, an easy modular arithmetic argument shows that

$$\left\{ \sum_{j=0}^{n-1} a_j x_j + b \mid a_i \in \mathbb{Z}_p \right\} = \{0, 1, \ldots, p-1\}.$$

For each $m \in \mathbb{Z}_p$, each $\langle a_0, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{n-1}\rangle \in \mathbb{Z}_p^n$ corresponds to a unique a_i such that $\sum_{j=0}^{n-1} a_j x_j + b = m$. So for each $m \in \mathbb{Z}_p$, the number of $\langle a_0, \ldots, a_{n-1}, b\rangle \in \mathbb{Z}_p^{n+1}$ such that $\sum_{j=0}^{n-1} a_j x_j + b = m$ is equal to p^n. Thus

$$\mathbb{P}(h_{ab}(\boldsymbol{x}) = m) = \frac{p^n}{p^{n+1}} = \frac{1}{p}.$$

□

Problem 2.4. Suppose that Alice and Bob secretly agree on a hash function h from a 2-universal family $\mathscr{H}$ of hash functions. Each $h \in \mathscr{H}$ maps from a universe of keys U to $\mathbb{Z}_p$, where p is prime. Later, Alice sends a message m to Bob over the Internet, where $m \in U$. She authenticates this message to Bob by also sending an authentication tag $t = h(m)$, and Bob checks that the pair (m, t) he receives indeed satisfies $t = h(m)$. Suppose that an adversary intercepts (m, t) en route and tries to fool Bob by replacing the pair (m, t) with a different pair (m', t'). Argue that the probability that the adversary succeeds in fooling Bob into accepting (m', t') is at most $1/p$, no matter how much computing power the adversary has, and even if the adversary knows the family $\mathscr{H}$ of hash functions used.

Solution. Fix some $k \neq m'$. The number of functions $H \in \mathscr{H}$ such that $\langle H(m'), H(k)\rangle = \langle h(m'), k'\rangle$ for any $k' \in \mathbb{Z}_p$ is $|\mathscr{H}|/p^2$. Varying k' over $\mathbb{Z}_p$ gives us that the number of functions H such that $H(m') = h(m')$ is $|\mathscr{H}|/p$. Thus the probability that the function h' chosen by the adversary satisfies $h'(m') = h(m')$ is $\frac{|\mathscr{H}|/p}{|\mathscr{H}|} = \frac{1}{p}$. □

3 Rabin-Karp pattern matching algorithm

Let $T[1 \ldots n]$ be a text and $P[1 \ldots m]$ be a pattern over an alphabet $\Sigma = \{0, 1, \ldots, |\Sigma| - 1\}$. We treat each character as an integer in $\{0, 1, \ldots, |\Sigma| - 1\}$ and interpret each m-length substring as a number in base $|\Sigma|$. For a string $S[1..m]$, define its *fingerprint* (or *hash value*) as

$$h(S[1..\ell]) = \sum_{j=1}^{\ell} S[j] \cdot |\Sigma|^{\ell - j} \pmod{p} \tag{1}$$

where p is a prime number chosen uniformly at random from all primes up to M (the value of M will be chosen later).

Problem 3.1. Define $t_i := h(T[i \ldots i + m - 1])$. Derive a formula to compute t_{i+1} from t_i in $O(1)$ time.

Solution.

$$h(T[i+1..i+m]) = \big(h(T[i..i+m-1]) - T[i] \cdot |\Sigma|^{m-1}\big) \cdot |\Sigma| + T[i+m] \pmod{p} \tag{2}$$

□

Problem 3.2. Design a randomized algorithm using fingerprints that finds all occurrences of P in T as efficiently as possible.

Solution. The algorithm is given as follows.

Algorithm : Rabin-Karp

1. Choose a prime p uniformly at random from all primes up to M (the value of M will be determined later).
2. Compute $h_P = h(P[1..m]) \pmod p$.
3. Compute $h_T = h(T[1..m]) \pmod p$.
4. If $h_T = h_P$, compare $T[1..m]$ with $P[1..m]$ character by character. If they match, report an occurrence at position 1.
5. For $i = 1$ to $n - m$:
   (a) Update h_T using the rolling hash formula (2).
   (b) If $h_T = h_P$, compare $T[i+1..i+m]$ with $P[1..m]$ character by character. If they match, report an occurrence at position $i + 1$.

□

Problem 3.3. For a random prime p chosen uniformly from all primes up to M, show that the probability of a false positive at any single position i where $T[i \dots i+m-1] \neq P[1 \dots m]$ is at most

$$\frac{m \log_2 |\Sigma|}{\pi(M)},$$

where $\pi(M)$ is the number of primes up to M.

Solution. When $T[i..i+m-1] \neq P[1..m]$, the integer

$$N_i = \sum_{j=1}^{m} \big(T[i+j-1] - P[j]\big) \cdot |\Sigma|^{m-j} \tag{3}$$

is nonzero and satisfies $|N_i| < |\Sigma|^m$. A false positive occurs if and only if p divides N_i.

The number of distinct prime factors of any positive integer N is at most $\log_2 N$, since each prime factor is at least 2 and the prime factors multiply to at most N. Therefore, N_i has at most $\log_2(|\Sigma|^m) = m \log_2 |\Sigma|$ distinct prime factors.

Since p is chosen uniformly at random among the $\pi(M)$ primes up to M, the probability that p divides N_i is at most $m \log_2 |\Sigma| / \pi(M)$. □

Problem 3.4. Show that for large enough M, the expected running time of the algorithm is $O(n + m)$. *Hint :* By the prime number theorem, $\pi(M) \geq M/(2 \ln M)$ for sufficiently large M.

Solution. By the prime number theorem, $\pi(M) \sim M / \ln M$ as $M \to \infty$. In particular, $\pi(M) \geq M/(2 \ln M)$ for sufficiently large M.

By the previous problem and a union bound over all $n - m + 1$ positions, the expected number of false positives is at most

$$(n - m + 1) \cdot \frac{m \log_2 |\Sigma|}{\pi(M)} \leq (n - m + 1) \cdot \frac{m \log_2 |\Sigma| \cdot 2 \ln M}{M} \tag{4}$$

With the choice $M = (n - m + 1) \cdot m^2 \cdot \lceil \log_2 |\Sigma| \rceil^2$, the expected number of false positives is

$$\leq \frac{2 \ln M}{m \cdot \lceil \log_2 |\Sigma| \rceil} = O(1) \tag{5}$$

since $\ln M = O(\log n + \log m + \log |\Sigma|)$.

Each false positive costs $O(m)$ for the character-by-character verification. Therefore, the expected total cost of false positives is $O(m) \cdot O(1) = O(m)$.

The rolling hash computation takes $O(n)$ time (one $O(1)$ update per position), and the initial fingerprint computations take $O(m)$ time. Reporting true occurrences costs $O(m)$ per occurrence, but this cost is charged to the output. The total expected running time is therefore $O(n + m)$.

We note that arithmetic operations are performed modulo p, so each operation involves numbers of $O(\log M)$ bits. For the bound to hold with unit-cost arithmetic, we need M to fit in $O(1)$ machine words, which is the case when n, m, and $|\Sigma|$ are polynomial in the word size. □