Universität Siegen
Lehrstuhl Theoretische Informatik
Markus Lohrey

Algorithmics 2
WS 2020/21

Exercise 5

Task 1

Let $f : \{0,1\}^* \to \mathbb{Z}^{2\times 2}$ be the homomorphism defined by

$$f(0) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad f(1) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Show that the entries of the matrix $f(w)$ are upper bounded by the $(|w|+1)$-th Fibonacci number $F_{|w|+1}$. Furthermore, give an example for a string w, where at least one entry of $f(w)$ takes indeed the value $F_{|w|+1}$.

Solution

Note that here we define $F_0 = 0$, $F_1 = 1$ and $F_{i+1} = F_i + F_{i-1}$ (there is another convention for the starting values!).

Part 1 of the task is an induction on $|w| = n$. We will actually prove a stronger claim: The entries of $f(w) = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ are bounded by

- either $a \leq F_{n+1}, b, c \leq F_n, d \leq F_{n-1}$
- or $b \leq F_{n+1}, a, d \leq F_n, c \leq F_{n-1}$
- or $c \leq F_{n+1}, a, d \leq F_n, b \leq F_{n-1}$
- or $d \leq F_{n+1}, b, c \leq F_n, a \leq F_{n-1}$.

By looking at the identity matrix, it is clear that the assumption is true for $n = 0$, if we set $F_{-1} = 1$. By the inductions hypotheses, we can assume that

$$f(w) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \leq \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$$

or one of the other 3 cases hold. Hence, for the induction step we will consider another 2 cases each:

$$f(0w) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a & b \\ a+c & b+d \end{pmatrix} \leq \begin{pmatrix} F_{n+1} & F_n \\ F_{n+2} & F_{n+1} \end{pmatrix}$$

and

$$f(1w) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a+c & b+d \\ c & d \end{pmatrix} \leq \begin{pmatrix} F_{n+2} & F_{n+1} \\ F_n & F_{n-1} \end{pmatrix},$$

where both matrices satisfy one of the four stated conditions. The other 6 cases work analogously.

Part 2: Take for instance $w = (10)^n$ for even $|w|$ and $w = 0(10)^n$ for odd $|w|$. This yields

$$f((10)^n) = \begin{pmatrix} F_{2n+1} & F_{2n} \\ F_{2n} & F_{2n-1} \end{pmatrix}, \qquad f(0(10)^n) = \begin{pmatrix} F_{2n+1} & F_{2n} \\ F_{2n+2} & F_{2n+1} \end{pmatrix}.$$

Task 2

Let $T = 001100$ and $P = 01$. Use the probabilistic algorithm of the lecture to compute the array MATCH$[1, \ldots, 6]$, which encodes the occurrences of the pattern P in the string T.

Solution

Choose $k = 1$. Let $M = 2 \cdot 6^2 = 72$. We randomly pick $p = 2 \leq 72$. We obtain $f(P) = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$ and hence $f_2(P) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$. Furthermore we compute

$$f(00) = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, \quad f(10) = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}, \quad f(11) = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix},$$

Thus, we only have a match for $i = 2$, which yields MATCH$[1, \ldots, 6] = (0, 1, 0, 0, 0, 0)$. If we would have chosen $P = 00$, we would have obtained MATCH$[1, \ldots, 6] = (1, 0, 1, 0, 1, 0)$ for $p = 2$, since $f_2(00) = f_2(11)$, so we have one false match.

Task 3

In this task we will consider an alternative class of fingerprint functions. For a word $w = a_1 \ldots a_n \in \{0, 1\}^*$ we define

$$h(a_1 \ldots a_n) = \sum_{i=1}^{n} a_i 2^{n-i}.$$

Let $h_p(w) = h(w) \bmod p$ be the *fingerprint* of w with respect to a prime p.

(a) Construct a randomised pattern matching algorithm by using these fingerprint functions.

(b) What is the probability of an invalid match of your algorithm?

Solution

Part (a): Let $T = a_1 \cdots a_n$ be the text and $P = b_1 \cdots b_m$ be the pattern. We denote $T[i, i + m - 1]$ by T_i. The algorithm works as follows:

- Choose a prime $1 \leq p \leq mn^2$ randomly.
- Compute $h_p(P)$ and $h_p(T_1)$ (via Horner's method)
- For all $i = 1, \ldots, n - m + 1$:
  - If $h_p(P) = h_p(T_i)$: **Match** at position i
  - Compute $h_p(T_{i+1}) = (h_p(T_i) - a_i 2^{m-1}) \cdot 2 + a_{i+m} \bmod p$.

Part (b): The probability for an invalid match is $\mathcal{O}(\frac{1}{n})$. In order to prove this claim, we will make use of two lemmas:

Lemma 1 Let $(X_1, Y_1), \dots, (X_t, Y_t)$ be pairs of strings of length m and let $1 \leq p \leq M$ be a randomly chosen prime. The probability of an invalid match is at most $\frac{\pi(mt)}{\pi(M)}$ (for large enough m, but $mt \leq M$), where $\pi(n)$ counts the number of primes in the set $\{1, \dots, n\}$.

Proof An invalid match means

$$\begin{aligned} &\exists i : h(X_i) \neq h(Y_i) \text{ and } h_p(X_i) = h_p(Y_i) \\ \Longrightarrow & p \text{ is a prime factor of } \prod_{\substack{1 \leq i \leq t \\ X_i \neq Y_i}} (|h(X_i) - h(Y_i)|) \leq 2^{mt} \end{aligned}$$

Furthermore, the number of prime factors of $u \leq 2^{mt}$ is bounded by $\pi(mt)$, if $mt \geq 29$ (slide 108). Hence the probability of an invalid match is $\frac{\pi(mt)}{\pi(M)}$.

Lemma 2 If we randomly choose $1 \leq p \leq M = mt^k$ then the probability of an invalid match is at most $\mathcal{O}(\frac{1}{t^{k-1}})$.

Proof This follows directly from Lemma 1 together with the fact $\pi(n) \in \mathcal{O}(\frac{n}{\ln(n)})$.
We put everything together. Our algorithm works with $k = 2$, $t = n$ and a pattern of length m. Thus we obtain an error probability of $\mathcal{O}(\frac{1}{n})$.
Example: $m = 250, n = 4000, M = mn^2 = 4 \cdot 10^9 < 2^{32}$ (32-bit-fingerprints)
Hence $\Pr(\text{invalid match}) < 10^{-3}$.

Task 4
For a given number $r \geq 1$ and a prime p let $x = (x_0, x_1, \dots, x_r)$ with $x_i \in \mathbb{F}_p$. Let $h_x : \mathbb{F}_p^{r+1} \to \mathbb{F}_p$ be the function defined by

$$h_x(a) = \sum_{i=0}^{r} a_i x_i \bmod p, \quad a = (a_0, \dots, a_r).$$

Show that $\mathcal{H} = \{h_x | x_i \in \mathbb{F}_p, 0 \leq i \leq r\}$ is a universal familiy of hash functions.
Is $\mathcal{H}$ also a familiy of pairwise independent hash functions?

Solution
For Part 1 of the task, we have to look at collisions. Let $a \neq \bar{a}$. Hence, there exists an i such that $a_i \neq \overline{a_i}$. W.l.o.g. we can choose $i = 0$. Then $h_x(a) = h_x(\bar{a})$ means

$$x_0(a_0 - \overline{a_0}) \equiv -\sum_{i=1}^{r} x_i(a_i - \overline{a_i}) \bmod p.$$

And since $a_0 \not\equiv \overline{a_0} \bmod p$ and p is a prime, we know $(a_0 - \overline{a_0})$ is invertible modulo p. Finally, by looking at

$$x_0 \equiv -\sum_{i=1}^{r} x_i \frac{a_i - \overline{a_i}}{a_0 - \overline{a_0}} \bmod p$$

we see that by randomly picking $x_1, \dots, x_r$, there is exactly one $x_0 \in \mathbb{F}_p$ fulfillig this congruence. This means there are p^r (out of p^{r+1}) many collisions and the probability for a collision in $\mathcal{H}$ is exactly $1/p$.
But: The family $\mathcal{H}$ is not a familiy of pairwise independent hash functions. Take for instance $r = 1$ and consider the system

$$\begin{aligned} 4x_0 + 2x_1 &\equiv 1 \bmod p \\ 2x_0 + x_1 &\equiv 1 \bmod p. \end{aligned}$$

It only works if $0 \equiv -1 \bmod p$, which is a contradiction. So there is no function h_x for the given values and hence the probability is 0.