**CSC473: Advanced Algorithm Design** Winter 2017

# Week 4: Karp-Rabin Algorithm Analysis

*Aleksandar Nikolov*

Here we give a rigorous analysis of the Karp-Rabin[1] algorithm as described in Section 32.2 of CLRS. We assume that the alphabet is $\Sigma = \{0, 1\}$, and $d = |\Sigma| = 2$, and also that $q$ is chosen to be a uniformly random prime in $\{2, \dots, Q\}$ for an appropriately large integer $Q$. A slight difference in notation is that we use $p$ and $t_s$ to denote the numbers $\sum_{i=1}^{m} 2^{m-i} P[i]$ and $\sum_{i=1}^{m} 2^{m-i} T[s+i]$, respectively, and we use $\tilde{p} = p \bmod q$ and $\tilde{t}_s = t_s \bmod q$. What we call $\tilde{p}$ and $\tilde{t}_s$ is called $p$ and $t_s$ in the code on page 993 of CLRS.

We prove the following theorem:

**Theorem 1.** *There exists a constant $C$ such that the following holds. Assume that $m \geq 1$, $n \geq 2$, $Q \geq Cmn \ln(mn)$, and $q$ is chosen uniformly at random in the set of prime numbers in $\{2, \dots, Q\}$. Then, with probability at least $1/2$, the Karp-Rabin algorithm reports no false matches.*

Here the constant $C$ is independent of $Q$, $m$, and $n$, and of all other parameters of the problem. It is an "absolute constant".

We need two basic facts from number theory to prove this theorem.

**Lemma 2.** *Any positive integer $X$ has at most $\log_2 X$ distinct prime divisors.*

*Proof.* By the Fundamental Theorem of Arithmetic, we can write $X$ uniquely as the product $X = p_1 p_2 \dots p_k$, where $p_1, \dots, p_k$ are all (not necessarily distinct) prime numbers. Then the number of prime divisors of $X$ is at most $k$. Because any prime number is at least 2, we have $X = p_1 \dots p_k \geq 2^k$, and therefore, $k \leq \log_2 X$. □

The second fact is the famous Prime Number Theorem.

**Lemma 3.** *Let $\pi(Q)$ be the number of primes in $\{2, \dots, Q\}$. There exists a constant $c > 0$, independent of $Q$, such that for any integer $Q \geq 2$, $\pi(Q) \geq \frac{cQ}{\ln Q}$.*

Proving this lemma would take us far beyond the scope of this class. The version we cited is a bit weaker than the prime number theorem, and is due to Chebyshev.

*Proof of Theorem 1.* Define $S = \{s : t_s \neq p\}$ to be the set set of shifts for which there is *no* match. Define the number

$$X = \prod_{s \in S} |t_s - p|.$$

why 2^?-

$X$ is a positive integer. Since $|S| \leq n - m$ and $|t_s - p| \leq 2^m$, we have $X \leq 2^{m(n-m)} \leq 2^{mn}$. By Lemma 2, this implies that $X$ has at most $mn$ prime divisors.

[1] Note that CLRS mysteriously calls this algorithm Rabin-Karp, but we will refer to it as Karp-Rabin, to match both the alphabetical order and the order in which the names appear on the original paper.

t_s != p and tilde{ t_s = p }

Notice that we have a false match exactly when there exists a shift $s \in S$ for which $\tilde{t}_s = \tilde{p}$, which happens exactly when $q$ divides $|t_s - p|$. Therefore, if there is any false match, then $q$ would divide $X$. This means that

$$\begin{aligned}\mathbb{P}(\text{false match}) \leq \mathbb{P}(q \text{ divides } X) &= \frac{\#\{\text{prime divisors of } X\}}{\pi(Q)} \\ &\leq \frac{mn\ln(Cmn\ln(mn))}{cCmn\ln(mn)} \\ &= \frac{1}{cC} + \frac{\ln(C)}{cC\ln(mn)} + \frac{\ln\ln(mn)}{cC\ln(mn)}.\end{aligned}$$

It is easy to prove (using some basic calculus, for example), that for large enough $C$ the right hand side can be made less than $\frac{1}{2}$, or any other fixed constant. □

A few comments on the algorithm are in order:

note this is from Q >= Cmnln(mn), so binary representation of Q <= ln(Cmn ln(mn)) <= ln(mn)

- Notice that $\tilde{p}$ and $\tilde{t}_s$ are numbers in $\{0, \ldots, q-1\} \subseteq \{0, \ldots, Q-1\}$, and any such number can be written in $\Theta(\log mn)$ bits. Therefore, in the word RAM model of computation, we can do perform arithmetic operations and comparisons on numbers of this size in time $O(1)$. This means that the algorithm runs in worst-case time $O(m+n)$, i.e. linear in the size of the input.

- This is a Monte Carlo algorithm, and may make a mistake with constant probability. As usual, by running the algorithm several times (or just choosing several different random primes, and comparing the fingerprints for all these choices) we can make this constant arbitrarily small without affecting the asymptotic complexity. However, we may want to verify every match. This turns the algorithm into a Las Vegas algorithm. The analysis above can be modified to show that if there are $\ell$ true matches, the Karp-Rabin algorithm with verification will take *expected time* $O(m\ell + n)$, which is still quite fast if $\ell$ is small.

- We have not discussed *how* to pick a random prime in $\{2, \ldots, Q\}$. The only method I am aware of is: pick random integers in $\{2, \ldots, Q\}$ and output the first one which is a prime number. This technique is called *rejection sampling*: **make sure you understand why it gives a uniform distribution on the primes in $\{2, \ldots, Q\}$.** We only need $O(\log Q)$ samples in expectation before we find a prime: this is a consequence of Lemma 3. (**Make sure you understand why this is true as well.**) In order to implement this algorithm, we need an algorithm to check whether a number is prime. There is a beautiful efficient Monte Carlo algorithm for this problem, due to Miller and Rabin. See Section 31.8. of CLRS.