

The work you submit must be your own. You may discuss problems with each other; however, you should prepare written solutions alone. In particular, you should not leave with any written notes from such discussions. The style and clarity of your answers will be an important factor in the grade.

This assignment has two questions, each worth 50%.

1. (a) $M(x)$ is well defined because matrix multiplication is associative.
- (b) Given $M = M(x)$ we can “decode” x uniquely as follows: if the first column of M is greater than the second, then the last bit of x is zero, and otherwise it is 1. Let M' be M where we subtract the smaller column from the larger, and repeat.
- (c) Let $F_0 = F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n > 1$. For a given string x , $M(x_1x_2 \dots x_n)$ is such that the “smaller” column is bounded by F_{n-1} and the “larger” column is bounded by F_n . We can show this inductively: the basis case, $x = x_1$, is obvious. For the inductive step, assume it holds for $x \in \{0, 1\}^n$, and show it still holds for $x \in \{0, 1\}^{n+1}$: this is obvious as whether x_{n+1} is 0 or 1, one column is added to the other, and the other column remains unchanged.
- (d) We wish to determine whether x is a substring of y , where $|x| = n$, $|y| = m$, $n \leq m$. Let $y(i) = y_i y_{i+1} \dots y_{n+i-1}$, for appropriate i 's. Select a prime $p \in \{1, \dots, nm^2\}$, and let $A = M(x) \pmod{p}$ and $A(i) = M(y(i)) \pmod{p}$. Note that

$$A(i+1) = M^{-1}(y_i)A(i)M(y_{n+i}) \pmod{p}.$$

So for all appropriate i 's, we check whether $A = A(i)$. If yes, we check whether we didn't get a false positive with a bit-by-bit comparison. If they match, we answer “yes”, otherwise we change the prime p and continue¹.

What is the probability of getting a false positive? It is the probability that $A(i) = M(y(i)) \pmod{p}$ even though $A(i) \neq M(y(i))$. This is less than the probability that $p \in \{1, \dots, nm^2\}$ divides a (non-zero) entry in $A(i) - M(y(i))$. Since these entries are bounded by $F_n < 2^n$, less than n distinct primes can divide any of them. On the other hand, there are $\pi(nm^2) = (nm^2)/(\log(nm^2))$ primes in $\{1, \dots, nm^2\}$. So the probability of a false positive is $\leq O(1/m)$.

Note that this algorithm has no error; it is randomized, but all potential answers are checked for false positives. Checking for these potential candidates is called *fingerprinting*.

How to select random primes? Random primes are needed to find public keys p, q for the RSA encryption scheme. It is a non-trivial problem, primarily because verifying the primality of a number is difficult. Here is how we go about it: we know by the prime number theorem that there are about $\pi(n) = n/\log(n)$ many

¹For more details on this question, and in particular for an interesting discussion of why we need to change the prime after a false positive, see the article *Efficient randomized pattern-matching algorithms*, by Richard M. Karp and Michael O. Rabin, 1987.

primes $\leq n$. This means that there are $2^n/n$ primes among n -bit integers, roughly 1 in n . So we pick an integer at random, in a given range, and apply a primality testing algorithm to it. The Rabin-Miller algorithm works as follows: given an odd number n , we let $n - 1 = 2^k m$, where m is also odd. Choose $1 < a < n - 1$, and calculate the following sequence:

$$a^m, a^{2m}, a^{4m}, \dots, a^{2^k m}$$

modulo n . We say that n passes the Rabin-Miller test for base a if either the first term is 1, or 1 occurs later in this sequence and is immediately preceded by -1 . If the test fails, we know for sure that n is composite (and a is a *witness* of the compositeness of n). If n is an (odd) composite, then it turns out that at least $3/4$ of a 's are witnesses to this fact. Hence we have a probabilistic primality testing algorithm: choose a_1, a_2, \dots, a_r independently at random. Then if (odd) n passes the Rabin-Miller test for each a_i , the probability that n is composite is no more than $(1/4)^r$.

2. From Papadimitriou we know that the probability that a fixed node v is *not* visited by a random walk of length dn^2 starting at some node u is $\leq 1/2$. If we repeat this "experiment" m times, or, equivalently, increase the random walk to $dn^2 m$ many steps, the probability of not visiting v becomes $\leq 1/2^m$. So the probability of missing some node during the random walk is $\leq n/2^m$. The number of d -regular graphs with n nodes is (grossly) bounded above by n^{dn} , so we want to choose an m such that $(n/2^m)(n^{dn}) < 1$, assuring that there is a universal traversal sequence of all d -regular graphs with n nodes. Choose $m = n^2$.

This tells us two things about undirected reachability: (i) it is in **RL** (randomized log-space, with no false positives), and (ii) it is in **L/poly**, i.e., log-space with polynomial advice. Note that a recent result of Omer Reingold placed undirected reachability in **L**.