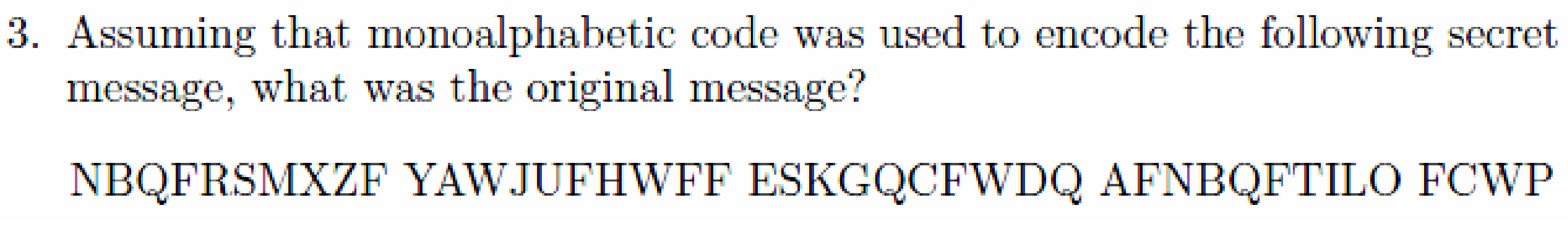
问题与反馈

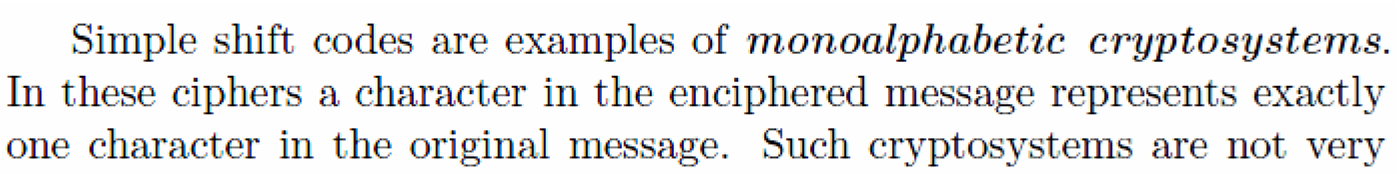
2015-4-2

Monoalphabetic cryptosystems



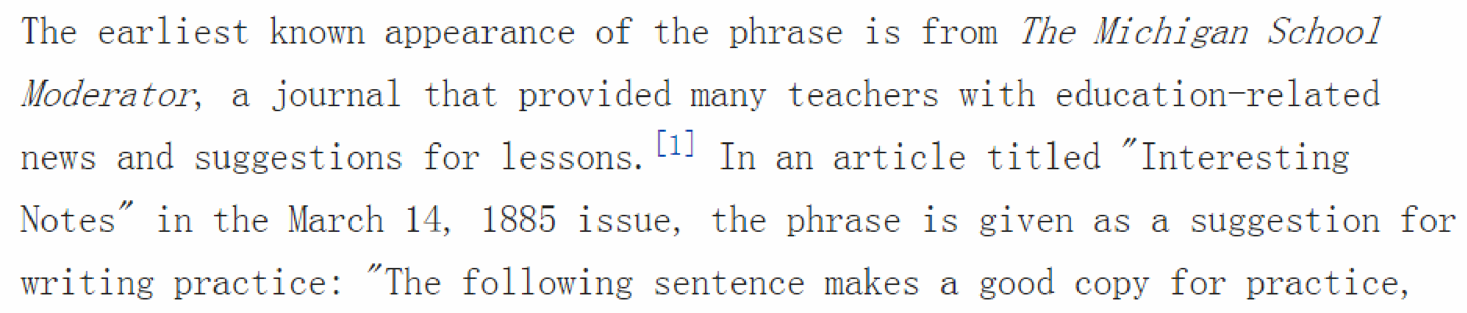
3. Assuming that monoalphabetic code was used to encode the following secret message, what was the original message?

NBQFRSMXZF YAWJUFHWFF ESKGQCFWDQ AFNBQFTILO FCWP



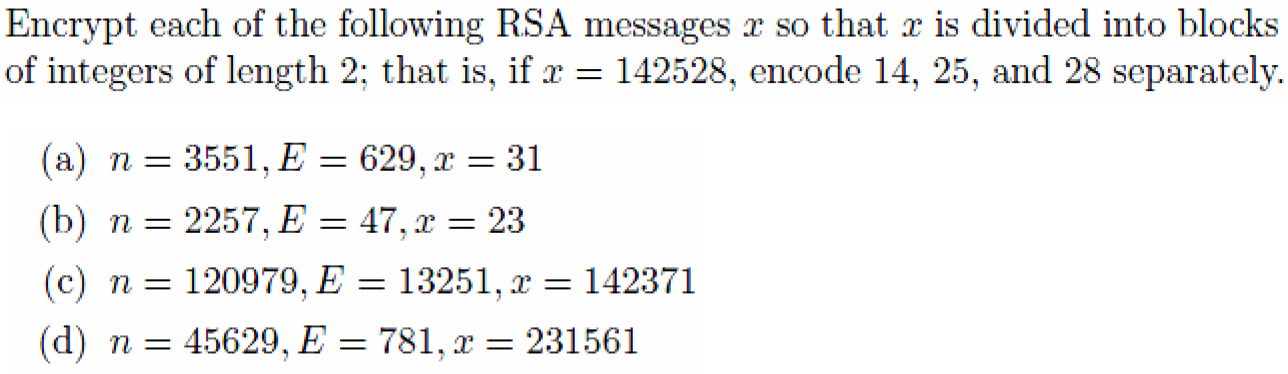
Simple shift codes are examples of *monoalphabetic cryptosystems*. In these ciphers a character in the enciphered message represents exactly one character in the original message. Such cryptosystems are not very

'T','H','E','X','Q','U','I','C','K','X','B','R','O','W','N','X',
'F','O','X','X','J','U','M','P','E','D','X','O','V','E','R',
'X','T','H','E','X','L','A','Z','Y','X','D','O','G'



The earliest known appearance of the phrase is from *The Michigan School Moderator*, a journal that provided many teachers with education-related news and suggestions for lessons.[1] In an article titled "Interesting Notes" in the March 14, 1885 issue, the phrase is given as a suggestion for writing practice: "The following sentence makes a good copy for practice,

RSA



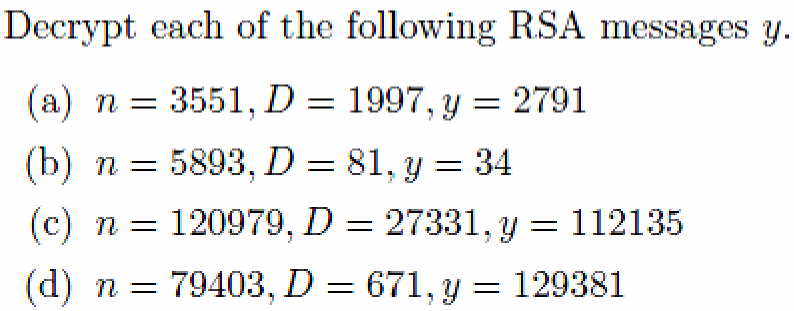
Encrypt each of the following RSA messages x so that x is divided into blocks of integers of length 2; that is, if $x = 142528$, encode 14, 25, and 28 separately.

(a) $n = 3551, E = 629, x = 31$

(b) $n = 2257, E = 47, x = 23$

(c) $n = 120979, E = 13251, x = 142371$

(d) $n = 45629, E = 781, x = 231561$



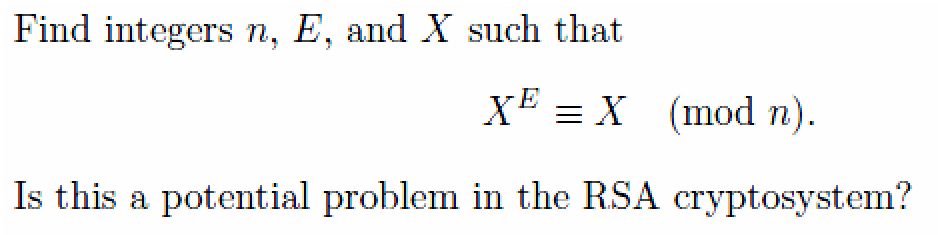
Decrypt each of the following RSA messages y.

(a) $n = 3551, D = 1997, y = 2791$

(b) $n = 5893, D = 81, y = 34$

(c) $n = 120979, D = 27331, y = 112135$

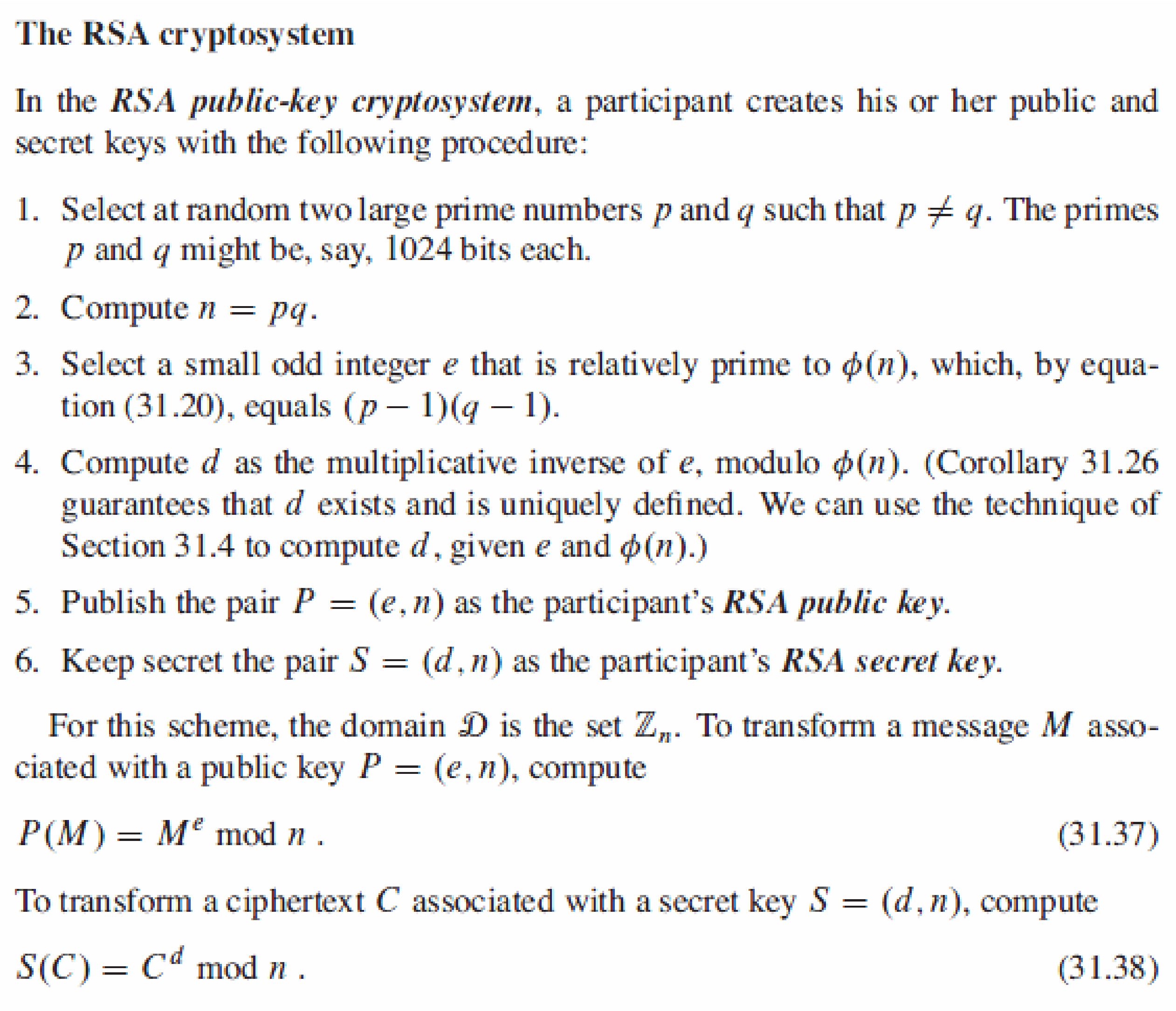
(d) $n = 79403, D = 671, y = 129381$



Find integers n, E, and X such that

$$X^E \equiv X \pmod{n}.$$

Is this a potential problem in the RSA cryptosystem?



The RSA cryptosystem

In the ***RSA public-key cryptosystem***, a participant creates his or her public and secret keys with the following procedure:

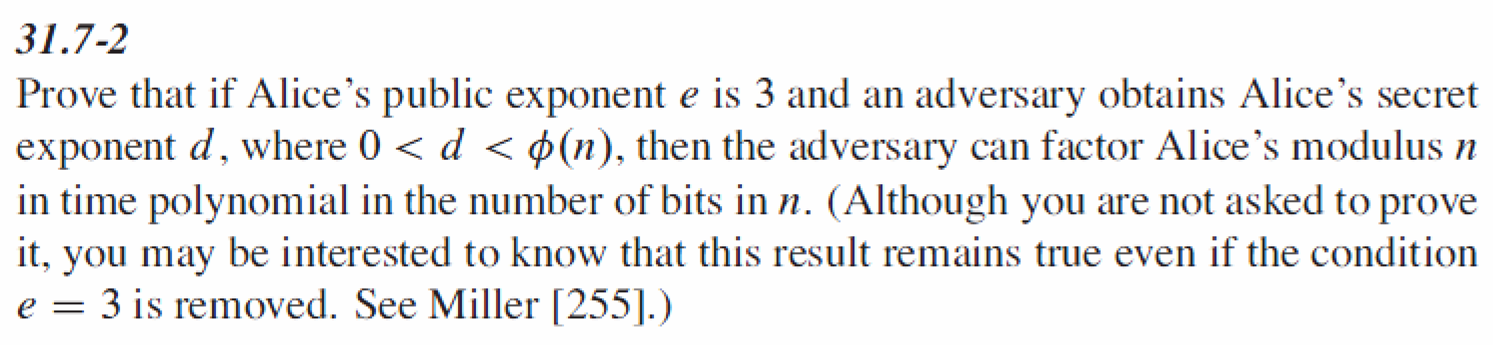
1. Select at random two large prime numbers p and q such that $p \neq q$. The primes p and q might be, say, 1024 bits each.
2. Compute $n = pq$.
3. Select a small odd integer e that is relatively prime to $\phi(n)$, which, by equation (31.20), equals $(p-1)(q-1)$.
4. Compute d as the multiplicative inverse of e, modulo $\phi(n)$. (Corollary 31.26 guarantees that d exists and is uniquely defined. We can use the technique of Section 31.4 to compute d, given e and $\phi(n)$.)
5. Publish the pair $P = (e, n)$ as the participant's ***RSA public key***.
6. Keep secret the pair $S = (d, n)$ as the participant's ***RSA secret key***.

For this scheme, the domain $\mathcal{D}$ is the set $\mathbb{Z}_n$. To transform a message M associated with a public key $P = (e, n)$, compute

$$P(M) = M^e \bmod n . \qquad (31.37)$$

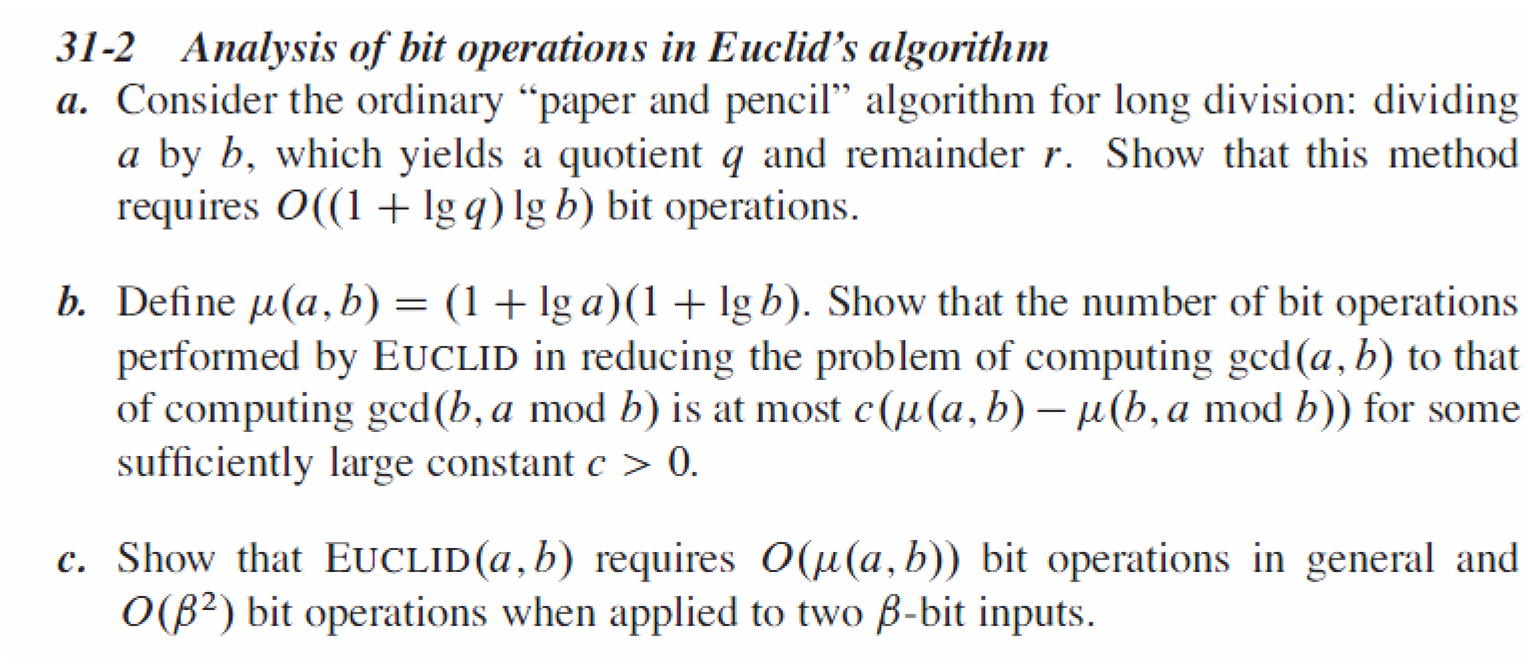
To transform a ciphertext C associated with a secret key $S = (d, n)$, compute

$$S(C) = C^d \bmod n . \qquad (31.38)$$



31.7-2

Prove that if Alice's public exponent e is 3 and an adversary obtains Alice's secret exponent d, where $0 < d < \phi(n)$, then the adversary can factor Alice's modulus n in time polynomial in the number of bits in n. (Although you are not asked to prove it, you may be interested to know that this result remains true even if the condition $e = 3$ is removed. See Miller [255].)

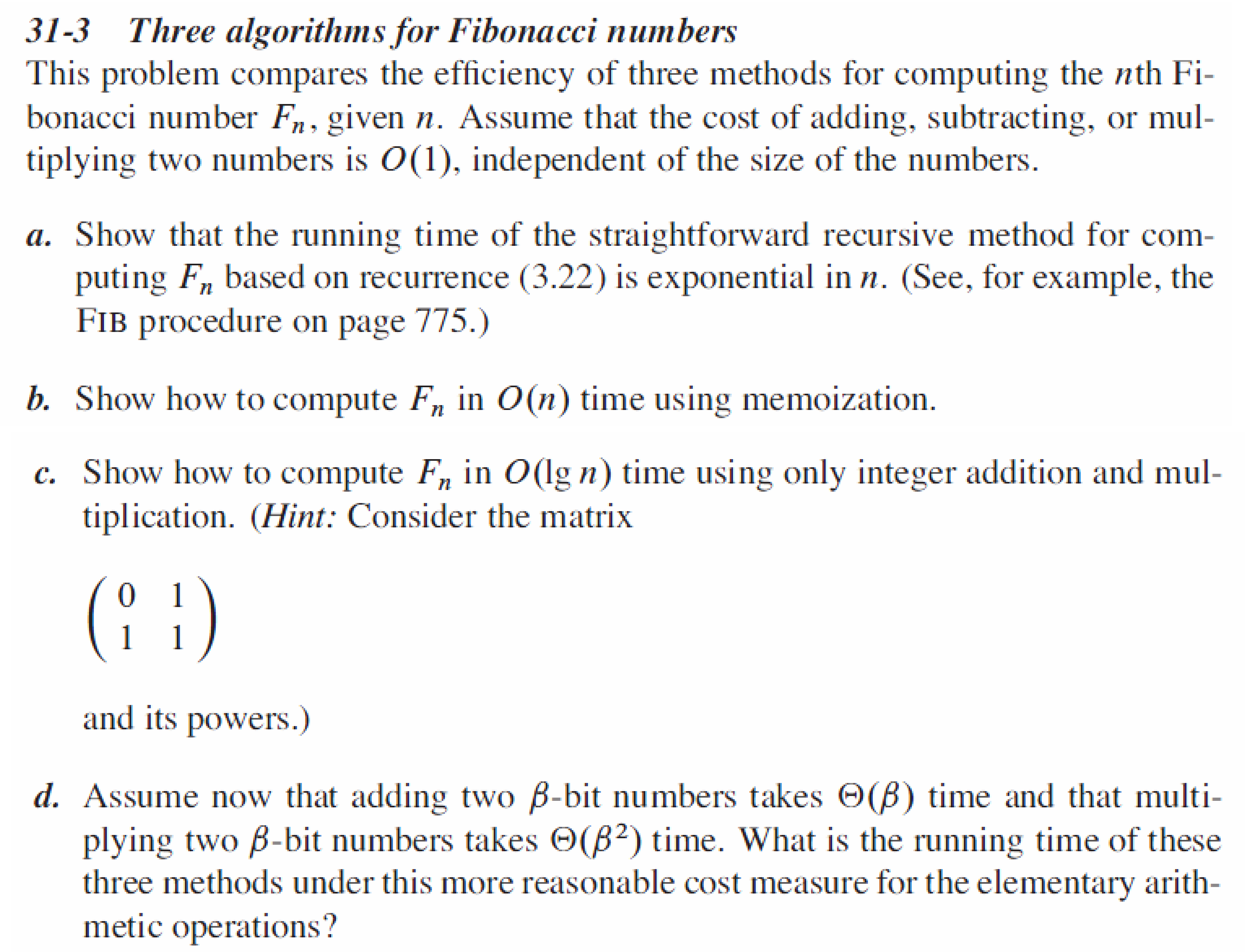


31-2 *Analysis of bit operations in Euclid's algorithm*

a. Consider the ordinary "paper and pencil" algorithm for long division: dividing a by b, which yields a quotient q and remainder r. Show that this method requires $O((1 + \lg q) \lg b)$ bit operations.

b. Define $\mu(a, b) = (1 + \lg a)(1 + \lg b)$. Show that the number of bit operations performed by EUCLID in reducing the problem of computing $\gcd(a, b)$ to that of computing $\gcd(b, a \bmod b)$ is at most $c(\mu(a, b) - \mu(b, a \bmod b))$ for some sufficiently large constant $c > 0$.

c. Show that EUCLID(a, b) requires $O(\mu(a, b))$ bit operations in general and $O(\beta^2)$ bit operations when applied to two β-bit inputs.



31-3 *Three algorithms for Fibonacci numbers*

This problem compares the efficiency of three methods for computing the nth Fibonacci number F_n, given n. Assume that the cost of adding, subtracting, or multiplying two numbers is $O(1)$, independent of the size of the numbers.

a. Show that the running time of the straightforward recursive method for computing F_n based on recurrence (3.22) is exponential in n. (See, for example, the FIB procedure on page 775.)

b. Show how to compute F_n in $O(n)$ time using memoization.

c. Show how to compute F_n in $O(\lg n)$ time using only integer addition and multiplication. (*Hint:* Consider the matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

and its powers.)

d. Assume now that adding two β-bit numbers takes $\Theta(\beta)$ time and that multiplying two β-bit numbers takes $\Theta(\beta^2)$ time. What is the running time of these three methods under this more reasonable cost measure for the elementary arithmetic operations?