CS 58000_01/02I Design, Analysis and Implementation Algorithms (3 cr.)

Assignment As_02 (Exam 01)

Student Name: _____

This assignment As_02 is due at 10:30 am, Thursday, October 18, 2022. Please submit your assignment to Brightspace (purdue.brightspace.com). No late turn-in is accepted. Please write your name on the first page of your assignment. Your file name should be your last name such as Ng.docx. Please number your problem-answer clearly such as Problem 1a, 1b, 1c, Problem 2a, 2b, …. The problems' answers must be arranged in order. Please answer your questions using only a Word file (.docx file only). No pdf file will be accepted. Without using a Word file (.docx file) the submitted problems' answers would not be graded.

The total number of points for this Assignment_02 (Exam 01) is 140 points.

Problem 1[30 points]:

In Ch 00_03, we addressed Figure 1.4 Modular Exponentiation: Given a function modexp(x, y, N) for computing x^y mod N, where x, y, and N are integers. We also addressed $a^k \ mod \ n$, when k is a power of 2, and a is any integer. We also addressed Fermat's Little Theorem.

1a. What is $4^{2^{2018}}$ (mod 17)?

1b. What is $4^{2^{2006}} (mod\ 31)$? (Hard Problem)

1c. Construct (Design) a polynomial-time algorithm for computing x^{y^z} (mod p), where x, y, z, and a prime p.

Problem 2 [60 points]:

This problem is an exercise using the formalization of the RSA public-key cryptosystem. For solving the problems, you are required to use the following formalization of the RSA public-key cryptosystem.

Given the following formalization of the RSA public-key cryptosystem, each participant creates their public key (n, g) where a is a small prime number, and n is the product of two large primes, p and q. However, the two large primes p and q are secret keys.

1. Select two very large prime numbers p and q. The number of bits needed to represent p and q might be 1024.

2. Compute

   n = pq

   φ(n) = (p – 1) (q – 1).

   The formula for φ(n) is owing to the Theorem: The number of elements in $z_n^* = \{ [1]_n, [2]_n, \ldots, [n-1]_n \}$ is given by Euler's totient function, which is

   $$\varphi(n) = n \prod_{p:p|n} (1 - \frac{1}{p}),$$

   where the product is over all primes that divide n, including n if n is prime.

3. Choose a small prime number as an encryption component g, that is relatively prime to φ(n). That means,

   gcd(g, φ(n)) = 1, i.e.,

   gcd(g, (p-1)(q-1)) = 1.

4. Compute the multiplicative inverse $[h]_{\varphi(n)}$ of $[g]_{\varphi(n)}$. That is,

   $$[g]_{\varphi(n)}[h]_{\varphi(n)} = [1]_{\varphi(n)}.$$

The inverse exists and is unique.
That is, the decryption component h = g^{-1} mod φ(n).

5. Let pkey = (n, g) be the public key, and skey = (p, q, h) be the secret key.
   - For any message M mod n, the encryption of M is C = M^g mod n.
   - The decryption of C is M = C^h mod n.

End of the formalization of the RSA public-key cryptosystem.

Use the RSA Cryptosystem formalism for solving problem 2.

Given g = 59, p = 991 and q = 997.

2a. Show that the given values of g, p, and q are prime.

2b. Compute n = pq and φ(n) = (p – 1) (q – 1).

2c. Given a plaintext **M = 5065747269**, what is the encryption of M, using $C = M^g \bmod n$. Show in detail how you derive C, which is the ciphertext of the plaintext M.

2d. Compute the multiplicative inverse $[h]_{\varphi(n)}$ of $[g]_{\varphi(n)}$. That is, the decryption component $h = g^{-1} \bmod \varphi(n)$.

[Hints: Compute a GCD as a Linear Combination. Then, find an Inverse Modulo n. In other words, you can apply the extended Euclid algorithm to find the linear combination of g and $\varphi(n)$. Then find a positive inverse of g mod $\varphi(n)$.]

2e. From problem 2d, what is your secret key (p, q, h)?

2f. What is the decryption of C using $M = C^h \bmod n$? Show in detail how you derive M, which is the plaintext M of the ciphertext C.

2g (Bonus)[5 points]:

What is the message (in terms of the alphabet)?

Problem 3[30 points]:

Assume that we define

$h_1(k) = k \bmod 13$, and

$h_2(k) = 1 + (k \bmod 11)$.

For the open addressing, consider the following methods

Linear Probing

Given an ordinary hash function $h: U \rightarrow \{0, 1, 2, \ldots, m-1\}$, the method of *linear probing* uses the hash function

$h(k, i) = (h_1(k) + i) \bmod m$ for $i = 0, 1, 2, \ldots, m-1$.

Quadratic Probing

Uses a hashing function of the form

$h(k, i) = (h_1(k) + c_1 i + c_2 i^2) \bmod m$,

where h_1 is an auxiliary hash function, c_1 and $c_2 \neq 0$ are auxiliary constants $c_1 = 3$ $c_2 = 5$,

and $i = 0, 1, 2, \ldots, m-1$.

Double hashing

Uses a hashing function of the form

$h(k, i) = (h_1(k) + i\, h_2(k)) \bmod m$,

where h_1 and h_2 are auxiliary hash functions.

The value of $h_2(k)$ must never be zero and should be relatively prime to m for the sequence to include all possible addresses.

Given K = {79, 69, 98, 72, 14, 50, 88, 99, 78, 65} and the size of a table is 13, with indices counting from 0, 1, 2, …, 12. Store the given K in a table with the size 13 counting the indices from 0, 1, 2, …, 12. Show the resultant table with 10 given keys for each method applied:

3a. if linear probing is employed.

The Resultant Table with 10 given keys is: Complete the table.

3b. if quadratic probing is employed.

The Resultant Table with 10 given keys is: Complete the table.

3c. if double hashing is employed.

The Resultant Table with 10 given keys is: Complete the table.

Problem 4 [20 points]:

For the division method for creating hash functions, map a key k into one of the m slots by taking the remainder of k divided by m. The hash function is:

$$h(k) = k \bmod m,$$

where m should not be a power of 2.

For the multiplication method for creating hash functions, the hash function is

$$h(k) = \lfloor m(kA - \lfloor k A \rfloor) \rfloor = \lfloor m(k A \bmod 1) \rfloor$$

where "k A mod 1" means the fractional part of k A and a constant A in the range $0 < A < 1$.

An advantage of the multiplication method is that the value of m is not critical.

Choose $m = 2^p$ for some integer p.

Give your explanations for the following questions:

4a. Why m should not be a power of 2 in the division method for creating a hash function?

4b. Why $m = 2^p$, for some integer p, could be (and in fact, favorably) used in the multiplication method?

Note: If you provide your answer in your handwriting, good handwriting is required. Proper numbering of your answer to each problem is strictly required. The problem's solution must be orderly given. (10 points off if not)