CS 5/7350 - Test#2
April 20, 2022

Name: Bingying Liang
ID: 48999397

1. [9 pts] Define the following Terms as succinctly as possible.

   (a) Algorthm

   Solution: A step-by-step procedure for solving a problem in a finite amount of time.

   (b) Dynamic Programming

   Solution: A dynamic-programming algorithm solves each subsubproblem just once and then saves its answer in a table, thereby avoiding the work of recomputing the answer every time it solves each subsubproblem.

   (c) $\Phi(N)$

   Solution: Euler's phi (or totient) function of a positive integer n is the number of integers in {1,2,3,...,n} which are relatively prime to n. This is usually denoted $\Phi(n)$.

   (d) Longest Common Subsequence

   Solution: Given two sequences X and Y, we say that a sequnence Z is common subsequence of X and Y if Z is a subsequence of both X and Y. For example, if X = {A,B,C,D,A,B} and Y ={B, D, C, A, B, A}, the sequence {B,C,A} is a common subsequence of both X and Y. The sequence{B,C,A} is not a longest common subsequence (LCS) of X and Y, however, since it has length 3 and the sequence {B, C, B, A}, which is also common to both sequences X and Y, has length 4. The sequence {B, C, B, A} is an LCS of X and Y, as is the sequence {B, C, A, B}, since X and Y have no common subsequence of length 5 or greater.

   (e) NP-Hard

   Solution: In computational complexity theory, NP-hardness (non-deterministic polynomial-time hardness) is the defining property of a class of problems that are informally "at least as hard as the hardest problems in 'NP". A simple example of an NP-hard problem is the subset sum problem.

   (f) Fibonacci sequence

   Solution: In mathematics, the Fibonacci sequence is a sequence in which each number is the sum of the two preceding ones. Numbers that are part of the Fibonacci sequence are known as Fibonacci numbers, commonly denoted F_n. The sequence commonly starts from 0 and 1, although some authors start the sequence from 1 and 1 or sometimes (as did Fibonacci) from 1 and 2. Starting from 0 and 1, the first few values in the sequence are:0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144.

2. [6 pts] Compute the following {note 91339 is the product of two primes 241 and 379}:

(a) Compute $\Phi(91339) =$

Solution:

$$\Phi(91339) = \Phi(241 \times 379) = (241 - 1) \times (379 - 1) = 240 \times 378 = 90720$$

(b) For which values of $|V|$ does a cycle with V vertices have an Euler Tour ___

Solution: All

(c) Compute ${}_{21}C_2 =$ ____

Solution:

$${}_{21}C_2 = \frac{21!}{2!(21-2)!} = \frac{21 \times 20 \times 19!}{2!19!} = 210$$

3. [8 pts] You have 2 different dice that are not evenly weighted:

   - Dice 1 has sides $\{1, 2, 3\}$ and a 10% chance of rolling a 1, a 40% chance of rolling a 2 and a 50% chance of rolling a 3.
   - Dice 2 has sides $\{2,2,3,3,3,4,4\}$ with a 20% chance for each 2, a 10% chance for each 3 and a 15% chance for each 4.
   - Set up the table for the dynamic programming algorithm and fill in the complete column for Dice 1 and Dice 2.
   - What is the probability of rolling a 5 with these dice?

   Solution:

	Die#1	Die #2	Die #1,#2
1	10%	0	0
2	40%	40%	0
3	50%	30%	4%
4	0	30%	19%
5	0	0	35%
6	0	0	27%
7	0	0	15%
Sum	1	1	1

   The probaility of rolling a 5 with these dice is 35%.

4. [8 pts] Consider the heapify algorithm for creating a heap from an array of random integers:

   (a) How many swaps(maximum) may be required for an array of 3 integers?

   Solution: 1

   (b) How many swaps(maximum) may be required for an array of 7 integers?

Solution: 4

(c) How many swaps(maximum) may be required for an array of 15 integers?

Solution: 11

(d) How many swaps(maximum) may be required for an array of 31 integer?

Solution: 26

5. [8 pts] Consider the following NP completeness questions.

   i. Assume you can solve an NP-Complete problem in polynomial time and mark the following as "true" or "false" with this assumption:
      - All P problems can be solved in polynomial time?
        Solution: true
      - All NP problems can be solved in polynomial time.
        Solution: true
      - All NP-Complete problems can be solved in polynomial time.
        Solution: true
      - All NP-Hard Problems can be solved in polynomial time.
        Solution: false

   ii. At least 1 NP problem can be solved in polynomial time? (True or False)
   Solution: True

   iii. NP-Complete problems are in P("true" "false" or "unknown")
   Solution: unknown

   iv. Which NP-Hard Problems are also NP-Complete? ("some" "all" "none" or "unknow")
   Solution: some

6. [10 pts] Consider an RSA encryption system that has a public key of 479767 for the value e and 561233 for the value of the modulus N. You also saw a message that had been encrypted by the public key. The value of this encrypted message is 3.

   (a) You are able to factor N = 561233 into the product of two prime numbers 677 * 829. What is the value of the private key? Show your work including the table for computing the Extended Euclidean Algorithm.

   Solution:
   public key: (e,n) = (479767, 561233)
   private key: (d,n)

$$d = \frac{1}{e} \bmod \Phi(n) = \frac{1}{479767} \bmod \Phi(561233)$$
$$\Phi(561233) = \Phi(677 \times 829) = (677-1) \times (829-1) = 676 \times 828 = 559728$$
$$\therefore d = \frac{1}{479767} \bmod 559728$$

	A	B	Q	R	α	β
-1					1	0
	559728	479767	1	79961	0	1
	479767	79961	6	1	1	-1
	79961	1	79961	0	-6	7
	79961	0	-	-	479767	-559728

$$-(6) \times (559728) + 7 \times 479767 = 1$$
$$(-6 \times 559728) \bmod 559728 + (7 \times 479767) \bmod 559728 = 1$$
$$(7 \times 479767) \bmod 559728 = 1$$
$$\because (\frac{1}{479767} \times 479767) \bmod 559728 = 1$$
$$\therefore d = 7 \bmod 559728 = 7$$

private key: (7, 561233)

(b) What was the message before it was encrypted (Give an integer)

Solution:

$$3^7 \bmod 561233 = 2187$$

7. [8 pts] Set up the table to find the longest increasing sub-sequence of the following sequence: 4, 6, 9, 5, 7, 8, 11, 2, 3, 13

Solution:

4	4									
6	4	6								
9	4	6	9							
5	5	6	9							
7	3	6	7							
8	3	6	7	8						
11	3	6	7	8	11					
2	2	6	7	8	11					
3	3	6	7	8	11					
13	3	6	7	8	11	13				

The longest incraseing subsequence is 4, 5, 7, 8, 11, 13

8. [9 pts] Consider the Levensthein Edit Distance for two strings A and B.

(a) Write the equation describing what you would put in the table for location T[i,j].

Solution:

```
// Base case
if (i == 0){
    T[i, j] = T[0,j];
}
if (j == 0){
    T[i, j] = T[i,0];
}

if (Ai == Bj){
    T[i,j] = min{T[i-1,j]+1, T[i,j-1]+1, T[i-1,j-1]};
}else{
    T[i,j] = min{T[i-1,j]+1, T[i, j-1]+1, T[i-1, j-1]+1};
}
```

(b) How would you modify this equation for a different version of the Levensthein Edit Distance where substituion is not allowed?

Solution:

```
// Base case
if (i == 0){
    T[i, j] = T[0,j];
}
if (j == 0){
    T[i, j] = T[i,0];
}

if (Ai == Bj){
    T[i,j] = min{T[i-1,j]+1, T[i,j-1]+1, T[i-1,j-1]};
}else{
    T[i,j] = min{T[i-1,j]+1, T[i, j-1]+1};
}
```

(c) Fill in the following table for finding the regular, unmodified "Levensthein Edit Distance" for two strings, M and N

M = L B B Y C N = L Z B C Y Y

9. [6 pts] You have two strings; String A and String B.

- The Levensthein Edit Distance between the strings is 9.
- The Longest Common Subsequence between the strings is 5.
- The length of String A is $<$ the length of String B

(a) What is the minimun length of String A?

Solution: 5

(b) If String A has a length of 15, what is the minimum length of string B?

(c) If String A has a length of 15, what is the maximum length of string B?

Solution: (b)(c): String A can not hold len=15 with LCS =5, LED =9; 14 is max

10. [6 pts] You know that problem C is NP-Complete and you want to use that to prove that problem A is NP-Complete. What two things must you show to do this?

11. [6 pts] Give an argument that sorting an array of integer is just as hard and posiibly harder than creating a Heap of that array of integers

Solution: By sorting the integers(indexed) it creates a heap. Since a solver for sorting solves the heapify problem sorting must be just as hard as harder than heapify.

12. [8 pts] Consider the following LCS problem

(a) Fill in the following table for finding the longest common subsequence for two strings, M and N

M = L B B Y C N = L Z B C Y Y

The Shortest Common Supersequence is the shortes sequence that contains both the string M as a subsequence and the string N as a subsequence. The following would be examples:

Example 1: L B Z B Y C Y Y Example 2: L Z B B Y C Y Y

Solution:

	-	L	B	B	Y	C
-	0	0	0	0	0	0
L	0	1	1	1	1	1
Z	0	1	1	1	1	1
B	0	1	2	2	2	2
C	0	1	2	2	2	3
Y	0	1	2	2	3	3
Y	0	1	2	2	3	3

(b) Given the length of string M, $|M|$ the length of string N, $|N|$ and the length of the longest common subsequence, $|LCS|$, write and equation for the length of the shortest common supersequence?

Solution:

$$|M| + |N| - |LCS|$$

(c) How can you use your solution for the longest common subsequence to determine the shortes common supersequence?

Solution: Use the LCS as 'anchor points" and fill in between anchor points. For example:

AxByC
AmBnC
$\rightarrow$ AxmBynC

13. Consider the following problem:

(a) [2 pts] How many swaps may be required (maximum) to heapify an array of size $2^n - 1$ integers? (You may write a summation for this)

Solution:

$$n(\frac{0}{2} + \frac{1}{4} + \frac{2}{8} + \frac{3}{16} + \frac{4}{32} + \dots)$$

(b) [3 pts] Setup the table for the extended Euclidian algorithm and compute

- $\frac{1}{21}$ modulo 98

Solution:

	A	B	Q	R	α	β
-1					1	0
	98	21	4	14	0	1
	21	14	1	7	1	-4
	14	7	2	0	-1	5
	7	0	-	-	3	-14

Does not exist since GCD(98,21) = 7 not 1