CSE 450 Assignment 3

14th October, 2022

Submission Instructions: Deadline is **11:59pm on 10/21/2022**. Late submissions will be penalized, therefore please ensure that you submit (file upload is completed) before the deadline. Additionally, you can download the submitted file to verify if the file was uploaded correctly. **Please TYPE UP YOUR SOLUTIONS and submit a PDF** electronically, via *Canvas*. Furthermore, please note that the graders will grade 2 out of the 4 questions randomly. Therefore, if the grader decides to check questions 1 and 4,and you haven't answered question 4, you'll lose points for question 4. Hence, please answer all the questions.

1. Let u and v be two n bit numbers where for simplicity n is a power of 2. The traditional multiplication algorithm requires $O(n^2)$ operations. A divide and conquer based algorithm splits the numbers into two equal parts, computing the product as

$$uv = (a2^{\frac{n}{2}} + b)(c2^{\frac{n}{2}} + d)$$
$$= ac2^n + (ad + bc)2^{\frac{n}{2}} + bd$$

   The multiplications ac, ad, bc, and bd are done using this algorihtm recuresiverly. **[25 points]**

   (a) Determine the computing time of the above algorithm

   (b) What is the computing time if $ad + bc$ is computed as $(a + b)(c + d) - ac - bd$?

2. Find the longest common subsequence (as described in class and the slides) of the following two sequences by Dynamic Programming Algorithm. Show all your work. **[25 points]**

$$X = ABCBDAB, \quad Y=BDCABA$$

3. Find the optimal order of multiplying four matrices whose sequence of dimensions is (13, 5, 89, 3, 34) (i.e. the first matrix is 13 x 5, the second matrix is 5 x 89, and so on) and also the fewest number of multiplications needed to find the resulting matrix. Show all the steps of computation. **[25 points]**

4. Let A_1, ..., A_n be the matrices where the dimensions of A_i are d_{n-1} x d_i, for i = 1, ..., n. Here is a strategy for determining the best order in which to perform the matrix multiplications to compute A_1 x A_2 x .... x A_n : At each step, choose the largest remaining dimension (from among d_1, ... d_{n-1}), and multiply two adjacent matrices which share that dimension.. [**25 points**]

(a) What is the order of the running time of this algorithm (only to determine the order in which to multiply the matrices, not including the actual multiplications)?

(b) Either give a convincing argument that this strategy will always minimize the number of multiplications, or give an example where it fails to do so.