

# CSE 551 Assignment 3

March 10, 2021

**Submission Instructions:** Deadline is **11:59pm on 03/16**. Late sub-missions 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. Submit your answers electronically, in a single PDF, via *Canvas*. **Please type up the answers and keep in mind that we'll be checking for plagiarism.**

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. 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. Furthermore, compute the fewest number of multiplications needed to find the resulting matrix. Show all the steps of your computation. [25]
2. Let  $A_1, A_2, \dots, A_n$  be matrices where the dimensions of  $A_i$  are  $d_{i-1} * d_i$  for  $i = 1, \dots, n$ . Here is a strategy to compute the best order (i.e., the order that requires the fewest number of multiplications) in which to perform the matrix multiplications to compute  $A_1, A_2, \dots, A_n$ . At each step, choose the largest remaining dimension (from among  $d_1, \dots, d_{n-1}$ ) and multiply two adjacent matrices that share the same dimension. State with reasoning whether the above strategy will always minimize the number of multiplications necessary to multiply the matrix chain  $A_1, \dots, A_n$  (i.e., if you think that this will result in the optimal number of multiplications, then please provide arguments as to why you think so, else, provide a counter example). [25]
3. Using Dynamic Programming technique, find the optimal solution to the Traveling Salesman Problem for the following data set (Distance Matrix).

$$M = \begin{bmatrix} 0 & 2 & 5 & 9 \\ 10 & 0 & 2 & 8 \\ 3 & 8 & 0 & 6 \\ 9 & 2 & 6 & 0 \end{bmatrix}$$

**Show all your work.** Also show the node ordering in the optimal tour. [25]

4. A subset of nodes in  $S \subset V$  is a “Special Node Set (SNS)” of a graph  $G = (V, E)$ , if there are no edges between the nodes in  $S$ . The SNS of the largest cardinality is referred to as “Largest Special Node Set (LSNS)”.

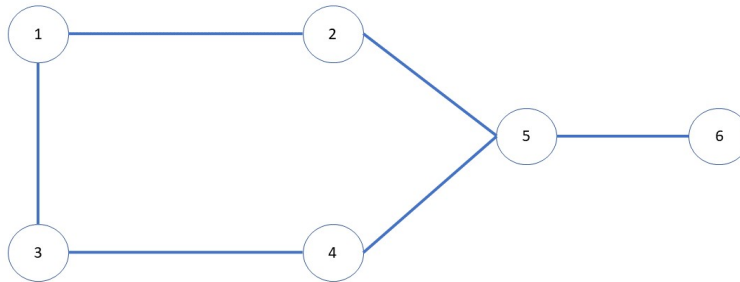


Figure 1: Figure for Q4

Design a Dynamic Programming based algorithm to compute the LSNS of a tree. Write down the recurrence relation on which your algorithm is based and show how the recurrence relation is being used by your algorithm to compute the LSNS of the tree. Analyze your algorithm to find it's complexity. **Show all our work.** Can your algorithm be used to find the LSNS of a graph which isn't a tree? If your answer is “yes”, explain how it can be used, otherwise, explain why it can't be used to compute the LSNS of a graph which isn't a tree. [25]