# Assignment 4

COMP 3804, Fall 2021

Upload in Brighspace by 11:59PM on December 6, 2021

## 1 Guidelines

General guidelines are as follows:

1. We will be discussing solutions to this assignment in Office Hours of December 7th and 9th and hence **no late submissions** will be entertained.
2. Please write clearly and answer questions precisely. It is your responsibility to ensure that what is uploaded is clearly readable. If we can't read, we can't mark!
3. Please cite all the references (including web-sites, names of friends, etc.) which you used/consulted as the source of information for each of the questions.
4. All questions/problems carry equal marks.
5. When a question asks you to design an algorithm - it **requires** you to
   (a) Clearly spell out the **steps** of your algorithm in pseudo code.
   (b) **Prove** that your algorithm is correct
   (c) **Analyze** the running time.
6. You can assume that a graph $G = (V, E)$ uses adjacency list representation.
7. $n$ is a positive integer, and it typically represents the size of the input to a problem.

# 2 Problems

1. We are given a sequence of $n$ integers, $a_1, \ldots, a_n$, some of which may be negative. For a contiguous subsequence $a_i, \ldots, a_j$, where $1 \le i \le j \le n$, define $\Delta[i,j] = a_i + \cdots + a_j$. In $O(n)$ time, determine a pair of indices $(i,j)$, where $1 \le i \le j \le n$, such that $\Delta[i,j] \ge \Delta[\alpha,\beta]$ for any choice of $\alpha, \beta$, where $1 \le \alpha \le \beta \le n$. (Hint: Think of dynamic programming and consider the subsequence ending at $j$ that maximizes $\Delta[i,j]$ for each choice of $j \in \{1, \ldots, n\}$. )

2. Assume that $O(pqr)$ number of operations (multiplications and additions) are required to compute the product $PQ$ of two matrices $P$ of dimension $p \times r$ and $Q$ of dimension $r \times q$. Note that the resulting product matrix has dimension $p \times q$ (i.e., $p$ rows and $q$ columns). As input, we are given six matrices $A, B, C, D, E, F$ and their dimensions are as follows:
   - $A$ is $5 \times 10$,
   - $B$ is $10 \times 3$,
   - $C$ is $3 \times 12$,
   - $D$ is $12 \times 5$,
   - $E$ is $5 \times 50$, and
   - $F$ is $50 \times 6$.

   What is the least number of operations required to compute the product $ABCDEF$? Justify your answer. (Hint: See the video lecture/notes.)

3. Let $T = (V, E)$ be a tree. Its vertex set is $V$, and edge set is $E$. We say $X \subset V$ is a cover of $T$, if for any edge $e = (uv) \in E$, $u \in X$ or $v \in X$. Design an algorithm, running in polynomial-time, to find a cover of the minimum size of a given tree $T$. (Hint: Assume $T$ is rooted at vertex $r$. For each vertex $v$, define $C(v,0)$ and $C(v,1)$, that indicates the size of the cover when $v$ is excluded or included in the cover, respectively. Set up the recurrence relation for $C(v,*)$ taking into account the $C$ values of each of its children. Note that we are interested in reporting $\min(C[r,0], C[r,1])$.)

4. Let $A$ be a $m \times n$ matrix where each element is 0 or 1. We are interested in finding the largest square sub-matrix of $A$ such that each of its elements is 1. Design a dynamic-programming algorithm, running in $O(mn)$ time, that finds such a largest square sub-matrix in $A$. For example, let $A = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$.
   In this case, the $3 \times 3$ square submatrix formed by columns $2, 3, 4$ and rows $4, 5, 6$ of all 1s should be returned by the algorithm.

5. You are given a set of $n$ positive numbers $A = \{a_1, \ldots, a_n\}$ and a positive integer $t$. Design a dynamic programming algorithm running in $O(nt)$ time that decides whether

there exists a subset $A' \subseteq A$ such that $\sum_{x \in A'} x = t$. Note that each element of $A$ can be used at most once.

6. In your own words describe the following
   (a) Complexity class $P$. Also, present a couple of examples.
   (b) Complexity class $NP$. Also, present a couple of examples.
   (c) Polynomial-time reducibility
   (d) What steps are involved in showing a decision problem $L \in P$.
   (e) What are the steps involved in showing a decision problem $L \in NP$-Hard.
   (f) What are the steps involved in showing a decision problem $L \in NP$-Complete.

7. Let $G = (V, E)$ be a simple connected undirected graph. We are given a subset $L \subseteq V$. We want to decide if there is a spanning tree of $G$ such that its set of leaves includes the set $L$. Is this decision problem in $P$? Is it in $NP$? Is it $NP$-Complete?

8. Consider the following decision problem on satisfying inequalities. Let $A$ be an integer $m \times n$ matrix. Let $b$ be a vector of length $m$ where each coordinate is an integer. You need to decide whether there exists a vector $x$ of length $n$ such that each of its coordinates is 0 or 1 and $Ax \leq b$. For example, let $A = \begin{bmatrix} 1 & 0 & -3 \\ 1 & -1 & 1 \end{bmatrix}$ and $b = \begin{bmatrix} -2 \\ 3 \end{bmatrix}$. Then $x = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$, satisfies $Ax \leq b$. Show that the problem of satisfying (integer) inequalities is $NP$-Complete. (Hint: Provide a polynomial-time reduction from 3CNF-SAT problem.)