Computer Science C73 October 21, 2026
Scarborough Campus University of Toronto

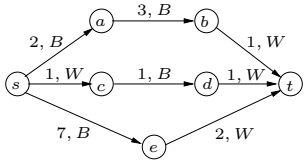
Homework Assignment #4
Due: November 4, 2026, by 11:59 pm

- ***You must submit your assignment through the Crowdmark system.*** *You will receive by email an invitation through which you can submit your work in the form of separate PDF docucments with your answers to each question of the assignment. To work with a partner, you and your partner must form a group on Crowdmark.* ***The group must be formed before either partner submits <u>any</u> answer.***
- *To minimize bias in grading, Crowdmark does not reveal your name to the grader. Please do not subvert this feature by including your name(s) on the files you submit.*
- *The PDF files you submit must be legible without undue effort by the grader. To this end, I encourage you to learn and use the LaTex typesetting system, which is designed to produce high-quality documents that contain mathematical notation. You are not required to produce the PDF files you submit using LaTex; you may produce them any way you wish, as long as the resulting document is legible.*
- *By virtue of submitting this assignment you (and your partner, if you have one) acknowledge that you are aware of the policy on homework collaboration for this course.*[a]
- *For any question, you may use data structures and algorithms previously described in class, or in prerequisites of this course, without describing them. You may also use any result that we covered in class, or is in the course notes by referring to it.*
- *Unless we explicitly state otherwise, you may describe algorithms in high-level pseudocode or point-form English, whichever leads to a clearer and simpler description. Do not provide executable code.*
- *Unless we explicitly state otherwise, you should justify your answers. Your paper will be graded based on* ***the correctness and efficiency of your answers, and the clarity, precision, and conciseness of your presentation****.*

[a] "For help with your homework you may consult only the instructor, TAs, your homework partner (if you have one), the CSCC73 Course Notes, and your personal class notes. You may not consult any other source, including AI-based tools."

Question 1. (15 marks) We are given a directed graph $G = (V, E)$ and two distinguished nodes $s, t \in V$. Each edge e has a positive weight $\mathbf{wt}(e)$ and a colour $\mathbf{col}(e) \in \{\text{black}, \text{white}\}$. The ***coloured-weight*** of a path in G is the sum of the weights of the edges plus the fixed amount 5 <u>each time</u> the path changes colour — i.e., it leaves a node with an edge of the opposite colour than the edge with which it enters the node.

For example, the colour-length of the three $s \rightsquigarrow t$ paths in the graph shown to the right (where the label 'x, y' of an edge indicates its weight x and color y) are: $2 + 3 + 5 + 1 = 11$ (for s, a, b, t), $1 + 5 + 1 + 5 + 1 = 13$ (for s, c, d, t), and $7 + 5 + 2 = 14$ (for s, e, t).



Given as input the graph G, the nodes s, t, and the edge weight and colour functions $\mathbf{wt}$ and $\mathbf{col}$, we want to find the minimum coloured-weight of an $s \to t$ path in G. Give an efficient algorithm to solve this problem by reducing it to a shortest-paths problem. Briefly justify the correctness of your algorithm (by noting the key relevant property that your reduction ensures) and analyze its running time.

Question 2. (15 marks) Design an efficient algorithm that takes as input a directed graph $G = (V, E)$, a node $s \in V$, a positive integer k, and an edge weight function $\mathbf{wt}$: $E \to \mathbb{Z}$, and returns, for every node u, the weight of a shortest $s \rightsquigarrow u$ path *with exactly k edges* if one exists (or ∞, if no such path exists). Justify the correctness of your algorithm and analyze its running time.

Note: We are not seeking the weight of a shortest *simple* $s \rightsquigarrow u$ path with exactly k edges; that additional restriction would make the problem NP-hard! (Those of you who have taken CSCC63 should recognize the Hamiltonian path problem hidding as a special case of this, apparently harder, version of the problem.) In particular, if $k \geq |V|$, any path with exactly k edges is necessarily *not* simple.

Question 3. (15 marks) Let $G = (V, E)$ be a directed graph and $\mathbf{wt}: E \to \mathbb{Z}$ be an edge-weight function, such that every cycle of G has positive weight (but note that the graph may have negative- or zero-weight edges). Modify the Floyd-Warshall algorithm to obtain an $O(n^3)$ algorithm that, given G and $\mathbf{wt}$, returns the array $N[u, v]$ for each pair of nodes $u, v \in V$ such that $N[u, v]$ is the number of minimum-weight $u \to v$ paths. Justify the correctness of your algorithm. (You don't need to analyze its running time, but it should be $O(n^3)$.)

Note: We want all cycles to have positive (rather than merely non-negative) weight because with zero-weight cycles the minimum weight of $u \to v$ paths is well-defined but the number of such paths may be infinite.