Computer Science C73 | November 22, 2023
Scarborough Campus | University of Toronto

Homework Assignment #8
Due: MONDAY December 4, 2023, by 11:59 pm (note unusual due day)

• ***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. Crowdmark does not enforce a limit on the size of groups.* ***The course policy that limits the size of each group to at most two remains in effect:*** *submissions by groups of more than two persons will not be graded.*
• *It is your responsibility to ensure that the PDF files you submit are legible. 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 it 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 assigned sections of the official course textbooks, 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] "In each homework assignment you may collaborate with at most one other student who is currently taking CSCC73. If you collaborate with another student on an assignment, you and your partner must submit only one copy of your solution, with both of your names. The solution will be graded in the usual way and both partners will receive the same mark. Collaboration involving more than two students is not allowed. **For help with your homework you may consult only the instructor, TAs, your homework partner (if you have one), your textbook, and your class notes. You may not consult any other source.**"

Question 1. (25 marks) A new semester is about to dawn and the Department of Commuter and Metaphysical Sciences (CMS) has to deal once again with the problem of teaching assistant (TA) assignments.

A set S of students have applied for TA positions in the set C of courses offered by CMS next semester. Each course $c \in C$ has a set of tutorials T_c. Let $T = \cup_{c \in C} T_c$ be the set of all tutorials of CMS courses; the sets of tutorials of different courses are disjoint. For each applicant $s \in S$ there is a known set of courses Q_s for which s is qualified to serve as a TA. A ***TA assignment*** A is a set of pairs (s, t), where $s \in S$ and $t \in T$, that satisfies requirements (1)-(3) below. The pair (s, t) indicates that applicant s has been assigned to tutorial t.

(1) Each tutorial has at most one applicant assigned to it.

(2) Each applicant $s \in S$ is assigned to at most 3 tutorials, possibly of different courses.

(3) If an applicant s is assigned to a tutorial of course c, then s is qualified for c (i.e., $c \in Q_s$).

CMS hires you to develop algorithms to find TA assignments that satisfy certain optimality criteria. The input to your algorithms consists of:

- The set of applicants S.
- The set of courses C.
- For each applicant $s \in S$, the set of courses Q_s for which s is qualified; this could be empty.
- For each course $c \in C$, the set of tutorials T_c of course c; this could be empty and the sets of tutorials of different courses are disjoint.

a. The first algorithm you must develop finds a TA assignment that ***maximizes the number of tutorials that have a TA assigned to them***.

By reduction to the maximum flow problem give a polynomial-time algorithm that determines such an assignment. The polynomial depends on $\ell = |S|$, $m = |C|$, and $n = |T|$. Justify the correctness of your algorithm and analyze its running time.

b. We say that a course c is ***fulfilled*** by a TA assignment A if ***every*** tutorial of c has a TA assigned to it in A. The second algorithm you must develop finds a TA assignment that ***maximizes the number of courses that are fulfilled***.

Formulate this problem as a 0-1 linear program, and explain how from an optimal solution to this program (i.e., an assignment of 0 or 1 to its variables that satisfies the constraints and optimizes the objective function) you can obtain the desired TA assignment. Explain the role of your program's variables, and how the constraints capture the above requirements. State the number of variables and constraints of your 0-1 LP; these, and the coefficients involved in the constraints and the objective function of your 0-1 LP, must be bounded by a polynomial of the input size (the relevant parameters are $\ell = |S|$, $m = |C|$, and $n = |T|$

Question 2. (10 marks) We are given the sequence of "weights" of n items, $w_1, w_2, \ldots, w_n$, where $0 < w_i \leq 1$ is the weight of item i, $1 \leq i \leq n$. We want to place these n items into the smallest possible number of identical containers, each of capacity 1. That is, the sum of the weights of the items placed on any container must not exceed 1. For example, suppose the weights are $0.5, 0.25, 0.75, 0.5, 0.75$. Then the minimum number of containers in which to place these items is three.

Finding the minimum number of containers is a well-known NP-hard problem.

Consider the following greedy heuristic for this problem: Suppose the containers are numbered $1, 2, \ldots$; these are initially empty. Consider the items in the order given. When considering item i, place it in the first container that (together with the items previously placed in it) can accommodate the item without exceeding its capacity; that is, place i in container j, where j is the smallest positive integer such that the weight of the items presently in container j plus w_i is at most 1. Applied to the example above, this heuristic places items 1 and 2 in the first container, and each of the remaining three items in a separate container — i.e., it uses four containers instead of the optimal three.

Prove that this heuristic is a 2-approximation algorithm. That is, for any input $w_1, w_2, \ldots, w_n$, where $0 < w_i \leq 1$, if the minimum possible number of containers for items with these weights is c^* and the number of containers used by the above greedy algorithm is c, then $c \leq 2c^*$.

Hint: (1) Note that $c^* \geq \lceil \sum_{i=1}^{n} w_i \rceil$. (You should explain why this is true.) (2) In the assignment of items to containers made by the greedy algorithm, how many containers can be at most half-full? (Prove your answer.)

Note: Two is not a tight bound for this greedy approximation algorithm: It can be shown that the algorithm always does better than twice the optimal, but the proofs of better bounds are more complicated.

THAT'S IT WITH HOMEWORK, FOLKS!