Computer Science C73 — November 24, 2021
Scarborough Campus — University of Toronto

Homework Assignment #8
(worth 4% of the course grade)
Due: December 6, 2021, 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. 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.*
• *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. (20 marks) A library has a set of subscribers S and owns a set of books B. Each subscriber $i \in S$ is interested in borrowing a subset $B_i \subseteq B$ of the library's books. The library owns b_j copies of each book $j \in B$. The set of books is partitioned into the sets F and N of fiction and nonfiction books.

The rules of the library are that a subscriber may borrow up to 12 copies of books in each lending period (say, a month), at most 8 of which can be in each of the two categories, fiction and nonfiction, and no two of which can be copies of the same book.

An ***assignment*** is a set of pairs $(i, j) \in S \times (F \cup N)$, indicating that subscriber i will borrow a copy of book j. An assignment is ***feasible*** if it satisfies the above rules of the library, it does not exceed the supply of copies of any book, and assigns to subscribers only books in which they are interested in borrowing.

You are hired to design algorithms to determine an assignment for the next lending period. Your algorithms take as input

- the set of subscribers S,
- the sets of fiction and nonfiction books F and N that the library owns,
- the number of copies b_j of each book $j \in F \cup N$ that it owns, and
- the set $B_i \subseteq F \cup N$ of books that each subscriber $i \in S$ is interested in borrowing,

and output a feasible assignment with the optimality requirement specified in each part below.

a. The library receives from the government a subsidy of \$2 per copy of book lent to a subscriber. Describe an algorithm that outputs a feasible assignment that ***maximizes the total government subsidy that the library receives***. Your algorithm ***must*** be by reduction to the maximum flow problem. Your answer can include (but should not consist only of) a helpful diagram. Explain why your algorithm works, and analyze its running time.

b. The government changes its policy and now subsidizes the library with a flat rate, regardless of how many copies of books are borrowed; thus, maximizing the number of copies of books borrowed is no longer important. To encourage reading, however, the library awards a special certificate to so-called ***star subscribers***, i.e., subscribers who borrow at least 10 books.

Define a 0-1 LP using which you can determine a feasible assignment that ***maximizes the number of star subscribers***. Describe the variables, their meaning, the objective function, and the constraints. Explain how and why the optimal solution to your 0-1 LP determines a feasible assignment that maximizes the number of star subscribers.

Hint. Among the variables you define, include a variable y_i for each subscriber i, whose intended semantics is that $y_i = 1$ if i is a star subscriber, and $y_i = 0$ otherwise. Then introduce constraints (relating this variable to other variables) so that the value of y_i satisfies these semantics.

Question 2. (15 marks) Consider the following problem.

INPUT: A set $A = \{a_1, \ldots, a_n\}$ of positive integers, and a positive integer b.
OUTPUT: A subset X of A such that the sum of the elements in X is maximum but does not exceed b.

This is an NP-hard problem.

a. Reduce this problem to 0-1 LP.

b. Consider the following greedy algorithm.

```
GreedySum(A, b)
X := ∅
total := 0
for i := 1..n do
    if total + a_i ≤ b then
        X := X ∪ {a_i}
        total := total + a_i
return X
```

Prove that this is not a c-approximation algorithm for the problem, for any constant $c > 1$. That is, for every $c > 1$, there is an input for which the algorithm outputs X, the optimal output is Y, and the sum of the elements in X is less than $1/c$ of the sum of the elements in Y.

c. Give a 2-approximation greedy algorithm for the problem. That is, for every input (A, b), if X is the output of your algorithm and Y is the optimal output for that input, then the sum of the elements in X is at most b and at least one-half the sum of the elements in Y. Prove that your algorithm is, indeed, a 2-approximation algorithm for the problem, and analyze its running time.

THAT'S IT WITH HOMEWORK, FOLKS!