COL 351 : Analysis and Design of Algorithms
Semester I, 2021-22, CSE, IIT Delhi

Assignment - 2 (due on 17th September, 11:00 PM)

Important Guidelines:

- Each assignment must be done in a group of size at most two.
- Handwritten submissions will not be accepted. Solutions must be typed-up (in Latex, Microsoft Word, etc.), and submitted in pdf format. Each solution must start on a new page.
- **Your answer to each question must be formal and have a proper correctness proof.** No marks will be granted for vague answers with intuition or for algorithms without proof. You must be very rigorous in providing mathematical detail in support of your arguments.
- Cheating of any form will lead to strict penalty.

1 Algorithms Design book

Alice, Bob, and Charlie have decided to solve all exercises of the Algorithms Design book by Jon Kleinberg, Éva Tardos. There are a total of n chapters, $[1, \ldots, n]$, and for $i \in [1, n]$, x_i denotes the number of exercises in chapter i. It is given that the maximum number of questions in each chapter is bounded by the number of chapters in the book. Your task is to distribute the chapters among Alice, Bob, and Charlie so that each of them gets to solve nearly an equal number of questions.

Device a polynomial time algorithm to partition $[1, \ldots, n]$ into three sets S_1, S_2, S_3 so that $\max\{\sum_{i\in S_1} x_i, \sum_{i\in S_2} x_i, \sum_{i\in S_3} x_i\}$ is minimized. [15 marks]

2 Course Planner

You are given a set C of courses that needs to be credited to complete graduation in CSE from IITD. Further, for each $c \in C$, you are given a set $P(c)$ of prerequisite courses that must be completed before taking the course c.

1. Device the most efficient algorithm to find out an order for taking the courses so that a student is able to take all the n courses with the prerequisite criteria being satisfied, if such an order exists. What is the time complexity of your algorithm? [5 marks]

2. Device the most efficient algorithm to find minimum number of semesters needed to complete all n courses. What is the time complexity of your algorithm? [5 marks]

3. Suppose for a course $c \in C$, $L(c)$ denotes the list of all the courses that must be completed before crediting c. Design an $O(n^3)$ time algorithm to compute a pair set $P \subseteq C \times C$ such that for any $(c, c') \in P$, the intersection $L(c) \cap L(c')$ is empty. [5 marks]

3 Forex Trading

Suppose you are a trader aiming to make money by taking advantage of price differences between different currencies. You model the currency exchange rates as a weighted network, wherein, the nodes correspond to n currencies - $c_1, \ldots, c_n$, and the edge weights correspond to exchange rates between these currencies. In particular, for a pair (i, j), the weight of edge (i, j), say $R(i, j)$, corresponds to total units of currency c_j received on selling 1 unit of currency c_i.

1. Design an algorithm to verify whether or not there exists a cycle $(c_{i_1}, \ldots, c_{i_k}, c_{i_{k+1}} = c_{i_1})$ such that exchanging money over this cycle results in positive gain, or equivalently, the product $R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_{k-1}, i_k] \cdot R[i_k, i_1]$ is larger than 1. [9 marks]

   (Hint: Use the fact that a number x is strictly larger than 1 if and only if $\log(1/x) < 0$).

2. Present a cubic time algorithm to print out such a cyclic sequence if it exists. [6 marks]

4 Coin Change

You are given a set of k denominations, $d[1], \ldots, d[k]$. Example for Rs. 1, 2, 5, 10, 20, 50, 100, you have d(1) = 1, d(2) = 2, d(3)=5, d(4) = 10, d(5)=20, d(6)=50, and d(7)=100.

1. Device a polynomial time algorithm to count the number of ways to make change for Rs. n, given an infinite amount of coins/notes of denominations, $d[1], \ldots, d[k]$. [7 marks]

2. Device a polynomial time algorithm to find a change of Rs. n using the minimum number of coins (again you can assume you have an infinite amount of each denomination). [8 marks]