NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES ISLAMABAD CAMPUS

Advanced Analysis of Algorithms (CS5005) – Fall 2021

ASSIGNMENT-1

Due Date: 15th October 2021. Submission in google classroom

Problem 1 [10 marks]: You have ***N*** problem sets (psets) due right now, but you haven't started any of them, so they are all going to be late. Each pset requires d_i days to complete, and has a cost penalty of c_i per day. So, if pset ***i*** ends up being finished ***t*** days late, then it incurs a penalty of $t \cdot ci$. Assume that once you start working on a pset, you must work on it until you finish it, and that you cannot work on multiple psets at the same time.

For example, suppose you have three problem sets: 1st takes 3 days and has a penalty of 12 points/day, 2nd takes 4 days and has a penalty of 20 points/day, and 3rd takes 2 days and has a peanlty of 4 points/day. The best order is then 2nd, 1st, 3rd which results in a penalty of $20 \cdot 4 + 12 \cdot (4 + 3) + 4 \cdot (3 + 4 + 2) = 200$ points.

Give a greedy algorithm that outputs an ordering of the psets that minimizes the total penalty for all the psets. Analyze the running time and prove correctness.

Problem 2 [15 marks]: Prof. John is cooking from his lawn, that is arranged in rectangular form with **x** rows and **y** columns. Each cell (i, j) (x >= i >=1, y >= j >= 1) in lawn has an components growing in it, with savor value given by a positive value $S_{i,j}$. Prof. John don't like cooking "by the book". To prepare dinner, he will stand at a cell (i, j) and pick one component from each quadrant relative to that cell. The savor of his dish is the product of the savor of the four components he chooses. Here the four quadrants relative to a cell (i, j) are defined as follows:

top-left = {all cells (a, b) | a < i, b < j},

bottom-left = {all cells (a, b) | a > i, b < j},

top-right = {all cells (a, b) | a < i, b > j},

bottom-right = {all cells (a, b) | a > i, b > j}.

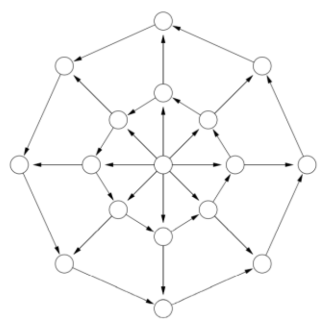
Because Prof. John needs all four quadrants to be non-empty, she can only stand on cells (i, j) where 1 < i < x and 1 < j < y. Help Prof. John find an **O(xy)** dynamic programming algorithm to maximize the tastiness of her dish.

Problem 3 [15 marks]: Consider, we have **N** books and we want to store these in the shelfs in library. The order of books is fixed by the catalog system and it cannot be changed. Therefore, we can speak of a book $\mathbf{b_i}$, where $1 \le \boldsymbol{i} \le n$, that has a thickness $\mathbf{t_i}$ and height $\mathbf{h_i}$. The length of each bookshelf at this library is **L**.

(a) Suppose all the books have the same height **h** (i.e., h = hi = hj for all i, j) and the shelves are all separated by a distance of greater than **h**, so any book fits on any shelf. The greedy algorithm would fill the first shelf with as many books as we can until we get the smallest i such that bi does not fit, and then repeat with subsequent shelves. Show that the greedy algorithm always finds the optimal shelf placement, and analyze its time complexity.

(b) Give an algorithm for this problem, and analyze it's time complexity.

Problem 4 [10 marks]: A wheel graph is a directed graph of the following form, i.e., a wheel graph consists of a center vertex **c** with **k** outgoing 'spokes of **s** outward oriented edges at each circle; furthermore, all the spokes at each circle are connected to form a directed cycle, and all cycles are oriented the same way (k = 8 and s = 2 for the following figure)



(a) What are the number of edges in the wheel-graph as a function of the number vertices n?

(b) Assume we have assigned integer edge-weights and want to find the shortest path from **c** to all other vertices. How long time the Dijkstra's algorithm takes to solve the problem (as a function of n)?

Problem 5 [10 marks]: Let G = (V, E) be a connected, undirected graph with edge-weight function $w : E \to \mathbb{R}$, and assume all edge weights are distinct. Consider a cycle $(v_1, v_2, \ldots, v_k, v_{k+1})$ in G, where $v_{k+1} = v_1$, and let (v_i, v_{i+1}) be the edge in the cycle with the largest edge weight. Prove that (v_i, v_{i+1}) does ***not*** belong to the minimum spanning tree T of G.