Term 2, 2021 COMP3121/9101: Assignment 2

You have five problems, marked out of a total of 100 marks; each problem is worth 20 marks. You should submit your solutions by Monday, July 12. Please do not wait till the last moment to submit your work - we WILL NOT accept emailed solutions regardless of whether you had network problems or not. Also follow closely the instructions for how to submit your solutions. Your solutions to each of the problems must be submitted in a separate file. There are 1000+ students in this class and thus NO EXCEPTIONS can be granted except for special considerations or for ELS students who can get one week extension. Your solutions must be typed, machine readable .pdf files. All submissions will be checked for plagiarism!

1. You are given a sequence of n songs where the i^{th} song is ℓ_i minutes long. You want to place all of the songs on an ordered series of CDs (e.g. $CD_1, CD_2, CD_3, \ldots CD_k$) where each CD can hold m minutes. Furthermore,

   (a) The songs must be recorded in the given order, song 1, song 2, ..., song n.

   (b) All songs must be included.

   (c) No song may be split across CDs.

   Your goal is to determine how to place them on the CDs as to minimize the number of CDs needed. Give the most efficient algorithm you can to find an optimal solution for this problem, prove the algorithm is correct and analyze its time complexity.

2. At a trade school, there are N workers looking for jobs, each with a skill level x_i. There are P entry-level job openings, and the i^{th} opening only accepts workers with a skill level less than or equal to p_i. There are also Q senior job openings, the i of which requires a skill level of at least q_i. Each worker can take at most one job, and each job opening only accepts a single worker.

   Your task is to determine the largest number of workers you can assign to jobs in time $O(N \log N + P \log P + Q \log Q)$.

3. A city is attacked by N monsters and is defended by a single hero with initial strength of S units. To kill a monster i the hero must dissipate

a_i units of strength; if monster i is killed successfully the hero gains g_i units of strength. Thus, if the hero had strength $c \geq a_i$ before tackling monster i he can kill the monster and he will end up with $c - a_i + g_i$ units of strength. Note that for some monsters i you might have $g_i \geq a_i$ but for some other j you might have $a_j > g_j$. You are given S and for each i you are given a_i and g_i. Design an algorithm which determines in which order the hero can fight the monsters if he is to kill them all (if there is such an ordering). In case there is no such ordering the algorithm should output "no such ordering". Assume all values are positive integers.

4. You are given n stacks of blocks. The i^{th} stack contains $h_i > 0$ identical blocks. You are also able to move for any $i \leq n - 1$ any number of blocks from stack i to stack $i + 1$. Design an algorithm to find out in $O(n)$ time whether it is possible to make the sizes of stacks strictly increasing. (For example, 1,2,3,4 are strictly increasing but 1,2,2,3 are not). The input for your algorithm is an array A of length n such that $A[i] = h_i$. Note that you are not asked to actually move the blocks, only to determine if such movements exists or not.

5. You are given n jobs where each job takes one unit of time to finish. Job i will provide a profit of g_i dollars ($g_i > 0$) if finished at or before time t_i, where t_i is an arbitrary integer larger or equal to 1. Only one job can be done at any instant of time and jobs have to be done continuously from start to finish. (Note: If a job i is not finished by t_i then there is no benefit in scheduling it at all. All jobs can start as early as time 0.) Give the most efficient algorithm you can to find a schedule that maximizes the total profit.