Name: ______________________________ **Entry number**: ____________________

COL351 Quiz 2 Duration: 1 hour

Read the following instructions before you begin writing.

1. Keep a pen, your identity card, and optionally a water bottle with you. Keep everything else away from you, at the place specified by the invigilators.
2. Do not detach sheets. Write your entry number and name on every page. (We will detach sheets prior to grading.)
3. Answer only in the designated space. Think before you use this space. No additional space will be provided for writing answers. Blank pages will be provided for rough work on demand, but they cannot be used for writing answers.
4. No clarifications will be given during the exams. If something is unclear or ambiguous, make reasonable assumptions and state them clearly. The instructor reserves the right to decide whether your assumptions were indeed reasonable.
5. You may use any result discussed in class or tutorials without proving it. Similarly, you can use any algorithm discussed in class or tutorials as a "black-box" i.e., without reproducing any details of how it works.

---

1. (10 points) Recall the problem of scheduling jobs to minimize total waiting time discussed in class. The input is the processing times of n jobs – $p[1], \ldots, p[n]$, and the goal is to order them in such a way that their total waiting time is minimized. The waiting time of job i is defined to be the sum of the processing times of all jobs placed before job i.

   Consider the following modified problem in which not all jobs are equally important. For each job i, in addition to its processing time $p[i] > 0$, you are also given its *importance* expressed by a number $w[i] > 0$. Your goal is to order the jobs in such a way that the sum over all i of $w[i]$ times the waiting time of i is minimized. Design a polynomial time algorithm which, given the arrays p and w, finds an optimal ordering of the jobs.

Class Notes

Name: ____________________________ **Entry number:** ____________________

COL351 Quiz 2 Duration: 1 hour

2. (10 points) Consider a workshop which manufactures cars from their n parts. The task of making a car happens on an assembly line consisting of n machines, where the i'th machine attaches the i'th part to a partially constructed car. The workshop actually has two assembly lines, L^0 and L^1, where L^j consists of machines $M_1^j, \dots, M_n^j$. Machines M_i^0 and M_i^1 both perform the task of attaching the i'th part to a partially assembled car with parts $1, \dots, i-1$ attached already. Let t_i^j denote the processing time of machine M_i^j.

   A car is assembled correctly if and only if it is processed by one of M_1^0, M_1^1, one of M_2^0, M_2^1, one of M_3^0, M_3^1, ..., one of M_n^0, M_n^1, in this order. Since machines M_i^j and M_{i+1}^j are on the same assembly line, the transfer of a partially assembled car from the former to the latter is instantaneous. On the other hand, since M_i^j and M_{i+1}^{1-j} are on different assembly lines, it takes time T to transfer a partially assembled car from the former to the latter. Thus, the total time taken to assemble a car is equal to the sum of the processing times of the machines involved in assembling it, plus T times the total number of transfers between the assembly lines. For example, if $n = 4$, the production of a car on machines $M_1^0, M_2^0, M_3^0, M_4^0$ takes time $t_1^0 + t_2^0 + t_3^0 + t_4^0$, while the production on machines $M_1^0, M_2^1, M_3^0, M_4^1$ takes time $t_1^0 + t_2^1 + t_3^0 + t_4^1 + 3T$.

   Design a polynomial time algorithm which, given T and the $2n$ numbers t_i^j for $j \in \{0, 1\}$ and $i \in \{1, \dots, n\}$, outputs the minimum time in which a car can be assembled in the workshop.

4n nodes M_{ik}^j , $k = \{0,1\}$

M_{i0}^j → denotes machine M_i^j before ~~attaching~~ attaching parts

M_{i1}^j → " " " after " "

$c(M_{i0}^j, M_{i1}^j) = t_i^j$

$c(M_i^0, M_{i+1}^1) = c(M_i^1, M_{i+1}^0) = T$

→ $|V| = O(4n)$, $|E| = O(n)$

→ Dijkstra → $O(n + n\log n) = O(n\log n)$