Week 2 Tutorial
Question 1

Proposed algorithm for interval scheduling:
1. Sort the intervals in **decreasing** order of **start** time.
2. Consider the intervals in that order. Keep the interval under consideration if it does not overlap any of the previously kept intervals; otherwise discard the interval.
3. Return the intervals kept.

Does this algorithm find an optimal set of intervals?
Question 2

Minimum lateness scheduling problem. True or false?

1. In some optimal schedule the jobs appear in increasing deadline

2. If a schedule is optimal, the jobs in it appear increasing deadline

3. (Do at home) The algorithm discussed in class minimizes the SUM of all latenesses (not only the MAXIMUM of all latenesses)
Vassos has \( n \) assignments due now that he has not started working on yet. Assignment \( i \) takes him \( d_i \) days to do, and has penalty \( p_i \) for each day it is submitted late.

The total penalty (sum of penalties) if Vassos does the assignments in the order 1, 2, ..., \( n \) is:

\[
p_1 d_1 + p_2 (d_1 + d_2) + p_3 (d_1 + d_2 + d_3) + \cdots + p_n (d_1 + d_2 + \cdots + d_n)
\]

(He will finish assignment \( i \) after \( d_1 + d_2 + \cdots + d_i \) days.)
Question 3 (cont’d)

In what order should Vassos do his assignments to minimize the total penalty?

**Greedy strategy A (shortest first):** Do the assignments in *increasing length* (number of days they take)

**Greedy strategy B (harshest penalty first):** Do the assignments in *decreasing penalty*

**Greedy strategy C:** Do the assignments in *decreasing penalty-to-length ratio* (equivalently: *increasing length-to-penalty ratio*)