Greed

1. A single machine must process n jobs, one at a time, starting at time 0. Every job is available immediately, and once a job begins, it must run to completion without interruption.

   Job i has processing time p_i, deadline d_i, and completion time C_i. Its *tardiness* is $T_i = \max\{0, C_i - d_i\}$. Your goal is to find a schedule that minimizes the maximum tardiness,

   $$T_{\max} = \max_{1 \le i \le n} T_i.$$

   Consider the following jobs:

Job	Processing time p_i	Deadline d_i
A	3	4
B	2	8
C	4	6
D	1	9

   (a) The the schedule for each of the following greedy rules:
   - Shortest processing time first.
   - Earliest deadline first.

   Which rule performs better on this instance?

   (b) Suppose a schedule contains two consecutive jobs i, j with $d_i > d_j$: the job with the later deadline appears first. Let t be the time when job i starts.
   Write expressions for the completion times of both jobs before swapping them, when the order is i, j, as well as after swapping them, when the order is j, i.
   Assume that the pair begins at time t and has no idle time in either order.

   (c) Show that after the swap, neither job has tardiness greater than the maximum tardiness of the pair before the swap. *Hint: Before the swap, job j finishes at $t + p_i + p_j$. Compare each job's new tardiness with job j's old tardiness.*

   (d) Explain why the swap does not affect the completion time of any other job. What does this prove about the earliest-deadline-first rule?