Exercise 1 Consider the following instance of the scheduling problem $1||\sum_j w_j C_j$. Give an optimal schedule and its value.

jobs	1	2	3	4
w_j	6	11	9	5
p_j	3	5	7	4

Exercise 2 Consider the following instance of the scheduling problem $1||L_{\max}$. Give an optimal schedule and its value.

jobs	1	2	3	4
p_j	5	4	3	6
d_j	3	5	11	12

Exercise 3 De decision problems PARTITION and 3-PARTITION are both NP-complete and are defined as follows:

PARTITION: An instance is given by positive numbers A and $a_1, a_2, \ldots, a_n$ with $\sum_i a_i = 2A$. Question: Is there an $S \subset \{1, 2, \ldots, n\}$ such that $\sum_{i \in S} a_i = A$.?

3-PARTITION: An instance is given by positive numbers B and $b_1, b_2, \ldots, b_{3m}$ with $\sum_i b_i = mB$. Question: Is there a partition of $\{1, 2, \ldots, 3m\}$ into $S_1, S_2, \ldots, S_m$ such that $\sum_{j \in S_i} b_j = B$ for all $i = 1, \ldots, m$?

(a) Show that the PARTITION problem can be reduced to the scheduling problem $P2||C_{\max}$.
(b) Show that the 3-PARTITION problem can be reduced to the scheduling problem $P||C_{\max}$.

Exercise 4 Consider the scheduling problem $1|r_j|\sum_j C_j$ and the following algorithm (SPT):
When the machine is not processing any job, then start the job that has the smallest processing time p_j among the available jobs. (We say that a job is available if it has been released but not started yet).
Show by an example that this algorithm does not always lead to an optimal schedule.

Exercise 5 Consider the scheduling problem $P|r_j, pmtn|C_{\max}$. Give a polynomial time algorithm which solves the problem by formulating it as a linear program (LP). Assume for simplicity that $0 = r_1 \leq r_2 \leq \cdots \leq r_n$ where n is the number of jobs.
Hint: Use a variable Z for the length of the schedule. The objective then becomes: minimize Z. Take as variables x_{tj} $(t = 1, 2 \ldots, n)$ which denote the amount of time spent on job j between time r_t and r_{t+1} $(t \leq n-1)$ and between r_n and Z $(t = n)$. Explain how an optimal LP-solution can be translated into a feasible schedule.

Exercises from the slides.

Exercise 1 (Slides) Show (by an example) that SRPT is not optimal on parallel machines.

SPRT on m parallel machine:
At any moment in time, process the m jobs with smallest remaining processing time (or all jobs if there are less than m jobs available at that time.

Exercise 2 (Slides) This exercise refers to problem $R||\sum C_j$ on the slides. Form this exercise, it follows that this scheduling problem can be solved efficiently. Let $G = (V_1 \cup V_2, E)$ be a complete bipartite graph with $|V_1| \leq |V_2|$. For any pair $u \in V_1$ and $v \in V_2$ let c_{uv} be the cost of edge (u, v). Say that a matching M is perfect if all vertices in V_1 are matched. Since the graph is complete and $|V_1| \leq |V_2|$, a perfect matching exists. In the MINCOST PERFECT MATCHING problem we need to find a perfect matching for which the total cost of the edges in the matching is minimized.
Show how the Mincost perfect matching problem can be reduced to a mincost flow problem.
Hint: Remember from week 1 how the maximum matching problem can be reduced to the maximum flow problem.

Exercise 3 (Slides) (Difficult) We have seen a 2-approximation for the problem $1|prec|\sum C_j$. Consider the following generalizations:

- $1|prec|\sum w_j C_j$
- $1|r_j, prec|\sum C_j$

(a) Does the same algorithm and proof apply for the weighted version $1|prec|\sum w_j C_j$?

(b) Try to apply the same technique to the problem $1|r_j, prec|\sum C_j$. What is the approximation ratio that you get?