1. **Scheduling algorithms** The following $n = 12$ jobs with given processing times have to be scheduled on $m = 3$ parallel and identical processors with the objective of minimizing the makespan. $C_j$ is the completion time of a job.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>9</td>
<td>12</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>11</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

(a) Draw the List-Scheduling-schedule as a Gantt-Chart. How much is $C_{\text{max}}$ and how much is $\sum C_j$?
(b) Draw the LPT-schedule as a Gantt-Chart. How much is $C_{\text{max}}$ and how much is $\sum C_j$?
(c) Draw the SPT-schedule as a Gantt-Chart. How much is $C_{\text{max}}$ and how much is $\sum C_j$?
(d) Find the optimal and the worst list concerning the objective of minimizing the makespan.
(e) Find the optimal and the worst list concerning the objective of minimizing the sum of the processing times.
(f) Draw McNaughton’s schedule (preemptions allowed).

2. **Upper and lower bounds for scheduling algorithms, NP-completeness**

(a) Explain why $\sum_{j=1}^n p_j/m$ and $p_k$ (for any $k \in \{1, \ldots, n\}$) are Lower Bounds for the makespan of any algorithm solving $P||C_{\text{max}}$.

(b) Find Upper Bounds for the makespan of List Scheduling for $P||C_{\text{max}}$.

(c) Find a tight worst-case example for the makespan achieved by List Scheduling in comparison to the optimum makespan for $m = 5$ processors.

(d) Repeat in your own words the most important ideas behind Ron Graham’s proof for the worst-case behaviour of List Scheduling.

(e) Prove that the decision version of $P || C_{\text{max}}$ is NP-complete by reducing Partition.

3. **Time analysis of scheduling algorithms**

(a) Write the Pseudocode of the List Scheduling algorithm to solve $P || C_{\text{max}}$. Proof the correctness and perform a time analysis. Does LPT has the same time complexity?

(b) Write the Pseudocode of McNaughton’s algorithm to solve $P | pmtn | C_{\text{max}}$. Proof the correctness and perform a time analysis.

(c) Prove that McNaughton’s algorithm needs never more than $n - 1$ preemptions to produce an optimal solution.