Aufgabe 1: EDD

Check whether the Earliest Deadline Due (EDD) algorithm produces a feasible schedule for the following task set, given that all tasks are synchronous and arrive at time $t = 0$.

<table>
<thead>
<tr>
<th></th>
<th>$J_1$</th>
<th>$J_2$</th>
<th>$J_3$</th>
<th>$J_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>$D_i$</td>
<td>8</td>
<td>15</td>
<td>3</td>
<td>11</td>
</tr>
</tbody>
</table>

Aufgabe 2: LDF

Given the precedence graph in Figure 1 and the following table of task execution times and deadlines, determine the Latest Deadline First (LDF) schedule. Is the schedule feasible?

<table>
<thead>
<tr>
<th></th>
<th>$J_1$</th>
<th>$J_2$</th>
<th>$J_3$</th>
<th>$J_4$</th>
<th>$J_5$</th>
<th>$J_6$</th>
<th>$J_7$</th>
<th>$J_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>$D_i$</td>
<td>5</td>
<td>8</td>
<td>11</td>
<td>15</td>
<td>12</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>
Aufgabe 3: EDF

Given are five tasks with arrival times, execution times and deadlines according to the following table. (1) Determine the Earliest Deadline First (EDF) schedule. Is the schedule feasible?

<table>
<thead>
<tr>
<th></th>
<th>J1</th>
<th>J2</th>
<th>J3</th>
<th>J4</th>
<th>J5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i$</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>$C_i$</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$d_i$</td>
<td>16</td>
<td>7</td>
<td>8</td>
<td>11</td>
<td>18</td>
</tr>
</tbody>
</table>

(2) At time $t = 3$, a new task $J_x$ arrives with execution time $C_x = 2$ and deadline $d_x = 10$. Can you guarantee the schedulability of the task set with this new task or do you have to reject it?

Aufgabe 4: EDF

Given are seven tasks $A, B, C, D, E, F, G$ with following precedence constraints:

$$
A \rightarrow C, \quad B \rightarrow C, \quad C \rightarrow E, \quad D \rightarrow F, \quad B \rightarrow D, \quad C \rightarrow F, \quad D \rightarrow G
$$

All tasks arrive at time $t_0 = 0$, have a common deadline $d = 20$ and the following execution times:

<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>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

(1) Construct the precedence graph for this task set. Then, modify the release times and deadlines so that EDF* can be used for its scheduling.

(2) Draw the resulting EDF* schedule and compute the average response time of the tasks.

(3) Assume the additional precedence constraint $E \rightarrow A$. Is there still a feasible schedule for the above task set? Justify your answer.
Aufgabe 5: EDF with Precedence Constraints

Given are eight aperiodic tasks \( J_1, \ldots, J_8 \) with their arrival times, deadlines, and execution times as shown in the table below. Task precedence constraints are as follows: \( J_1 \rightarrow J_2, J_2 \rightarrow J_3, J_3 \rightarrow J_4, J_5 \rightarrow J_6, J_6 \rightarrow J_7, J_6 \rightarrow J_8, J_2 \rightarrow J_7, J_7 \rightarrow J_4 \) and \( J_8 \rightarrow J_7 \).

\[
\begin{array}{cccccccc}
& J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 & J_8 \\
r_i & 0 & 3 & 4 & 0 & 0 & 2 & 0 & 2 \\
d_i & 3 & 8 & 15 & 15 & 10 & 10 & 10 & 11 \\
C_i & 1 & 3 & 3 & 3 & 1 & 1 & 2 & 1 \\
\end{array}
\]

(a) Construct the precedence graph.

(b) Using EDF\(^*\) algorithm, modify the arrival times and deadlines of the tasks in order to make the tasks schedulable under EDF. Enter the modified arrival times and deadlines in the table below.

\[
\begin{array}{cccccccc}
& J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 & J_8 \\
r_i^* & & & & & & & & \\
d_i^* & & & & & & & & \\
C_i & 1 & 3 & 3 & 3 & 1 & 1 & 2 & 1 \\
\end{array}
\]

(c) Assume that the application is executed on a dual-core platform. At any time \( t \), both cores execute the two ready tasks (\( r_i^* \leq t \)) with earliest deadlines (Note: A single task cannot be executed on two cores simultaneously). Using the arrival times and deadlines obtained in (b), construct the EDF schedule in Figure 6.

![Figure 2: EDF schedule for part (c)](image)

(d) Now assume that the application is executed on a quad-core platform with the same scheduling rule (4 cores execute the four ready tasks with earliest deadlines). Will executing on the quad-core platform reduce the completion time of the application? Justify your answer with an explanation.