## Critical Path Method

**Forward procedure** that yields a schedule with minimum makespan.

**Notation:**
- $p_j$: processing time of job $j$
- $S_j$: the earliest possible starting time of job $j$
- $C_j$: the earliest possible completion time of job $j$
- $\{ k \rightarrow j \}$: jobs that are predecessors of job $j$

**Step 1.**
For each job $j$ that has no predecessors $S_j=0$ and $C_j=p_j$

**Step 2.**
Compute inductively for each remaining job $j$
- $S_j = \max_{k \rightarrow j} S_k + p_j$
- $C_j = S_j + p_j$

**Step 3.**
$C_{\text{max}} = \max(C_1, \ldots, C_n)$

**Backward procedure** determines the latest possible starting and completion times.

**Notation:**
- $C'_j$: the latest possible completion time of job $j$
- $S'_j$: the latest possible starting time of job $j$
- $\{ j \rightarrow k \}$: jobs that are successors of job $j$

**Step 1.**
For each job $j$ that has no successors $C'_j = C_{\text{max}}$ and $S'_j = C_{\text{max}} - p_j$

**Step 2.**
Compute inductively for each remaining job $j$
- $C'_j = \min_{j \rightarrow k} S'_k$
- $S'_j = C'_j - p_j$

**Step 3.**
Verify that $0 = \min(S'_1, \ldots, S'_n)$

- The jobs whose earliest possible starting times are earlier than latest possible starting times are referred to as **slack jobs**.
- The jobs whose earliest possible starting times are equal to their latest possible starting times are **critical jobs**.
- A critical path is a chain of jobs which begin at time 0 and ends at $C_{\text{max}}$. 

### Example

<table>
<thead>
<tr>
<th>jobs</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_j$</td>
<td>4</td>
<td>9</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>12</td>
<td>6</td>
</tr>
</tbody>
</table>

![Graph example]
**Extensions to the classical project scheduling problems**

- Stochastic activity (job) durations
- Nonavailability of resources
- Multiple resource types
- Preemption of activities
- Multiple projects with individual project due-dates

**Objectives:**
- common one: minimising overall project duration
- resource leveling
  - minimise resource loading peaks without increasing project duration
- maximise resource utilisation factors