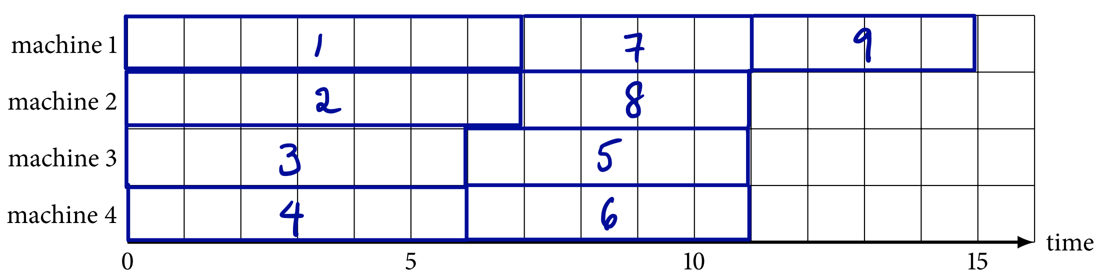
Lesson 9. Machine Scheduling

Problem. The Markov Micromanufacturing Company has 9 production jobs it needs to process in the next 24 hours. The company has 4 identical machines that run in parallel. Each of these 9 jobs must be run on one of these machines **nonpreemptively**: that is, once a job is started on a machine, it must stay on that machine until it is completed. The processing times of these jobs are given below:

job	1	2	3	4	5	6	7	8	9
processing time (hours)	7	7	6	6	5	5	4	4	4

The company wants to minimize the **makespan**, or the completion time of the last job to finish processing.

- Let m be the number of machines — in this case, $m = 4$
- Suppose we schedule the jobs using the **longest processing time first (LPT)** rule:
  - First, schedule the m longest jobs on the m machines
  - Whenever a machine becomes free, put the longest unprocessed job on that machine
- Idea: LPT puts shorter jobs towards the end of the schedule, where they can be used to balance the loads on each machine
- For our problem, this yields a schedule that looks like this:



  - This kind of diagram is known as a **Gantt chart**
- Therefore, the makespan for the LPT schedule is 15
- It turns out that the makespan of an LPT schedule is always at most $33.\bar{3}\%$ larger than the minimum makespan
- So... can we do better?
- Let's formulate this problem as a dynamic program

- Stages:

Stage t represents assigning job t to a machine $(t = 1, \ldots, 9)$ or the end of the decision-making process $(t = 10)$

- States in stage t (nodes):

stage t → $t_{n_1,\ n_2,\ n_3,\ n_4}$

load on machine 1

load on machine 2

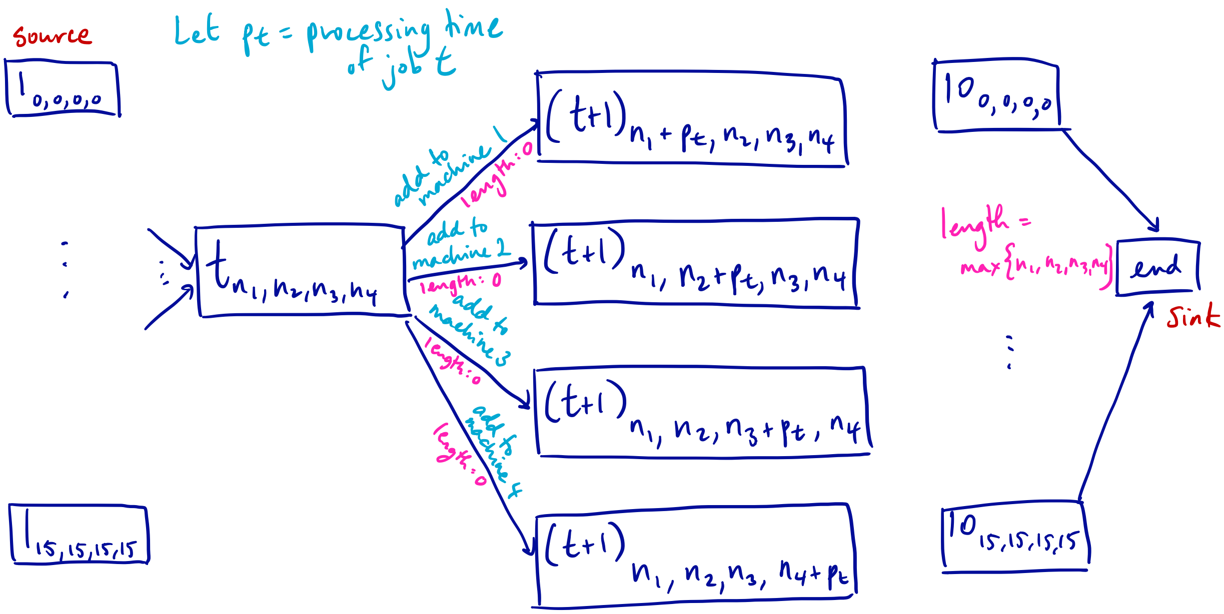
load on machine 3

load on machine 4

$n_1, n_2, n_3, n_4 = 0, 1, \ldots, 15$

We know that the load on each machine ≤ 15 in an optimal solution by LPT.

- Decisions, transitions, and rewards/costs at stage t (edges):



- Shortest/longest path? Shortest

- Minimum makespan ⟷ Length of shortest path

- Assignments of jobs to machines ⟷

Examine edges in shortest path.

e.g. $((t_{n_1,n_2,n_3,n_4}),\ ((t+1)_{m_1,m_2,m_3,m_4}))$ If $n_i \neq m_i$, then assign job t to machine i