

Fifteen jobs, each with its processing time, should be scheduled on three machines. If two jobs cannot be scheduled on the same machine, they are called conflicting jobs. Table 1 lists the job IDs, processing times, and sets of conflicting jobs. For example, we cannot schedule any pair of jobs out of jobs 2, 5, 8 on the same machine. Note that a job may have no conflicting jobs.

Job	Processing time	Conflicting jobs
1	7	None
2	4	5, 8
3	6	None
4	9	None
5	12	2, 8
6	8	9
7	10	10
8	11	2, 5
9	8	6
10	7	7
11	6	15
12	8	None
13	15	None
14	14	None
15	3	11

We want to schedule the jobs to minimize makespan. For example, we may schedule jobs 1, 4, 7, 8, and 13 to machine 1, jobs 2, 6, 10, 11, and 14 to machine 2, and jobs 3, 5, 9, 12, and 15 to machine 3. The total processing times on the three machines are 52, 39, and 37, respectively. The makespan is thus 52. While this is a feasible schedule, this may or may not be an optimal schedule. When we try to improve the schedule, be careful about conflicting jobs. For example, we cannot exchange jobs 8 and 11 (even though this reduces the makespan) because that will result in machine 2 processing conflicting jobs 2 and 8, which is infeasible.

Formulate a linear integer program that generates a feasible schedule to minimize makespan. Then write a computer program (e.g., using Python to invoke Gurobi Optimizer) to solve this instance and obtain an optimal schedule. Write down the minimized makespan (i.e. the objective value of an optimal solution). Do not have any symbol other than numeric values in your answer.