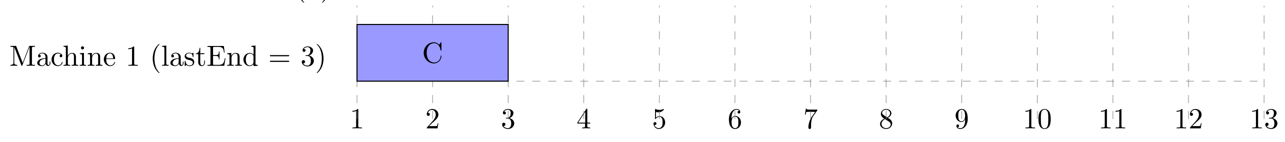
Solve the task scheduling problem for the following tasks (labeled alphabetically) with the specified start and end times: A [12, 13], B [8, 11], C [1, 3], D [1, 2], E [3, 4], F [6, 10], G [7, 9], H [10, 12], I [4, 6], J [3, 7], K [2, 5], L [9, 13], M [3, 6], N [5, 8]

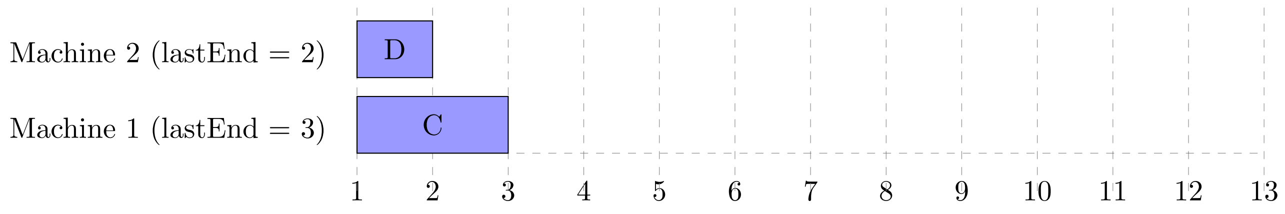
Solution: First, sort tasks by increasing start time. For tasks that have the same start time, we can choose arbitrarily among them. So, we process tasks in this order:

1. Task C: [1, 3]
2. Task D: [1, 2]
3. Task K: [2, 5]
4. Task M: [3, 6]
5. Task J: [3, 7]
6. Task E: [3, 4]
7. Task I: [4, 6]
8. Task N: [5, 8]
9. Task F: [6, 10]
10. Task G: [7, 9]
11. Task B: [8, 11]
12. Task L: [9, 13]
13. Task H: [10, 12]
14. Task A: [12, 13]

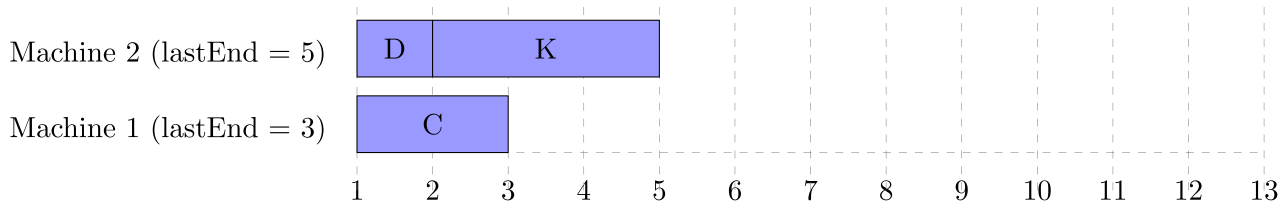
Schedule the first task (Task C: [1, 3]) on a machine. We have no machines, so create one. Store with this machine the last end time (3).



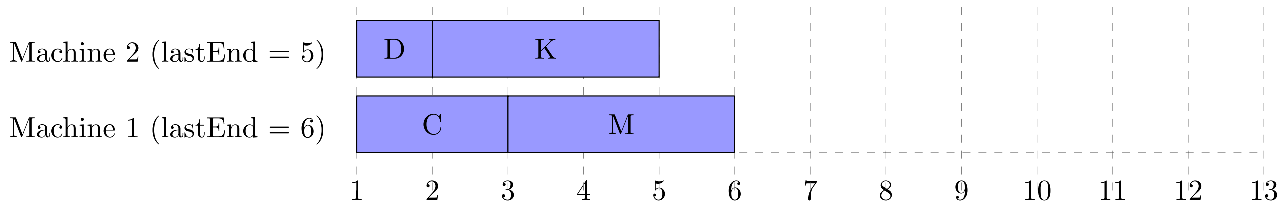
Process the next task (Task D: [1, 2]). Check if it conflicts on the machine with the earliest end time (machine 1). It does, so we create a new machine, schedule a task on it, and store with the machine the last end time (2).



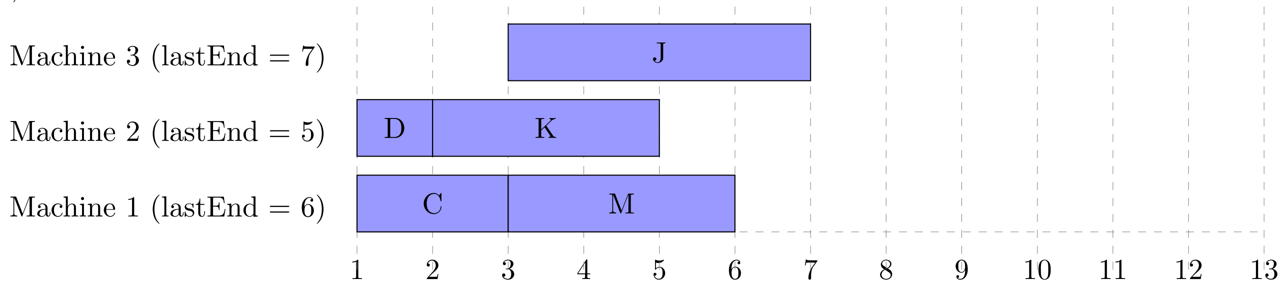
Process the next task (Task K: [2, 5]). Check if it conflicts on the machine with the earliest end time - that's machine 2 now. It doesn't, so we can schedule it on machine 2. Update machine 2's last end time to be the end time of task K.



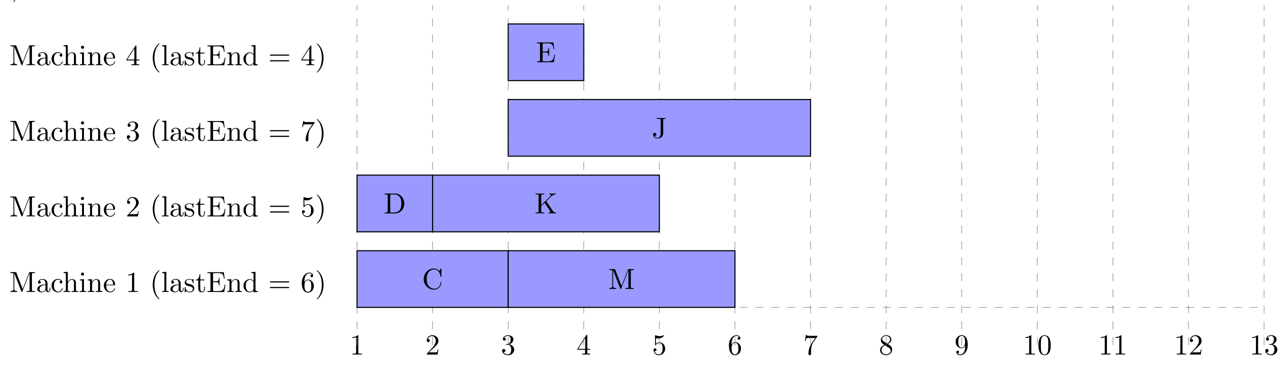
Process Task M: [3, 6]. Check if it conflicts with the machine with the earliest end time - that's machine 1 now. It doesn't, so we can schedule it on machine 1. Update machine 1's last end time to be the end time of task M.



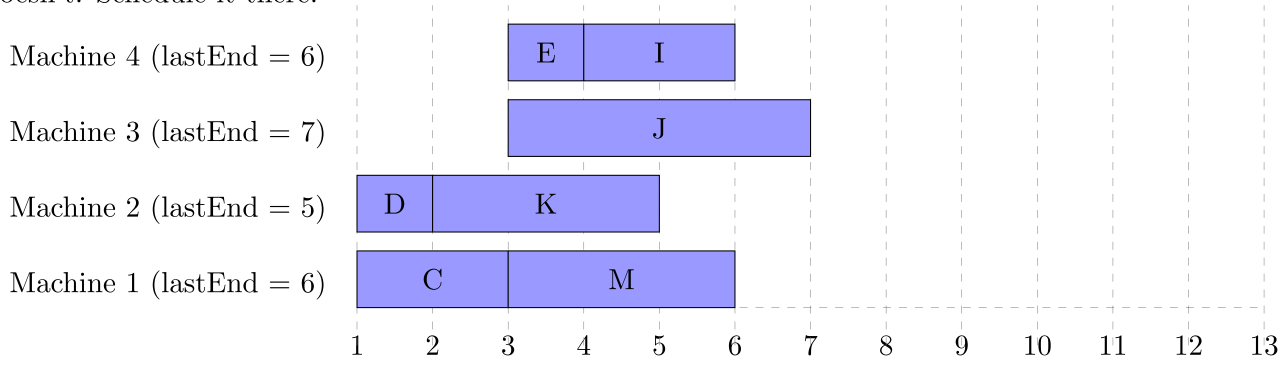
Process Task J: [3, 7]. Check if it conflicts with machine 2 (since that machine has the earliest end time). It does, so create a new machine.



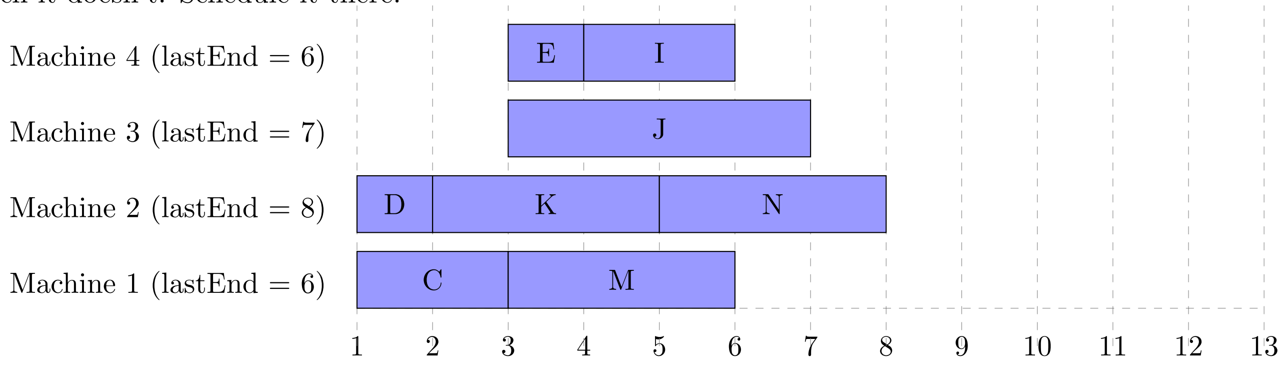
Process Task E: [3, 4]. Check if it conflicts with machine 2 (since that machine has the earliest end time). It does, so create a new machine.



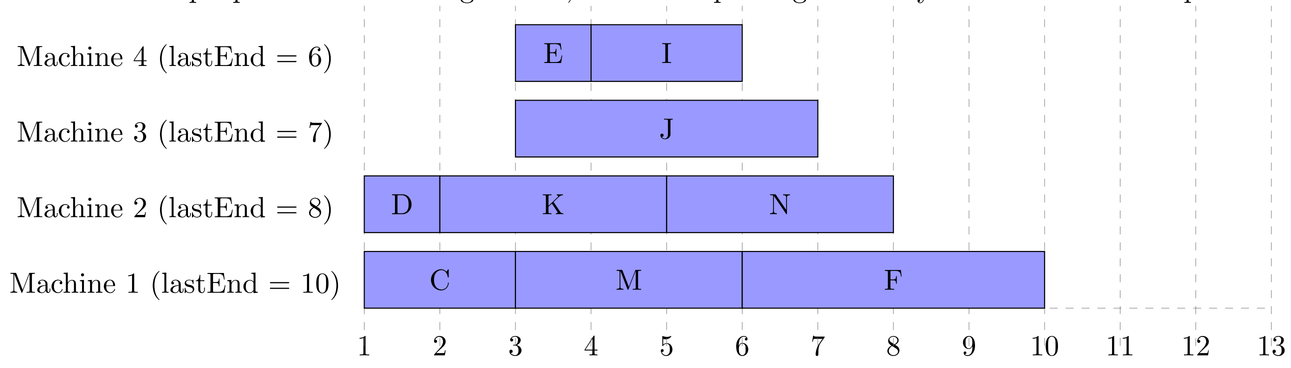
Process Task I: [4, 6]. Check if it conflicts on machine 4 (since that machine has the earliest end time), which it doesn't. Schedule it there.



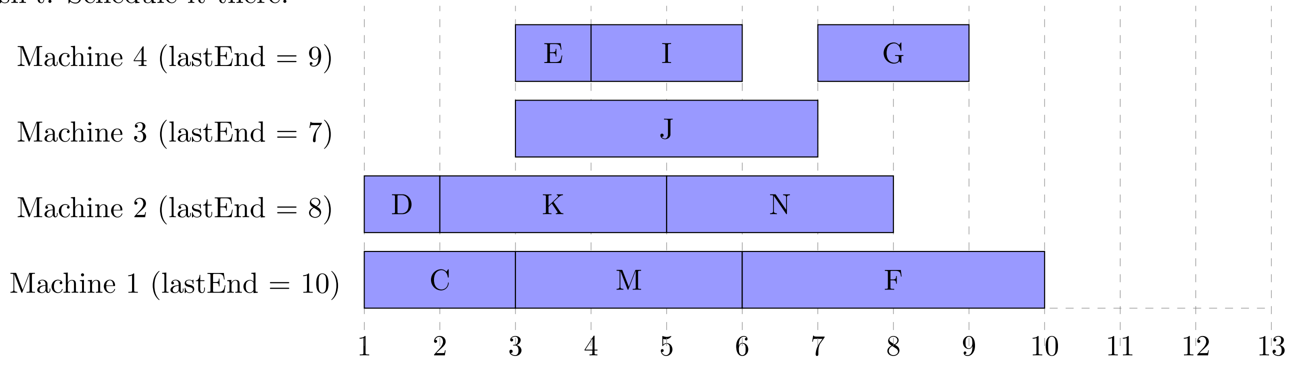
Process Task N: [5, 8]. Check if it conflicts with machine 2 (since that machine has the earliest end time), which it doesn't. Schedule it there.



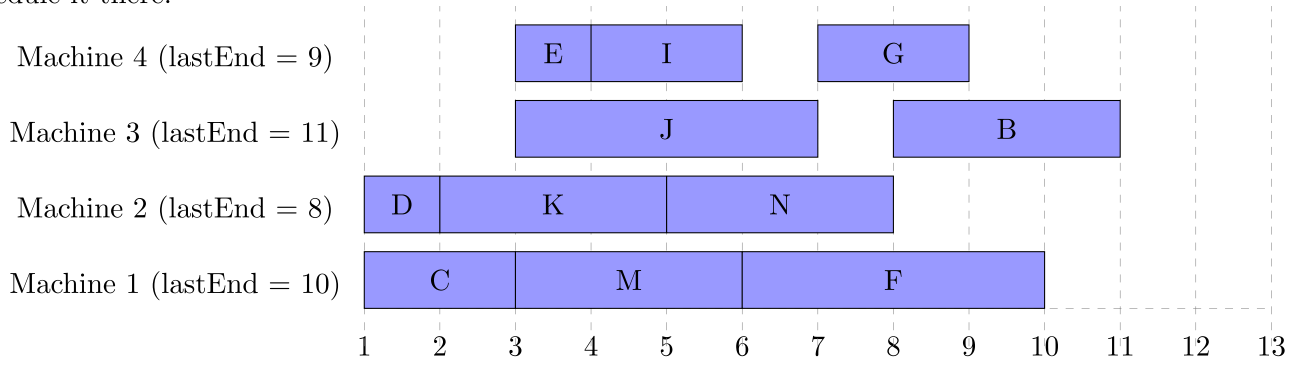
Process Task F: [6, 10]. The machine with the earliest end time is machine 1 and machine 4. They are equivalent for our purposes of scheduling task F, so we compare against only one of them - let's pick machine 1.



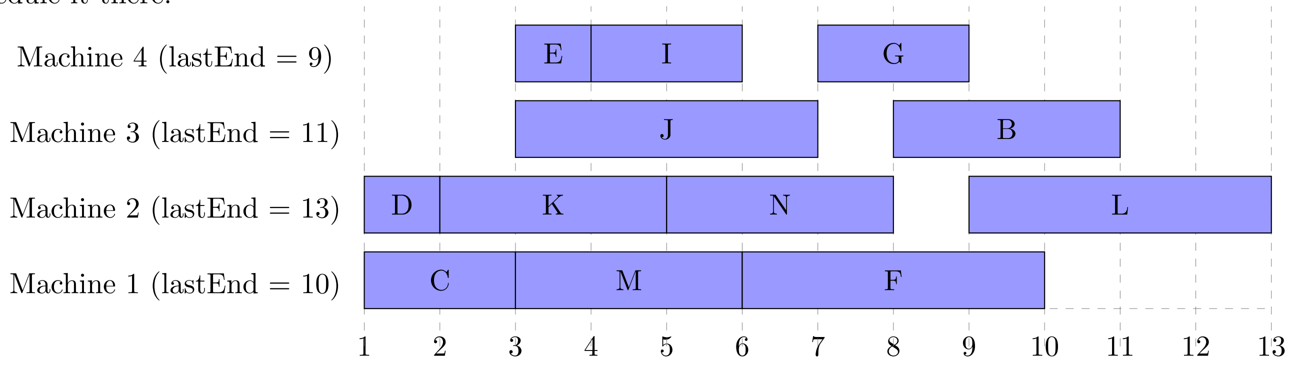
Process Task G: [7, 9]. Check if it conflicts with machine 4 (the machine with the earliest end time), and it doesn't. Schedule it there.



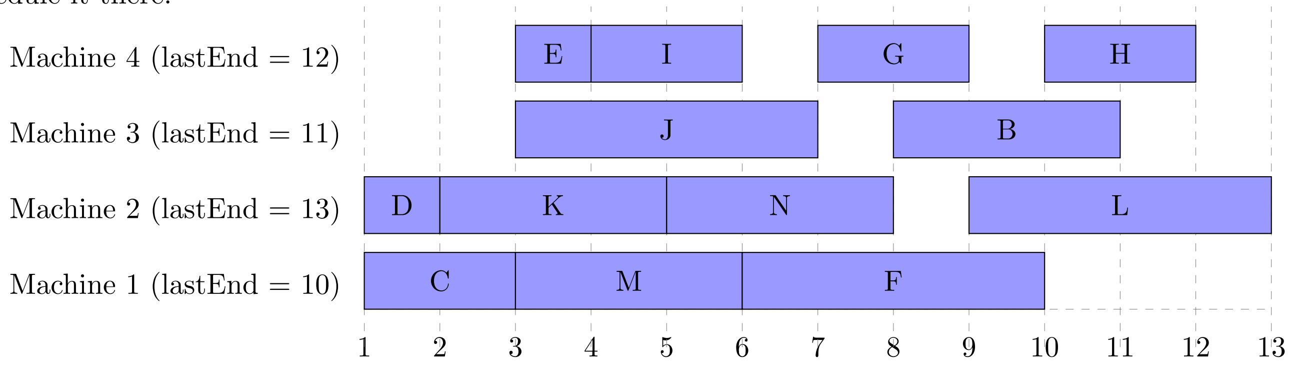
Process Task B: [8, 11]. Check if it conflicts with machine 3 (machine with earliest end time). It doesn't, so schedule it there.



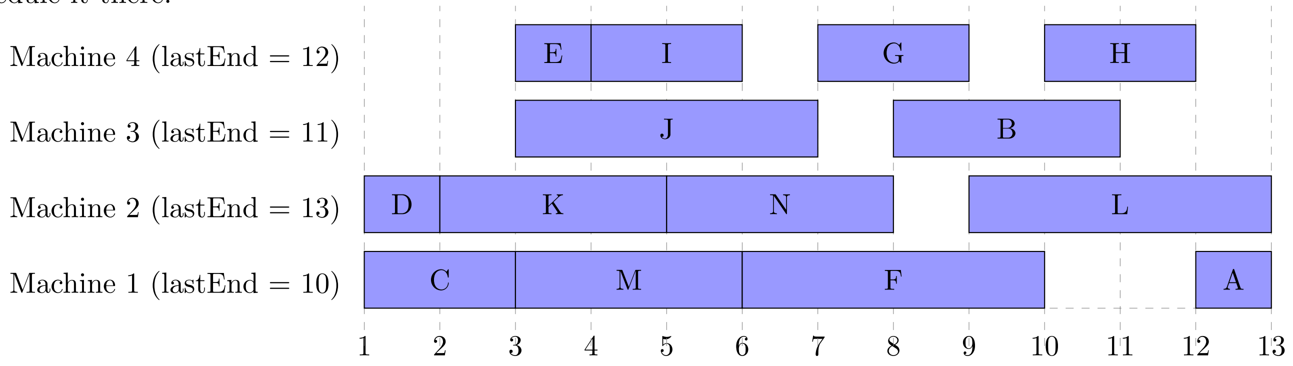
Process Task L: [9, 13]. Check if it conflicts with machine 2 (machine with earliest end time). It doesn't, so schedule it there.



Process Task H: [10, 12]. Check if it conflicts with machine 4 (machine with earliest end time). It doesn't, so schedule it there.



Process Task A: [12, 13]. Check if it conflicts with machine 1 (machine with earliest end time). It doesn't, so schedule it there.


Machine 4 (lastEnd = 12)
Machine 3 (lastEnd = 11)
Machine 2 (lastEnd = 13)
Machine 1 (lastEnd = 10)
E
I
G
H
J
B
D
K
N
L
C
M
F
A