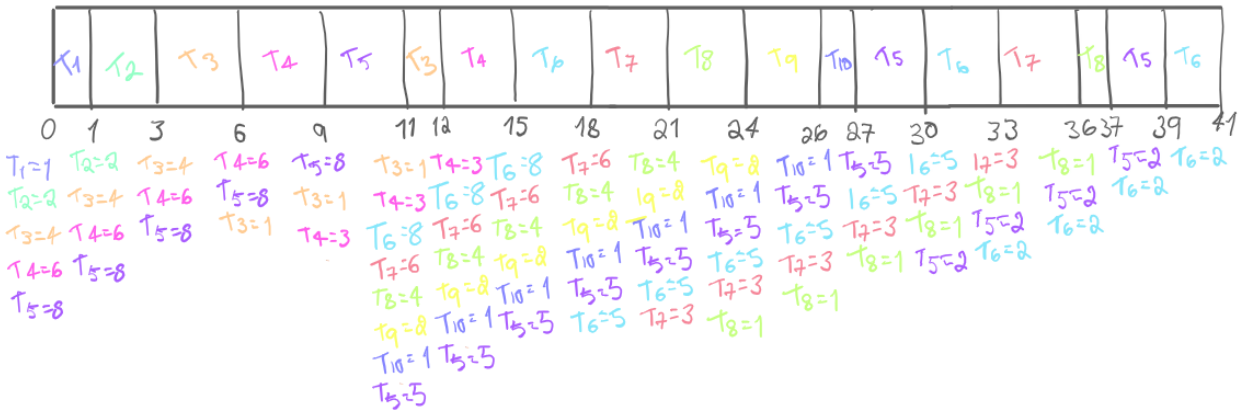
Task 3

Process	BurstTime	Arrival Time	Process	BurstTime	Arrival Time
$T1$	1	0	$T6$	8	11
$T2$	2	0	$T7$	6	11
$T3$	4	0	$T8$	4	11
$T4$	6	0	$T9$	2	11
$T5$	8	0	$T10$	1	11

Time quantum for RR scheduler is $3\ units$

Testing scheduling of $43\ quanta\ of\ time$

Round Robin scheduling



Shortest waiting time: $T1 = 0q$

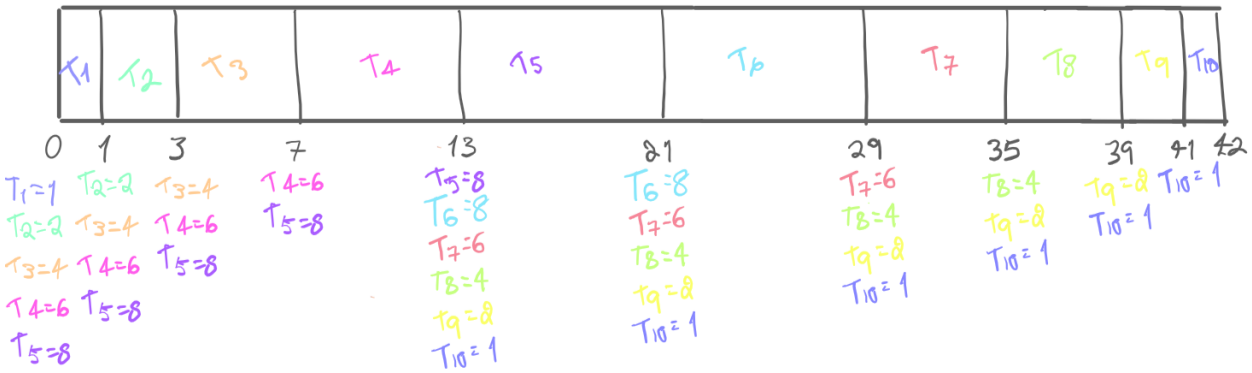
Longest waiting time: $T8 = 21q$

Total execution time: $41q$

Average waiting time: $\frac{T1+T2+T3+T4+T5+T6+T7+T8+T9+T10}{number\ of\ processes} = \frac{0+1+9+10+32+22+19+22+13+15}{10} = 14{,}3q$

With RR the average waiting time is higher than FCFS, but the highest waiting time for a process is lower. The total execution time is shorter. This means that with a lot of processes, the short processes won't be buried under long processes hogging the CPU quanta and they are rather delegated a timeslot where they can be completed within a reasonable time. This halts bigger processes which will be resumed after their preemption, which in turn increases the average waiting time.

First Come First Serve scheduling



Shortest waiting time: $T1 = 0q$

Longest waiting time: $T10 = 30q$

Total execution time: $42q$

Average waiting time: $\frac{T1+T2+T3+T4+T5+T6+T7+T8+T9+T10}{number\ of\ processes} = \frac{0+1+3+7+13+10+18+24+28+30}{10} = 13{,}4q$

With FCFS the average waiting time is lower than with RR, but the highest waiting time for a process is higher. The total execution time is longer. This means that with a lot of processes, some short processes may wait for a long time, if not “forever” to just run a tiny process, because the longer processes will have to finish before they can execute.