1. Preemptive scheduling is a method where the operating system can interrupt a process that is currently running, stop its execution and allocate the CPU to another process with higher priority. This approach enables the system to switch between multiple processes seamlessly and ensure that no process hogs the CPU for an extended period, which could result in poor performance or even a system crash.

On the other hand, nonpreemptive scheduling allows a process to keep the CPU until it voluntarily releases it or completes its execution. In nonpreemptive scheduling, a process is responsible for deciding when to yield the CPU, which can lead to longer wait times for other processes that need to access the CPU.

Both preemptive and nonpreemptive scheduling have their advantages and disadvantages. Preemptive scheduling is more responsive and can ensure better utilization of the CPU, particularly in high-load scenarios. Nonpreemptive scheduling, on the other hand, can reduce the overhead of context switching, leading to better performance in scenarios where processes have similar priorities and durations.

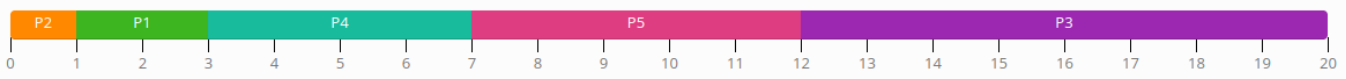
2. I/O-bound programs typically require a relatively short burst of CPU time and then spend more time waiting for I/O operations to complete. Therefore, this algorithm would favor I/O-bound programs as they would not require much CPU time and would have shorter wait times compared to CPU-bound programs.

Moreover, I/O-bound programs often relinquish the CPU voluntarily to wait for I/O operations to complete, allowing CPU-bound programs to get a chance to run. As a result, CPU-bound programs would not starve and would eventually receive their share of CPU time.

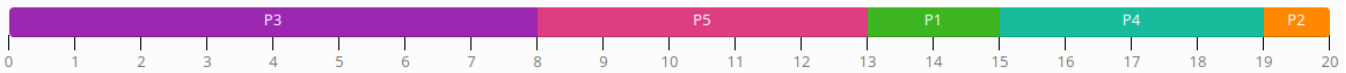
3.a)

FCFS:

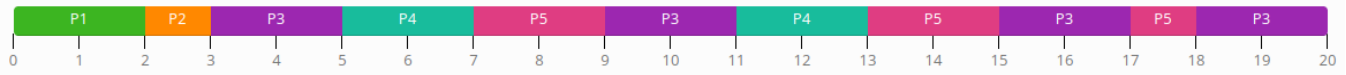
SJF:



Non-preemptive priority:



RR (quantum=2):



3.b) Turnaround time:

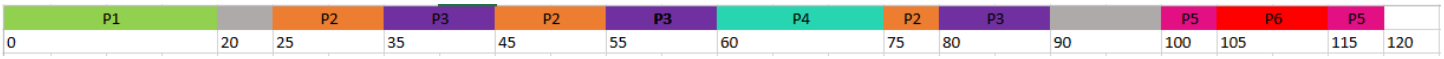
	FCFS	SJF	Priority	RR
P1	2	3	15	2
P2	3	1	20	3
P3	11	20	8	20
P4	15	7	19	13
P5	20	12	13	18

3.c) Waiting time:

	FCFS	SJF	Priority	RR
P1	0	1	13	0
P2	2	0	19	2
P3	3	12	0	12
P4	11	3	15	9
P5	15	7	8	13

3.d) SJF has the smallest waiting time.

4.a)



b)

	Turnaround time	Waiting time
P1	20	0
P2	55	40
P3	60	35
P4	15	0
P5	20	10
P6	10	0

c) CPU Utilization Rate = 105/120 * 100 = 87.5%