1.

A.

CPU utilization and response time:

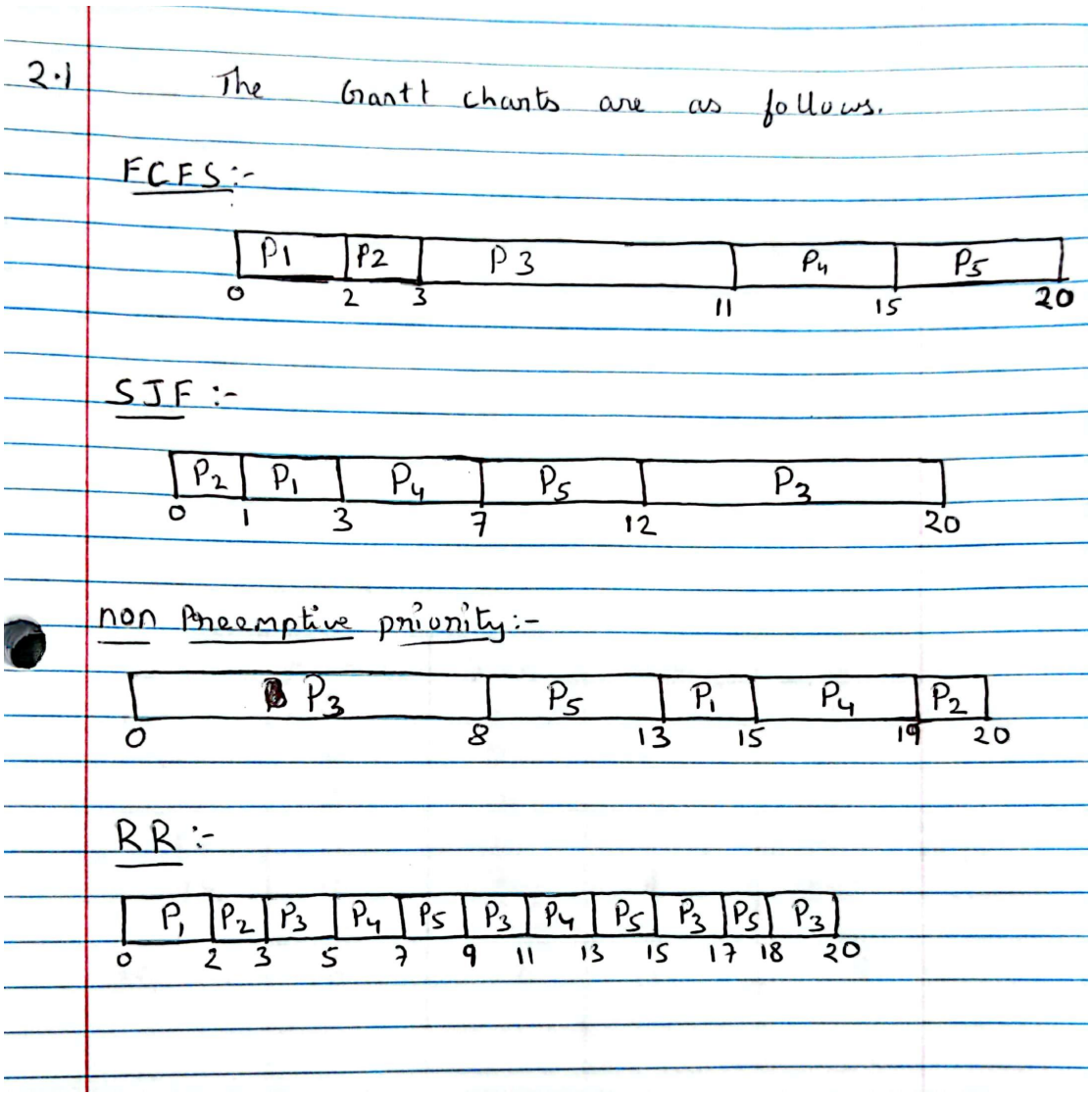
CPU utilization time is the time CPU is used. Response time is the time from the submission of a request until the first response is produced. CPU utilization is amplified if the overheads that are connected with context switching or alternating are minimized. The context switching outlay can be reduced by performing context switches occasionally. This could on the other hand lead to increasing the response time for processes.

B.

The Average turnaround time and maximum waiting time:

Turnaround time is the time between submission of a process to the time of completion. Waiting time is the amount of time a process spends waiting in the ready queue. Average turnaround time is reduced by implementing the shortest or simple tasks first. Such a scheduling and arrangement strategy could nevertheless starve long-running tasks and in so doing boost their overall waiting time.

2.



2.2 Turn around time :-

	FCFS	SJF	Priority	RR
P_1	2	3	15	2
P_2	3	1	20	3
P_3	11	20	8	20
P_4	15	7	19	13
P_5	20	12	13	18

2.3 Waiting time :-

	FCFS	SJF	Priority	RR
P_1	0	1	13	0
P_2	2	0	19	2
P_3	3	12	0	12
P_4	11	3	15	9
P_5	15	7	8	13
Average	6.2	4.6	11	7.2

2.4 Shortest Job First (SJF) resulted in the minimum average waiting time.

3.

2. Shortest job first 4. Priority could result in starvation.

Starvation is basically situation in which a process does not get required resources because other processes are being allocated to these resources and are utilizing those resources. So that process is in starvation as it does not get access to the resource and it is unable to execute.

In the First-come first-served scheduling algorithm the processes are executed in the order or sequence such that the first process is executed first and the second process is executed after the first process finishes its execution. So in this scheduling algorithm no starvation can occur as there is no process which will never get a resource and will never be able to execute.

Shortest job first chooses the process which has the shortest execution time from the ready queue to be executed first. As a result of that process with long execution times might suffer from starvation. Consider a situation where processes with short execution time keeps on adding and these processes are continuously picked leaving the process with long execution time to starve.

Round robin algorithm gives each process a fixed time slot called quantum time for its execution. So no process will have to wait for long and wait for a resource to be allocated for long in order to be executed. So this is a starvation free scheduling algorithm.

In priority scheduling the processes are given priorities to execute which means that the process with high priority will be allocated the resource and will be executed first. A process that is ready to run but waiting for the CPU can be considered blocked. A priority scheduling algorithm can leave some low priority processes waiting indefinitely. In a heavily loaded computer system, a steady stream of higher-priority processes can prevent a low-priority process from ever getting the CPU.

4.

FCFS will be resulted when b>a>0 because when b>a it means the priority for the processes which are currently executing is greater than the priority for the processes which are waiting in the queue. We know that FCFS is a non-preemptive algorithm, and when we have b>a it means its non-preemptive. Consider that a process is running and its priority is increasing at rate of b . Now another process arrives and waits in a queue at time t0. At time t1, priority of process which is running would be (t1-t0) * b while the one which is waiting would be (t1-t0) * a. Since b > a, running process will always have a higher priority. So the process in queue continues to wait . let there be another process entering the queue at time t2 the priority of the process already waiting in the queue is (t2-t1)*a and the priority of new process is zero hence the process which is already in the queue executes first. The process stated above is mimicking the FCFS algorithm, hence the condition b>a will lead to FCFS algorithm.