Process Scheduling

Scheduling is concerned with the way processes are assigned to the processor.

The purpose of scheduling is to:

- Improve user response time.
- Improve throughput (the time for an amount of data to be processed or moved from one place to another).
- Maximize processor efficiency.

Process Priorities

- The process scheduler will always choose a process of higher priority over one of lower priority.
- The scheduler may have multiple ready queues to represent each level of priority.
- Sometimes, lower-priority processes may suffer starvation i.e. indefinite access to the processor due to higher priority processes preempting them.

3 Types of Scheduling

- Long Term
- Medium Term
- Short Term

Long Term Scheduler

This is performed when a new process is created. It involves the decision to add to the pool of processes to be executed. It determines which programs are admitted to the system for processing.

Once admitted a job or user program becomes a process and usually added to the short-term queue. However, in some systems a newly created process begins in a swapped-out condition and is added to the queue for the medium term scheduler.

Long Term Scheduler (cont.)

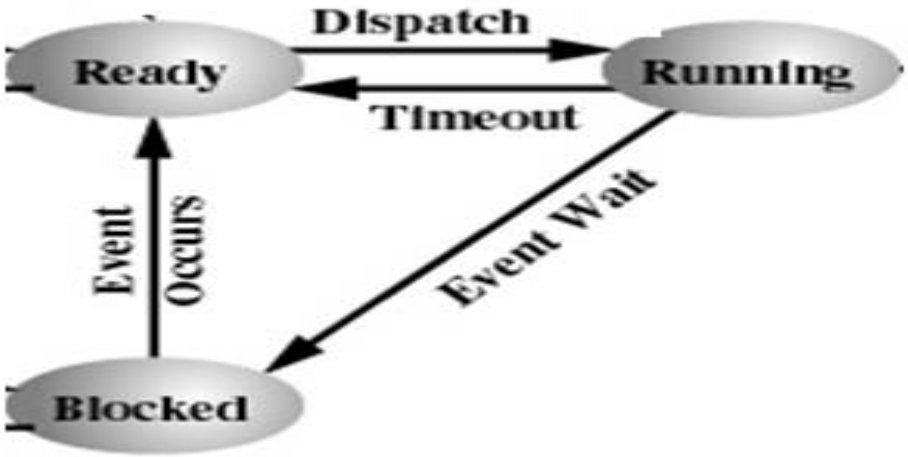
It involves moving a process from the new to ready or ready – suspend states.

Medium Term Scheduler

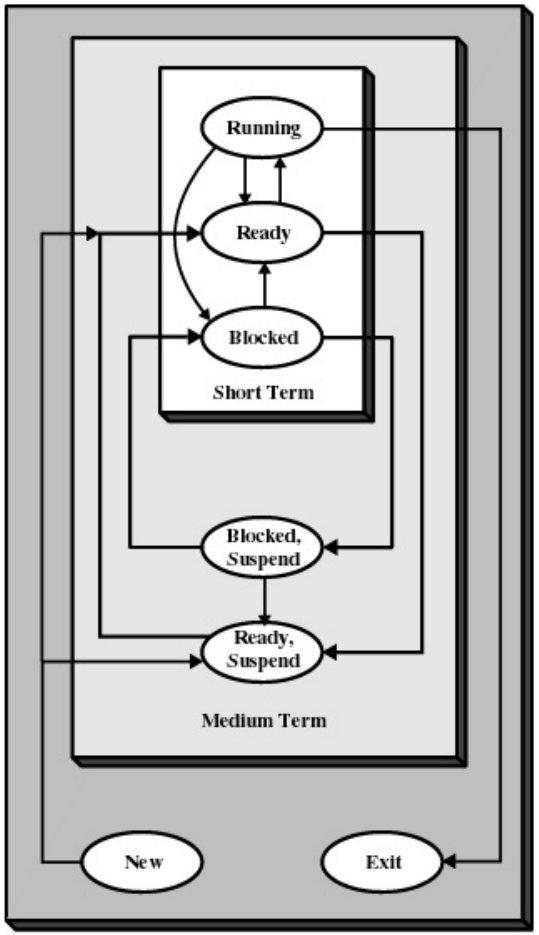
This makes the decision to add to the number of processes that are partially or fully in main memory. It is concerned with swapping in or swapping out processes from swap space.

Short Term Scheduler

This involves the decision as to which ready process will be executed by the processor. There are a number of short term scheduling algorithms which determine the priority of the processes.



Long, Medium + Short Scheduling



2 Types of Scheduling Algorithms

Non pre-emptive

- Once a process is in the running state, it will continue until it terminates or blocks itself for I/O.

Pre-emptive

- Currently running process may be interrupted and moved to the Ready state by the operating system. This allows for better service since any one process cannot monopolize the processor for very long.

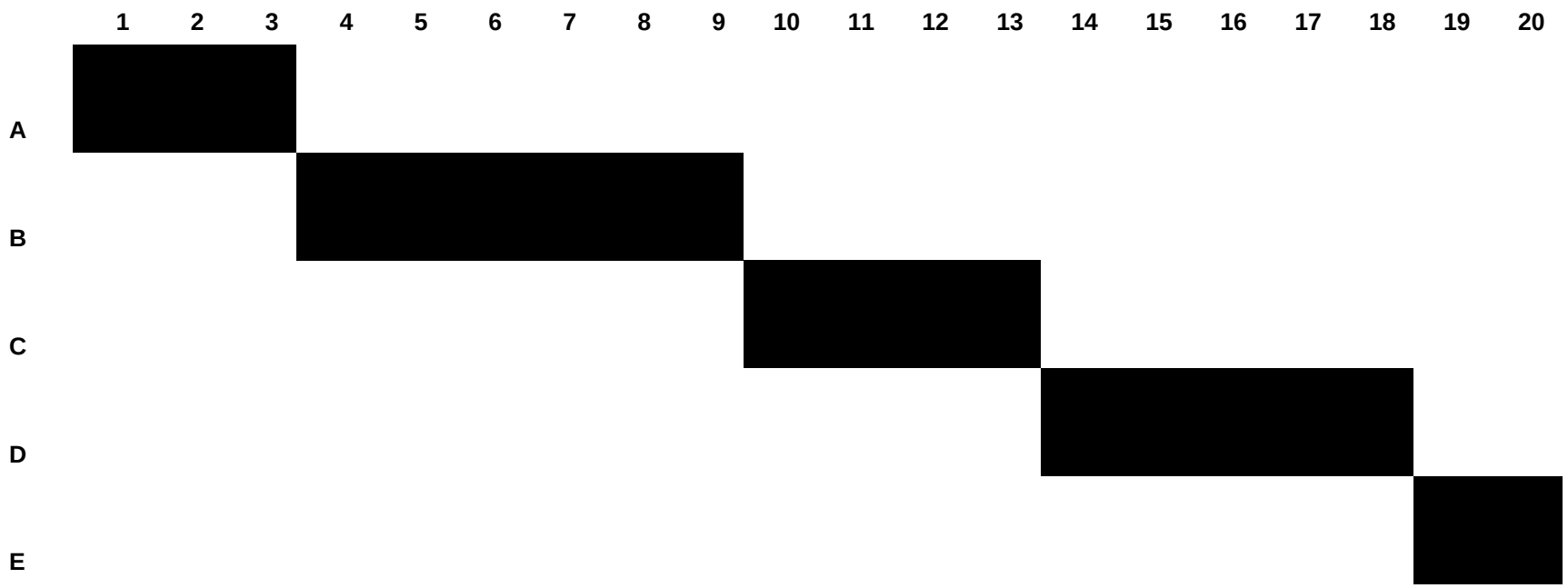
Examples

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Arrival Time – this is the instant the process arrives to the Ready State

Service Time – the time required to run the process

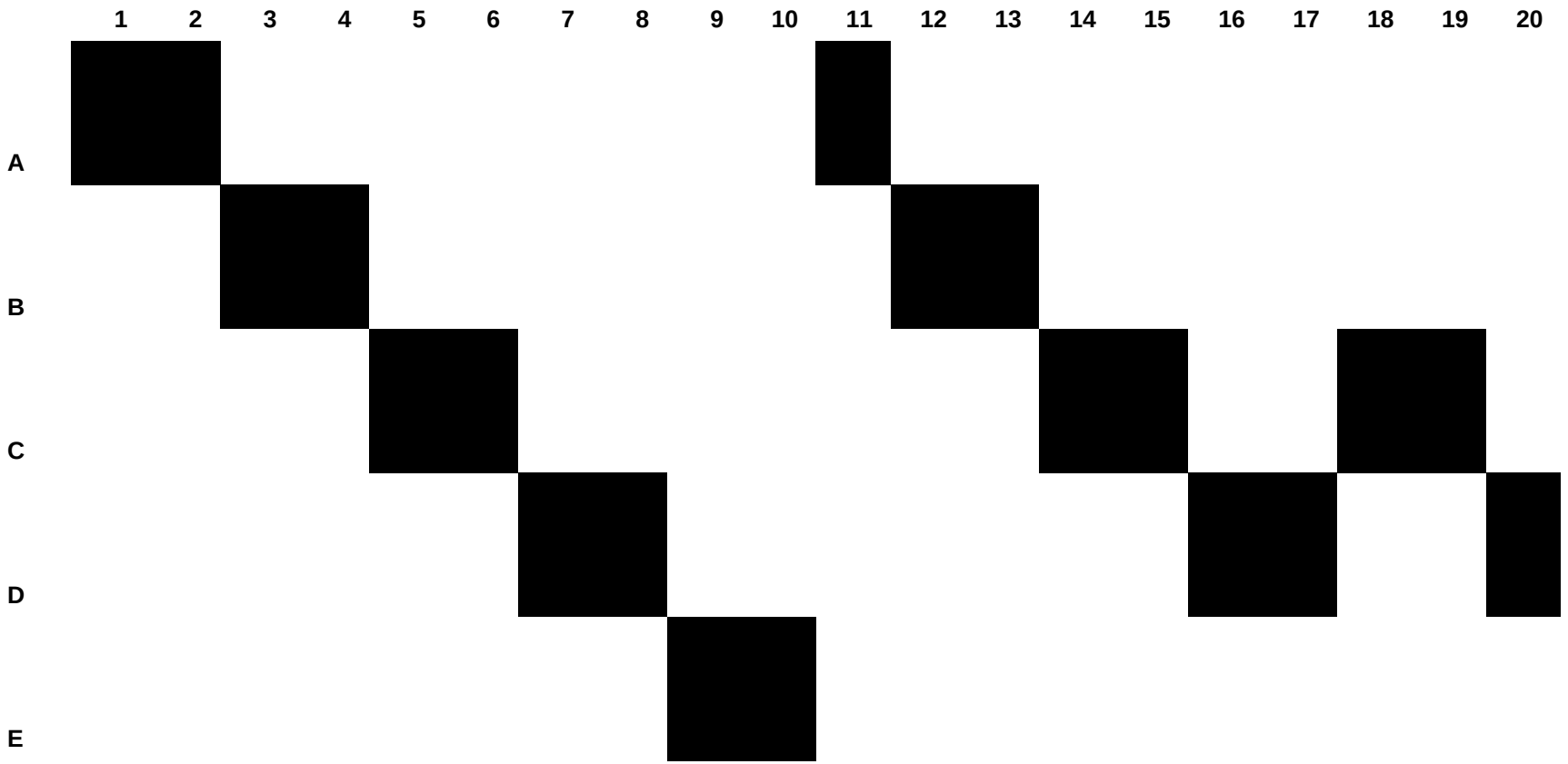
First Come First Served



Round Robin

- Clock interrupt is generated at periodic intervals.
- When an interrupt occurs, the currently running process is placed in the read queue - Next ready job is selected.
- Known as time slicing.
- Example has Time Slice (Quantum) = 2

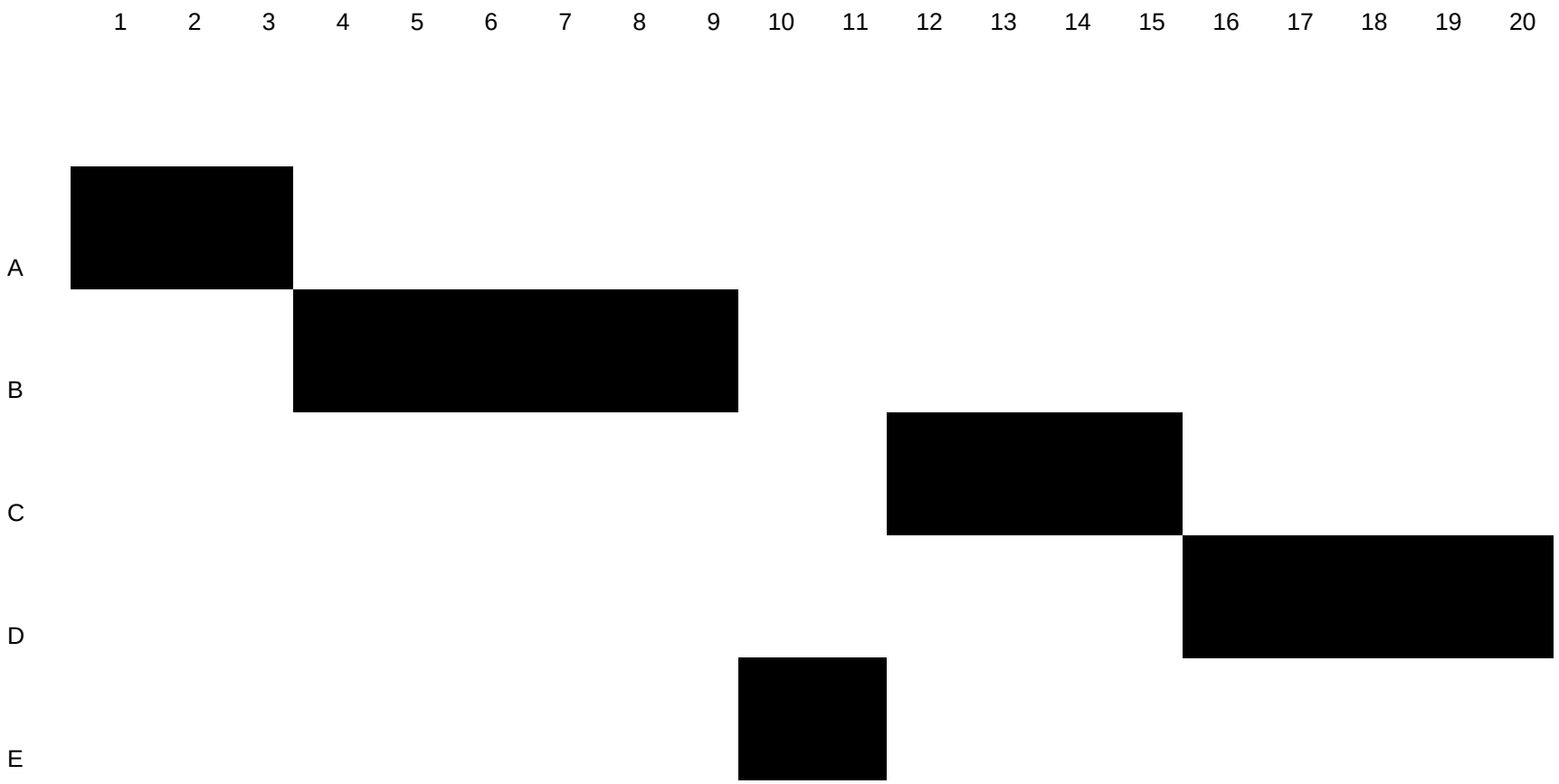
Round Robin



Shortest Process Next

- Process with shortest time gets priority
- Non pre-emptive
- Possibility of starvation for longer processes

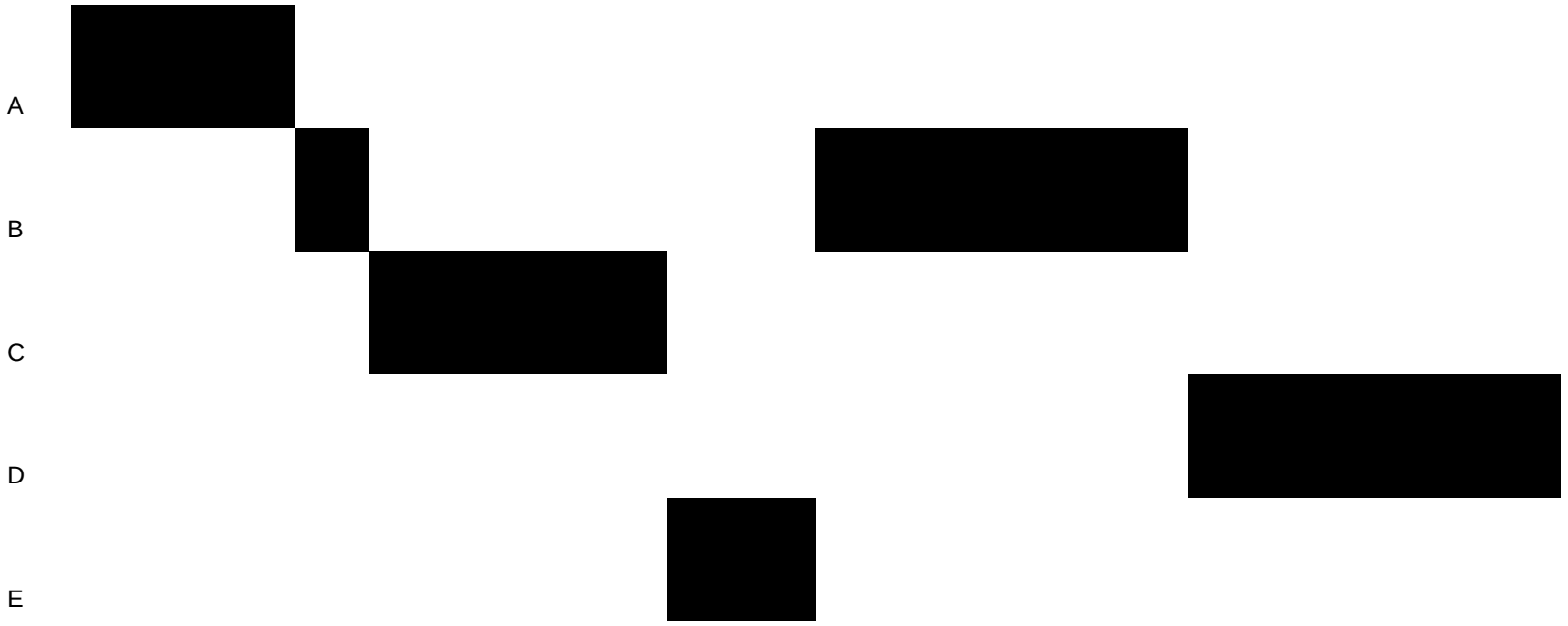
Shortest Process Next



Shortest Remaining Time

- Similar to SPN
- Pre-emptive
- Process with shortest time gets priority
- Possibility of starvation for longer processes

Shortest Remaining Time



Turnaround Time & NTT

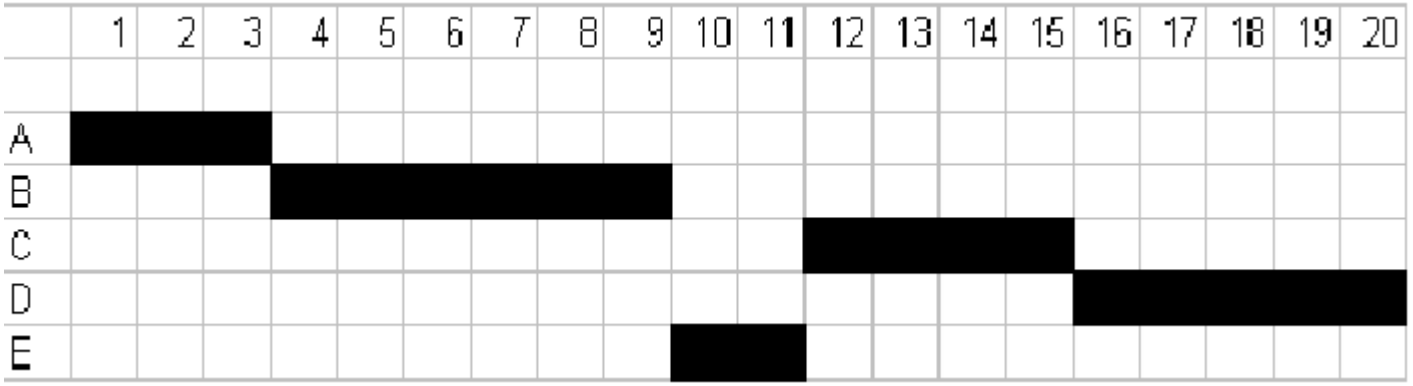
Turnaround Time

- The turnaround time is the total time a process is waiting in the queue and being serviced. If a process is waiting for 4 time units and its service time is 3 units then its turnaround time is 7.

Normalised Turnaround Time

- This is the ratio of the turnaround time to the service.

Turnaround Time & NTT



	A	S	T	NTT
A	0	3	3	1
B	2	6	7	1.17
C	4	4	11	2.75
D	6	5	14	2.8
E	8	2	3	1.5

Other scheduling algorithms

Other scheduling algorithm also exist such as:

Highest Response Ratio Next

Priority of a process is based on a ratio of the waiting time plus service time to the service time i.e. (w+s) / s.

This reduces instances of starvation.

Multilevel feedback queue: processes are moved across different priority queues – penalises jobs which have been running longest.

Exercise

Consider the following set of processes

Process	Arrival Time	Service
P1	0	2
P2	2	6
P3	4	3
P4	5	1

Exercise (cont.)

Draw a time line to show the execution of each of these processes for the scheduling algorithms First Come First Served, **Round Robin**, **Shortest Process Next** and the **Shortest Remaining Time**.

For each algorithm calculate the **Turnaround Time** and **Normalised Turnaround Time** for each process.