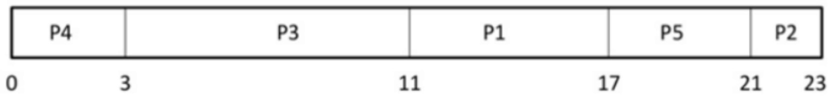
FCFS

https://mycareerwise.com/content/priority-scheduling/content/exam/gate/computer-science

1)

Process	Burst time	Arrival time
P1	6	2
P2	2	5
P3	8	1
P4	3	0
P5	4	4




```
Waiting time = Start time - Arrival time
```

P4 = 0-0 = 0

P3 = 3-1 = 2

PI = 11-2 = 9

P5= 17-4 = 13

P2= 21-5= 16

Average Waiting Time

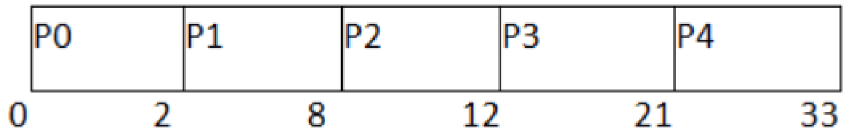
$$\frac{0+2+9+13+16}{5}$$

= 40/5

= 8

2) FCFS (Calculate Completion Time, Turn Around Time ,Waiting time)

Process ID	Arrival Time	Burst Time
0	0	2
1	1	6
2	2	4
3	3	9
4	6	12



Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
0	0	2	2	2	0
1	1	6	8	7	1
2	2	4	12	10	6
3	3	9	21	18	9
4	6	12	33	29	17

Avg Waiting Time=31/5

3) FCFS

Consider the set of 5 processes whose arrival time and burst time are given below-

Process Id	Arrival time	Burst time
P1	3	4
P2	5	3
P3	0	2
P4	5	1
P5	4	3

If the CPU scheduling policy is FCFS, calculate the average waiting time and average turn around time.

Gantt Chart

Here, black box represents the idle time of CPU.

Now, we know-

- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

Also read- Various Times of Process

Process Id	Exit time	Turn Around time	Waiting time
P1	7	7 – 3 = 4	4 – 4 = 0
P2	13	13 – 5 = 8	8 – 3 = 5
P3	2	2 – 0 = 2	2 – 2 = 0
P4	14	14 – 5 = 9	9 – 1 = 8
P5	10	10 – 4 = 6	6 – 3 = 3

Now,

- Average Turn Around time = (4 + 8 + 2 + 9 + 6) / 5 = 29 / 5 = 5.8 unit
- Average waiting time = (0 + 5 + 0 + 8 + 3) / 5 = 16 / 5 = 3.2 unit

Criteria - Burst time

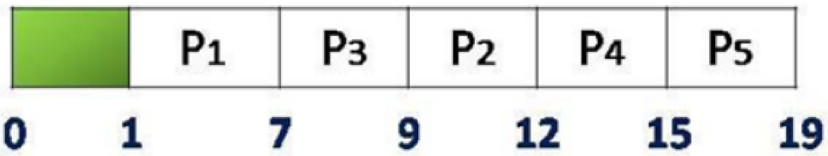
Mode - Non-preemptive

Data-structure - Min Heap is the efficient data structure of SJF.

Example:

Process No.	Arrival Time (AT)	Burst Time (BT)
1	1	6
2	2	3
3	3	2
4	4	3
5	5	4

Gantt chart:



TAT = CT - AT and WT = TAT - BT

Process No.	AT	BT	CT	TAT	WT
1	1	6	7	(7 – 1) = 6	(6 – 6) = 0
2	2	3	12	(12 – 2) = 10	(10 – 3) = 7
3	3	2	9	(9 – 3) = 6	(6 – 2) = 4
4	4	3	15	(15 – 4) = 11	(11 – 3) = 8
5	5	4	19	(19 – 5) = 14	(14 – 4) = 10

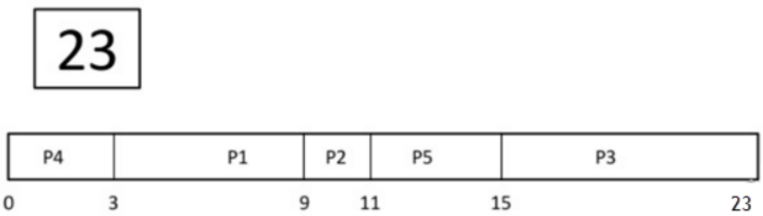
SJF also suffers from the **convoy effect** i.e. if the larger process comes earlier then **average waiting time** will be increased.

Non-Preemptive SJF

In non-preemptive scheduling, once the CPU cycle is allocated to process, the process holds it till it reaches a waiting state or terminated.

Consider the following five processes each having its own unique burst time and arrival time.

Process Queue	Burst time	Arrival time
P1	6	2
P2	2	5
P3	8	1
P4	3	0
P5	4	4



Step 11) Let's calculate the average waiting time for above example.

```
Wait time
P4= 0-0=0
P1=  3-2=1
P2= 9-5=4
P5= 11-4=7
P3= 15-1=14
```

```
Average Waiting Time= 0+1+4+7+14/5 = 26/5 = 5.2
```

Shortest Remaining Time First (SRTF) (Preemptive SJF)

SRTF process and examples:

The shortest remaining time First (SRTF) algorithm is **preemptive** version of **SJF**. In this algorithm, the scheduler always chooses the processes that have the shortest expected remaining processing time. When a new process joins the ready queue it may in fact have a shorter remaining time than the currently running process. Accordingly, the scheduler may preempt the current process when a new process (with a shorter burst time) becomes ready.

This feature helps to improve the **turnaround times** and **weighted turnarounds of processes**.

After every one unit of processing current process, scheduler can check any processes available at ready queue with shorter burst time.

Criteria: BT (Burst Time) + AT (Arrival Time)
Mode: Preemptive

Process No.	Arrival Time (AT)	Burst Time (BT)
P1	0	8
P2	1	4
P3	2	9
P4	3	5

Gantt Chart

Process P1 is started at time 0, since it is the only process in the queue. Process P2 arrives at time 1. The remaining time for process P1 (7 milliseconds) is larger than the time required by process P2 (4 milliseconds), so process P1 is preempted, and process P2 is scheduled.

TAT = CT - AT, WT = TAT – BT
And
Response Time (RT) = FR (First Response) – AR (Arrival Time)

P_No.	AT	BT	CT	TAT	WT	First Response (FR)	RT
P1	0	8	17	(17 – 0) = 17	(17 – 8) = 9	0	(0 – 0) = 0
P2	1	4	5	(5 – 1) = 4	(4 – 4) = 0	1	(1 – 1) = 0
P3	2	9	26	(26 – 2) = 24	(24 – 9) = 15	17	(17 – 2) = 15
P4	3	5	10	(10 – 3) = 7	(7 – 5) = 2	5	(5 – 3) = 2

$$\textbf{AVG Turn Around Time (ATAT)} = \frac{17 + 4 + 24 + 7}{4} = 13$$

$$\textbf{AVG Waiting Time (AWT)} = \frac{9 + 0 + 15 + 2}{4} = 6.5$$

In this algorithm there is no chance to occur **convoy effect** (no starvation). But **long processes** might face **starvation**. Like in the above example burst time of P3 is high as a result Response Time (RT) of P3 is pretty high (RT=15).

In general this algorithm is not used, because calculate burst time previously is not possible. But if we can predict the burst time (**See prediction method**) in a better way, it can be used.

Example: (GATE: 2007)

Process No.	Arrival Time (AT)	Burst Time (BT)
P1	0	20
P2	15	25
P3	30	10
P4	45	15

Calculate the waiting time of process P2 using Shorter Remaining Time First (SRTF).

Gantt Chart as per **SRTF**.

Note: If the arrival time is long then don't waste your time stopping after every unit. We can just stop at the next arrival time then check and carry on.

Now we calculate **Turn Around Time (TAT)** and **Waiting Time (WT)** using the following formula:

TAT = CT – AT and WT = TAT – BT.

Process No.	AT	BT	CT	TAT	WT
1	0	~~20~~ 0	20	20	0
2	15	~~25~~ ~~15~~ 10	55	40	15
3	30	10	40	10	0
4	45	15	70	25	10

Answer: Waiting time of P2 is **15**.

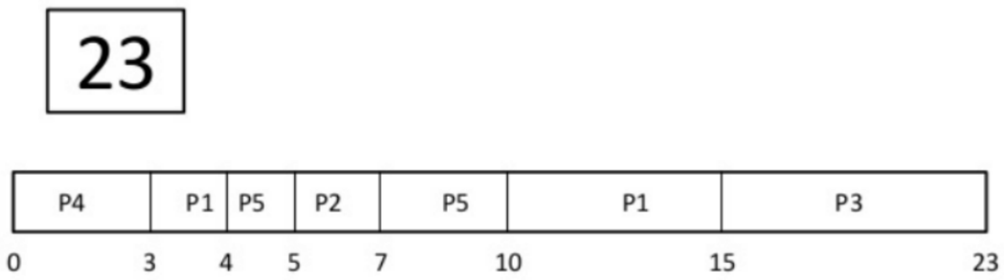
Preemptive SJF

In Preemptive SJF Scheduling, jobs are put into the ready queue as they come. A process with shortest burst time begins execution. If a process with even a shorter burst time arrives, the current process is removed or preempted from execution, and the shorter job is allocated CPU cycle.

Consider the following five process:

Process Queue	Burst time	Arrival time
P1	6	2
P2	2	5
P3	8	1
P4	3	0
P5	4	4

Step 0) At time=0, P4 arrives and starts execution.



Step 11) Let's calculate the average waiting time for above example.

```
Wait time
P4= 0-0=0
P1=  (3-2) + 6 =7
P2= 5-5 = 0
P5= 4-4+2 =2
P3= 15-1 = 14
```

```
Average Waiting Time = 0+7+0+2+14/5 = 23/5 =4.6
```