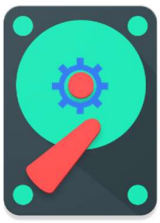


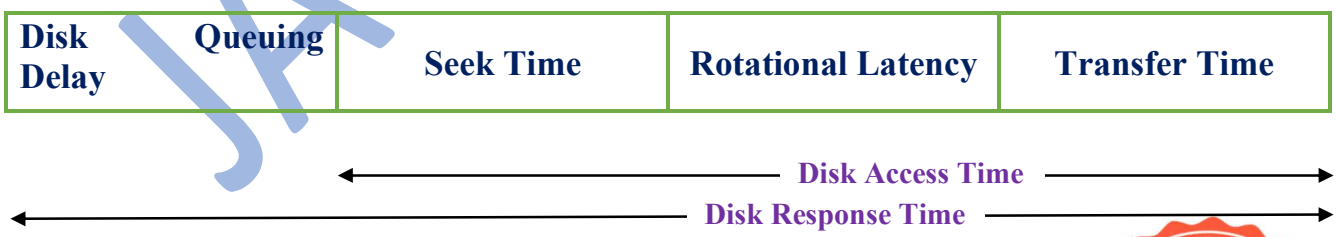
DISK SCHEDULING ALGORITHM

The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth. For this **Disk scheduling** is done by operating systems to schedule I/O requests arriving for the disk.

There are many **Disk Scheduling Algorithms** but before discussing them let's have a quick look at some of the important terms:

- **Seek Time:** It is the time for the disk are to move the heads to the cylinder containing the desired sector.
- **Rotational Latency:** It is the additional time waiting for the disk to rotate the desired sector to the disk head.
- **Transfer Time:** Transfer time is the time to transfer the data. It depends on the rotating speed of the disk and number of bytes to be transferred.
- **Disk Access Time:**

Disk Access Time = Rotational Latency + Seek Time + Transfer Time



Now, Let's discuss the **Disk Scheduling Algorithms**.

✓ Each algorithm is carrying some advantages and disadvantages!

✓ The limitation of each algorithm leads to the evolution of a new algorithm!

1. FIFO: FIRST IN FIRST OUT SCHEDULING

- **First in first out** scheduling follows the **first come first serve method**.
- The requests are addressed in the order they arrive in the disk queue.
- The average waiting time for FIFO is often quite long.
- It is non pre-emptive.

Example:

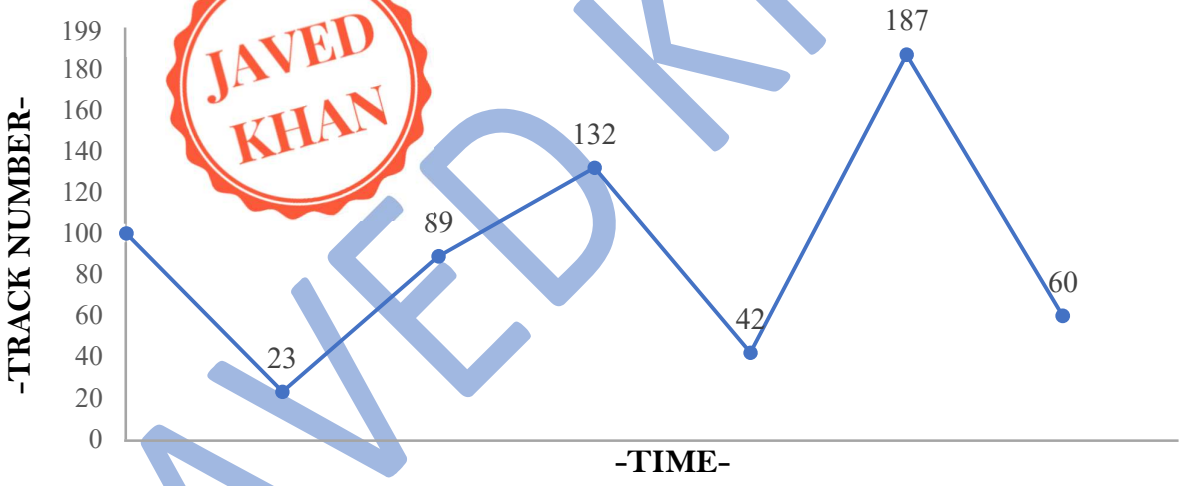
A disk of size **200** tracks receives the request for the block for I/O operation. Number of blocks requested on the disk are **23,89,132,42,187,60**. The disk arm is initially at track **100**. Find out the total number of head/track movement using FIFO disk scheduling algorithm.

Solution:

Here, Given that:

Work Queue = 23, 89, 132, 42, 187, 60

There are 200 cylinders numbered from 0 – 199.
The disk head stars at number 100.



Total number of track movement

= **|100-23| + |89-23| + |132-89| + |132-42| + |187-42| + |187-60|**

= **77 + 66 + 43 + 90 + 145 + 127**

= **548**

+Advantages+

1. Simple, fair to all requests.
2. No starvation.

-Disadvantages-

1. Not efficient because the average seek time is **high**.

---

2. SSTF: SHORTEST SEEK TIME FIRST SCHEDULING

- It selects the request with the minimum seek time from the current head position
- **SSTF** is certainly an improvement over **FCFS/FIFO** as it decreases the average response time and increases the **throughput** of system.

Example:

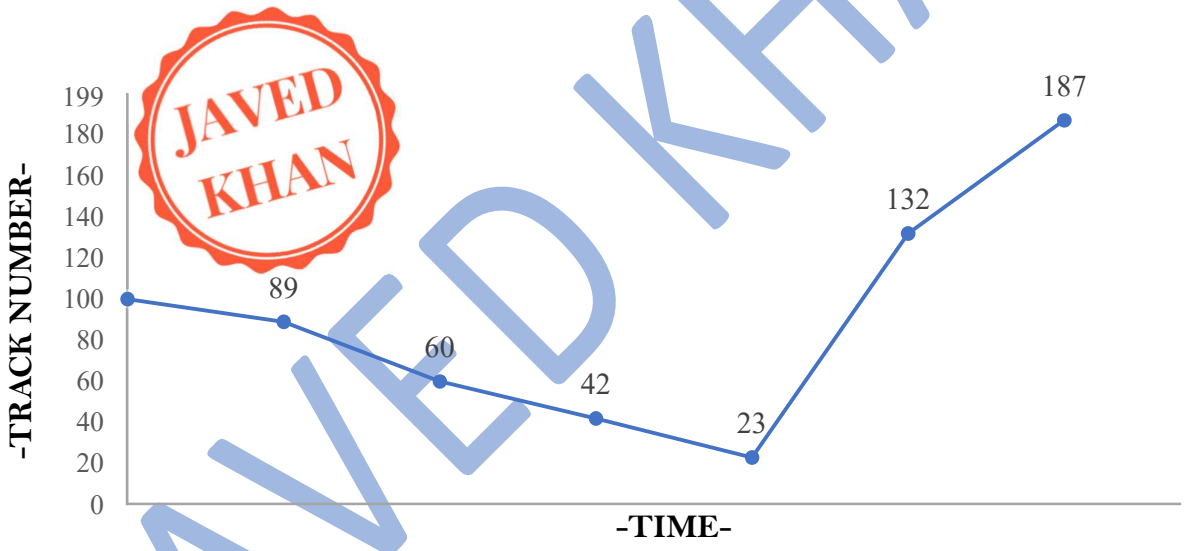
A disk of size **200** tracks receives the request for the block for I/O operation. Number of blocks requested on the disk are **23,89,132,42,187,60**. The disk arm is initially at track **100**. Find out the total number of head/track movement using SSTF disk scheduling algorithm.

Solution:

Here, Given that:

Work Queue: **23, 89, 132, 42, 187, 60**

There are 200 cylinders numbered from 0 – 199.
The disk head stars at number 100.



Total number of track movement

= |100-23| + |187-23|

= 77 + 164

= 241

+Advantages+

1. More efficient than **FCFS**.
2. **Average Response Time** decreases and **throughput** increases.

-Disadvantages-

1. **Starvation** is possible for request involving longer seek-time.
2. Overhead to calculate seek time in advance.

---

3. SCAN SCHEDULING

- The disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of disk, it reverses its direction and again services the request arriving in its path.
- This algorithm works as an elevator therefore it is also known as **elevator algorithm**.

Example:

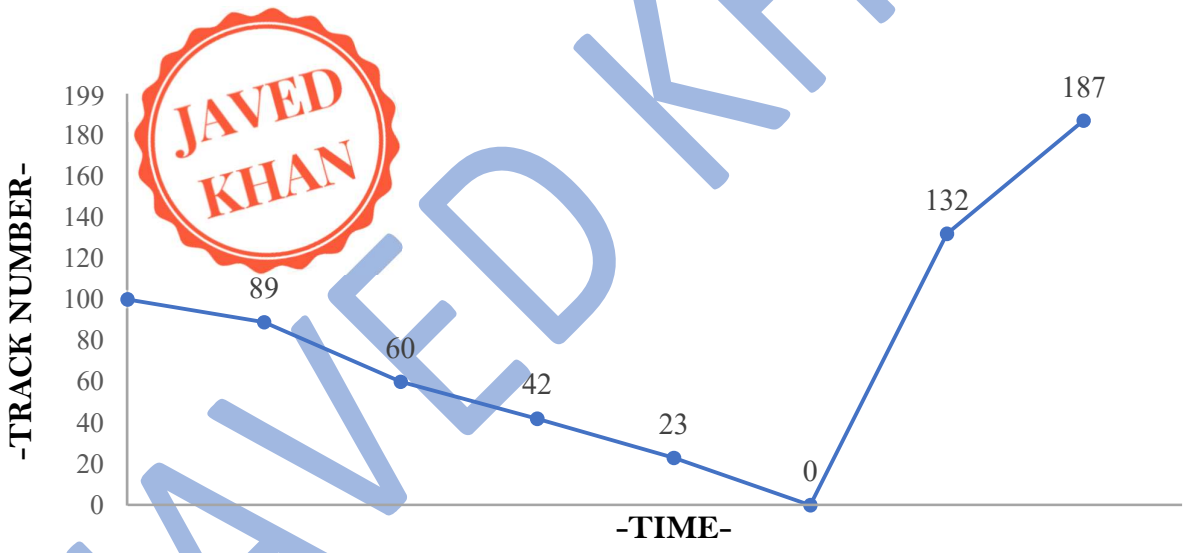
A disk of size **200** tracks receives the request for the block for I/O operation. Number of blocks requested on the disk are **23,89,132,42,187,60**. The disk arm is initially at track **100**. Find out the total number of head/track movement using SCAN scheduling algorithm.

Solution:

Here, Given that:

Work Queue: **23, 89, 132, 42, 187, 60**

There are 200 cylinders numbered from 0 – 199.
The disk head stars at number 100.



Total number of track movement

= **|100-0| + |187-0|**

= **100 + 187**

= **287**

+Advantages+

1. More efficient than **FCFS**.
2. No **starvation** for any request.

-Disadvantages-

1. Long waiting time for requests for locations just visited by disk arm.
2. Not so fair, because cylinder which are just behind the head will wait longer.

---

4. CIRCULAR SCAN SCHEDULING

- The disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of disk, it reverses its direction and again services the request arriving in its path.
- This algorithm works as an elevator therefore it is also known as **elevator algorithm.**

Example:

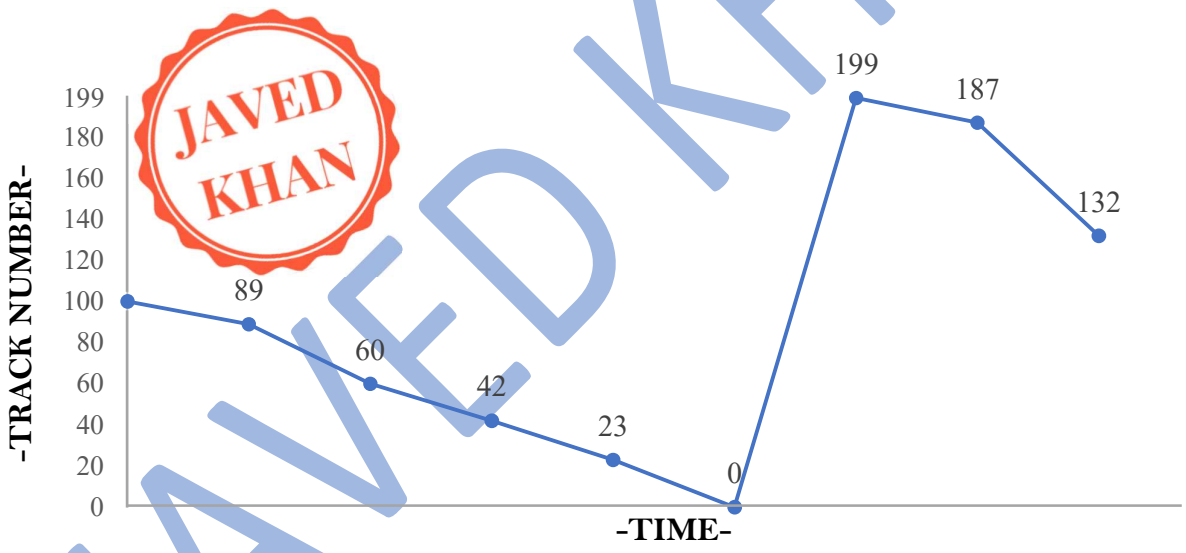
A disk of size **200** tracks receives the request for the block for I/O operation. Number of blocks requested on the disk are **23,89,132,42,187,60**. The disk arm is initially at track **100**. Find out the total number of head/track movement using C-SCAN scheduling algorithm.

Solution:

Here, Given that:

Work Queue: **23, 89, 132, 42, 187, 60**

There are 200 cylinders numbered from 0 – 199.
The disk head stars at number 100.



Total number of track movement

= **|100-0| + |199-0| + |199-132|**

= **100 + 199 + 67**

= **366**

+Advantages+

1. Provides more uniform wait time compared to **SCAN**.
2. It provides better response time.

-Disadvantages-

1. It causes more seek movements as compared to **SCAN** Algorithm.
2. It causes the head to move till the end of the disk even if there are no requests to be serviced.

---

5. LOOK SCHEDULING

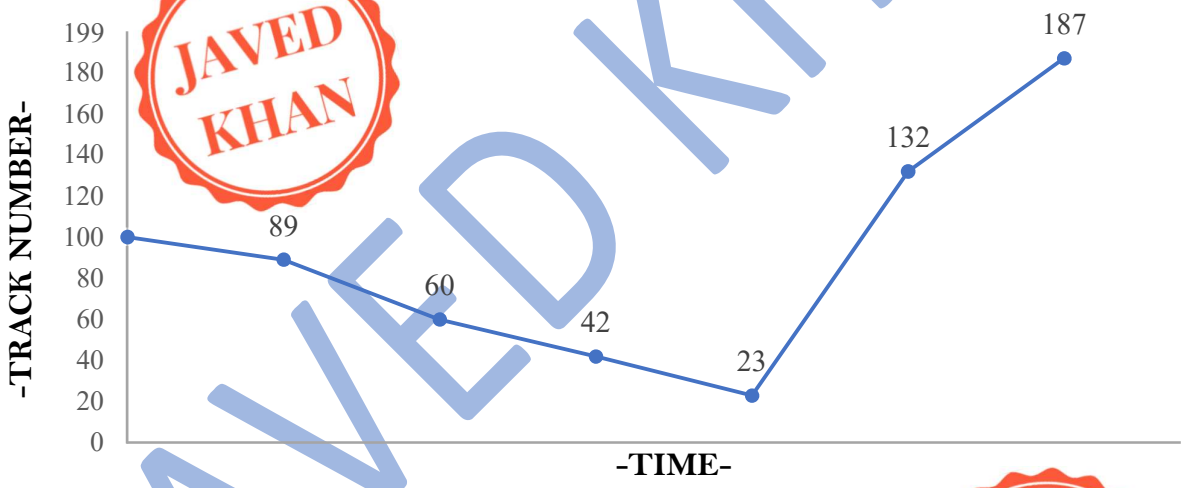
- It is same like **SCAN Scheduling** but the difference is that the end points are not visited unnecessarily.

Example:
A disk of size **200** tracks receives the request for the block for I/O operation. Number of blocks requested on the disk are **23,89,132,42,187,60**. The disk arm is initially at track **100**. Find out the total number of head/track movement using LOOK scheduling algorithm.

Solution:
Here, Given that:

Work Queue: **23, 89, 132, 42, 187, 60**

There are 200 cylinders numbered from 0 – 199.
The disk head stars at number 100.



Total number of track movement

= **|100-23| + |187-23|**
= **77 + 164**
= **241**

+Advantages+

1. It prevents the extra delay which occurred due to unnecessary traversal to the end of disk.
2. It provides better performance as compared to **SCAN Algorithm**.

-Disadvantages-

1. There is an overhead of finding the end requests.
2. It causes long waiting time for the cylinders just visited by the head.

---

6. CIRCULAR LOOK SCHEDULING

- It is a version of **C-SCAN** Disk Scheduling Algorithm.
- Head starts from the first request at one end of the disk and moves towards the last request at the other end servicing all the requests in between. After reaching the last request at the other end, head reverses its direction. It then returns to the first request at the starting end without servicing any request in between and the same process is repeated.

Example:

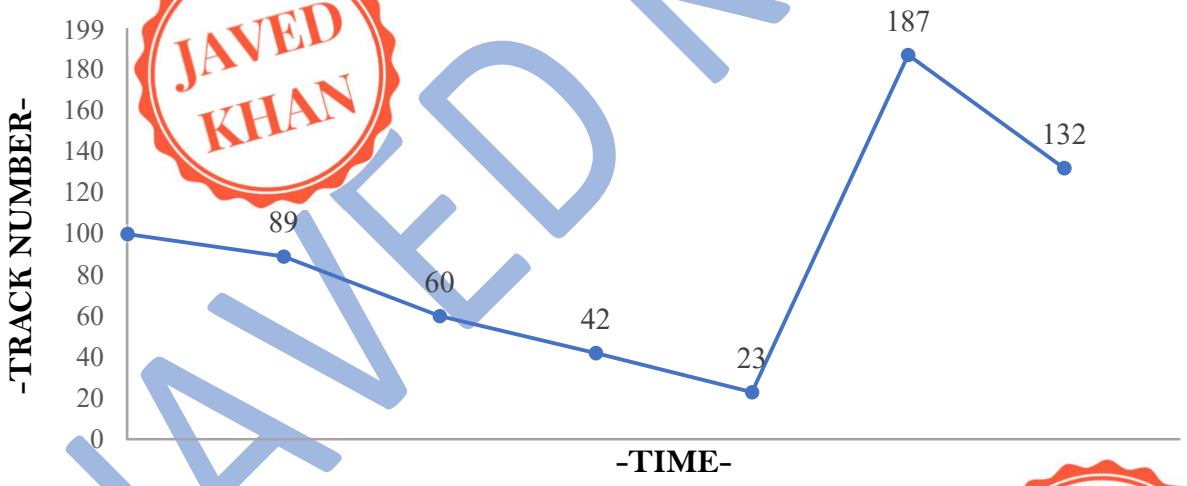
A disk of size **200** tracks receives the request for the block for I/O operation. Number of blocks requested on the disk are **23,89,132,42,187,60**. The disk arm is initially at track **100**. Find out the total number of head/track movement using C-LOOK scheduling algorithm.

Solution:

Here, Given that:

Work Queue: **23, 89, 132, 42, 187, 60**

There are 200 cylinders numbered from 0 – 199.
The disk head stars at number 100.



Total number of track movement

= **|100-23| + |187-23| + |187-132|**

= **77 + 164 + 55**

= **296**

+Advantages+

1. It reduces the waiting time for the cylinders just visited by the head.
2. It provides better performance as compared to **LOOK Algorithm**.
3. It provides low variance in response time and waiting time.

-Disadvantages-

1. There is an overhead of finding the end requests.

Points To Remember While Selecting a Disk Scheduling Algorithm –

- **SSTF** is common and has a natural appeal.
- **SCAN** and **C-SCAN** perform better for systems that place a heavy load on the disk.
- Performance depends on the number and types of requests.
- Requests for disk service can be influenced by the **file-allocation method**.
- The disk-scheduling algorithm should be written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary.
- Either **SSTF** or **LOOK** is a reasonable choice for the default algorithm.

Important Points To Be Noted –

- **Disk Scheduling** is also known as **I/O Scheduling**.
- **Disk Response Time** is the average of time spent by each request waiting for the I/O operation.
- **Disk Bandwidth** is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.
- **Search Time** (rotational delay):
  Time taken to rotate **DASD** i.e., **D**irect **A**ccess **S**torage **D**isk (Example: Magnetic drum and disk). Until requested record is under read/write head.
- **SSTF** disk scheduling is a form of **SJF scheduling**.
- In **SSTF** the seek time of every request is calculated in advance in the queue and then they are scheduled according to their calculated seek time.
- **SCAN Scheduling Algorithm** is also known as **Elevator Algorithm**.
- **C-SCAN** is another word for **Circular Scan**.
- **C-LOOK** is another word for **Circular Look**.

Concept By: Mr. Javed Khan (B.Sc., MCM, MCA, MBA Systems)