Disk Scheduling

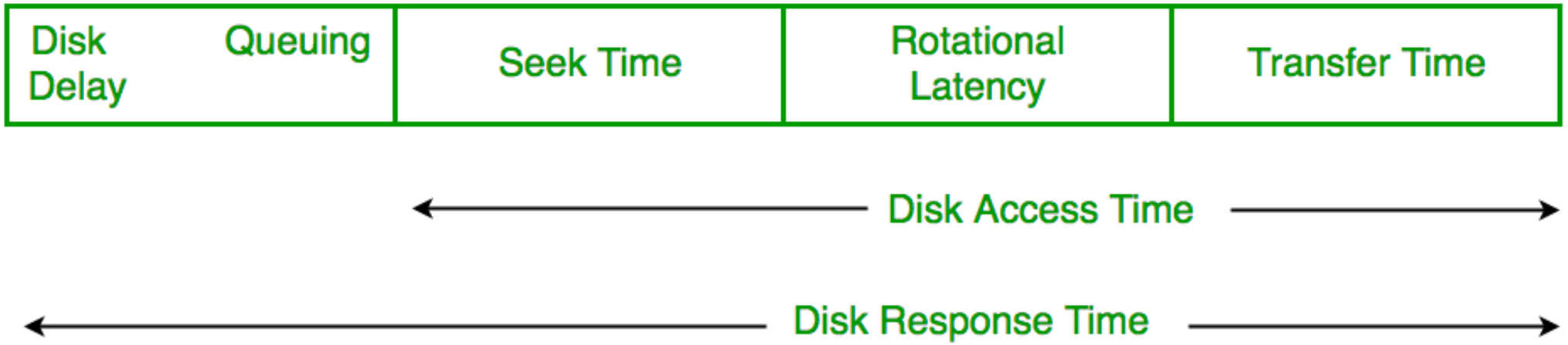
Disk scheduling algorithms

- The main goals of disk scheduling are to optimize the performance of disk operations, reduce the time it takes to access data and improve overall system efficiency.
- DSA are used to managing how data is read from and written to a computer’s hard disk.
- These algorithms help determine the order in which disk read and write requests are processed, significantly impacting the speed and efficiency of data access.
- Disk scheduling is a technique operating systems use to manage the order in which disk I/O (input/output) requests are processed.
- Disk scheduling is also known as I/O Scheduling.

Key Terms Associated with Disk Scheduling

- **Seek Time:** Seek time is the time taken to locate the disk arm to a specified track where the data is to be read or written. So the disk scheduling algorithm that gives a minimum average seek time is better.
- **Rotational Latency:** Rotational Latency is the time taken by the desired sector of the disk to rotate into a position so that it can access the read/write heads. So the disk scheduling algorithm that gives minimum rotational latency is better.
- **Transfer Time:** Transfer time is the time to transfer the data. It depends on the rotating speed of the disk and the number of bytes to be transferred.
- **Disk Access Time:**

- **Disk Access Time**
- *Disk Access Time = Seek Time + Rotational Latency + Transfer Time*
- *Total Seek Time = Total head Movement * Seek Time*
- **Disk Response Time:** Response Time is the average time spent by a request waiting to perform its I/O operation. The average *Response time* is the response time of all requests.

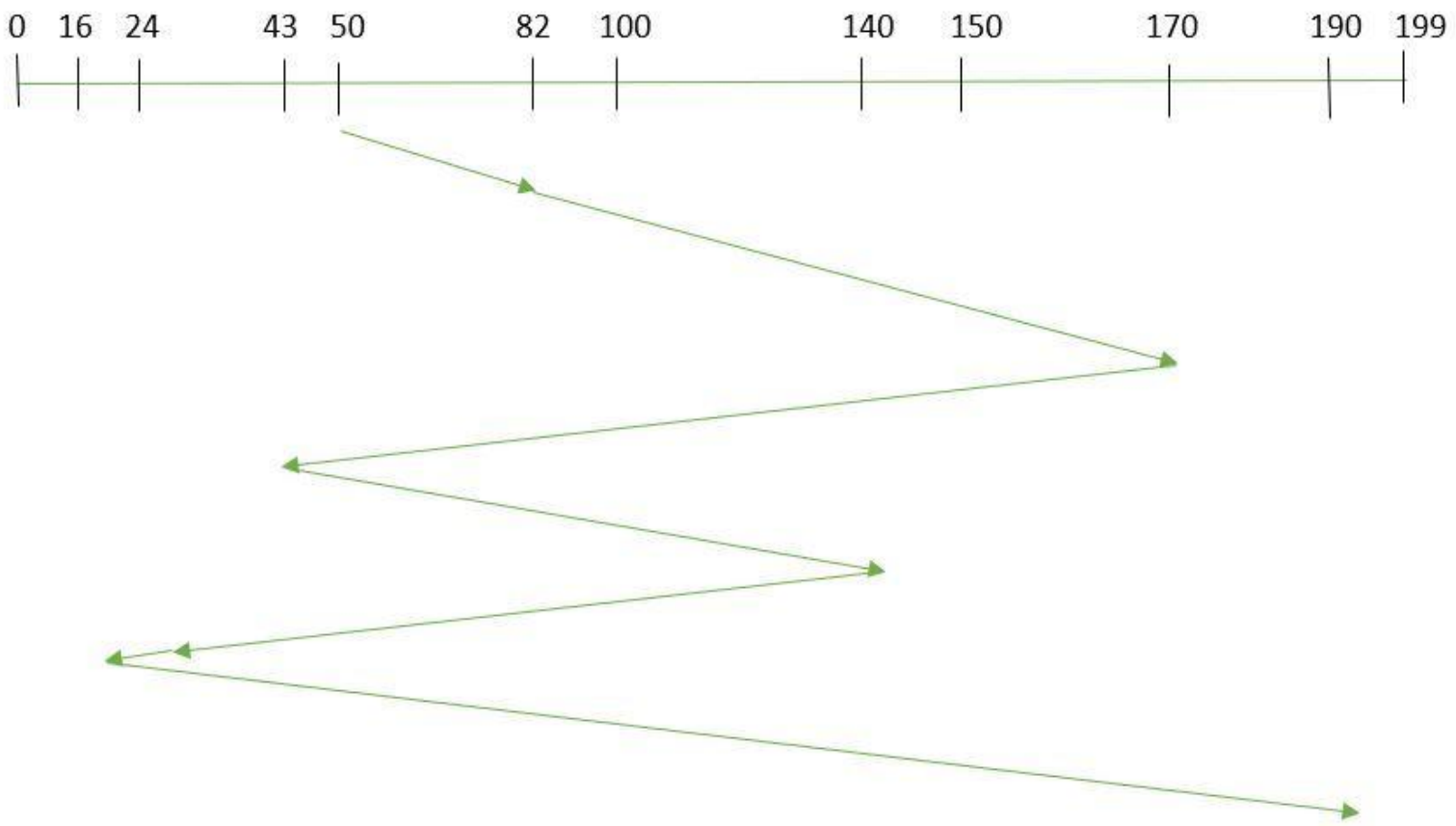


Disk Scheduling Algorithms

- FCFS (First Come First Serve)
- SSTF (Shortest Seek Time First)
- SCAN
- C-SCAN
- LOOK

FCFS (First Come First Serve)

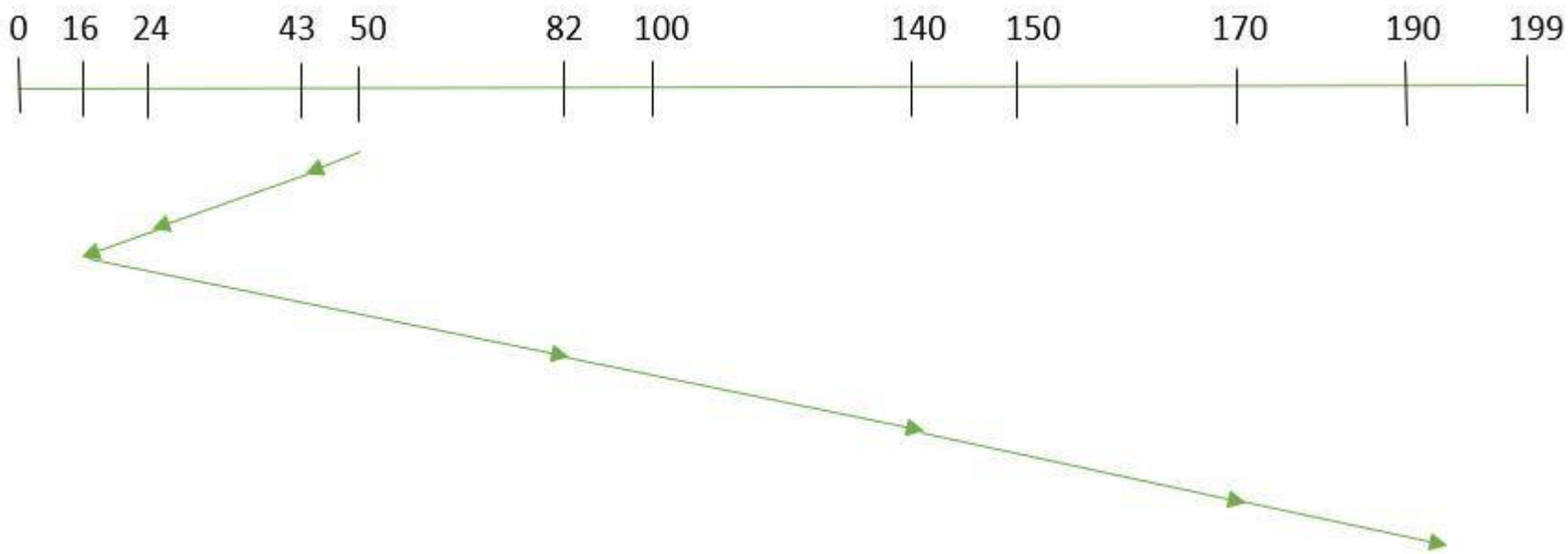
- FCFS is the simplest of all Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue. Let us understand this with the help of an example.



- Suppose the order of request is- (82,170,43,140,24,16,190)
  And current position of Read/Write head is: 50
- So, total overhead movement (total distance covered by the disk arm) = (82-50)+(170-82)+(170-43)+(140-43)+(140-24)+(24-16)+(190-16) =642

SSTF (Shortest Seek Time First)

- In SSTF (Shortest Seek Time First), requests having the shortest seek time are executed first. So, the seek time of every request is calculated in advance in the queue and then they are scheduled according to their calculated seek time. As a result, the request near the disk arm will get executed first.

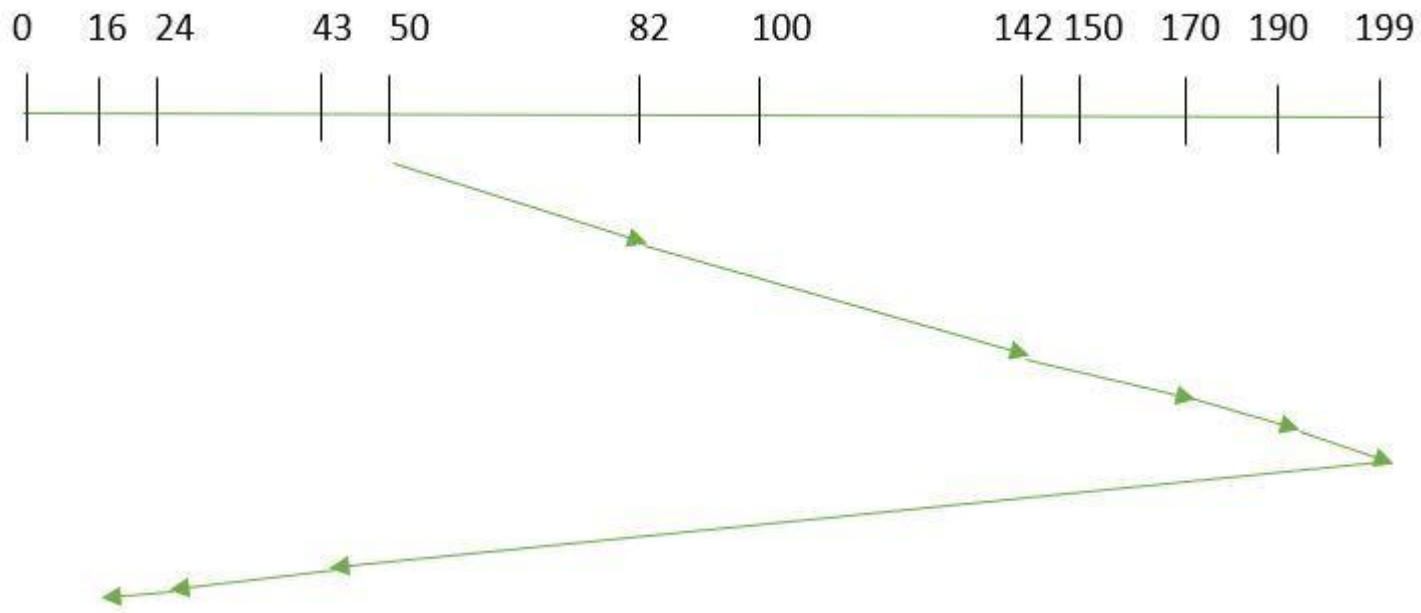


- Suppose the order of request is-
  (82,170,43,140,24,16,190)
  And current position of Read/Write head is: 50

```
total overhead movement (total distance covered by the disk arm) =
(50-43)+(43-24)+(24-16)+(82-16)+(140-82)+(170-140)+(190-170) =208
```

SCAN

- In the SCAN algorithm the disk arm moves in a particular direction and services the requests coming in its path and after reaching the end of the disk, it reverses its direction and again services the request arriving in its path. So, this algorithm works as an elevator and is hence also known as an **elevator algorithm**



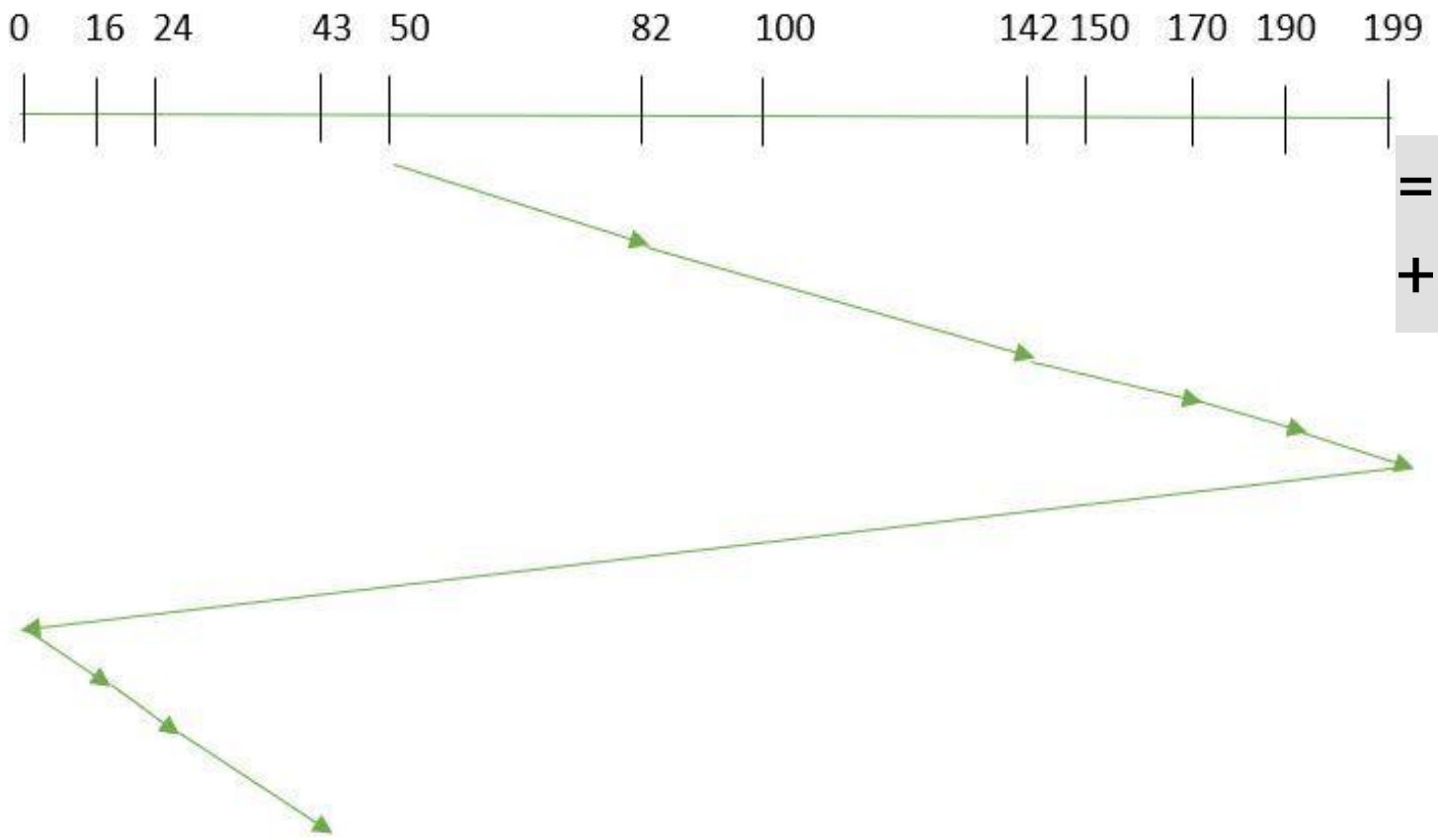
- Suppose the requests to be addressed are-82,170,43,140,24,16,190. And the Read/Write arm is at 50, and it is also given that the disk arm should move **“towards the larger value”.**
- Therefore, the total overhead movement (total distance covered by the disk arm) is calculated as
- `(199-50) + (199-16) = 332`

C-SCAN

- In the SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area.
- These situations are avoided in the *CSCAN* algorithm in which the disk arm instead of reversing its direction goes to the other end of the disk and starts servicing the requests from there. So, the disk arm moves in a circular fashion and this algorithm is also similar to the SCAN algorithm hence it is known as C-SCAN (Circular SCAN)

Suppose the requests to be addressed are-82,170,43,140,24,16,190. And the Read/Write arm is at 50, and it is also given that the disk arm should move **“towards the larger value”.**

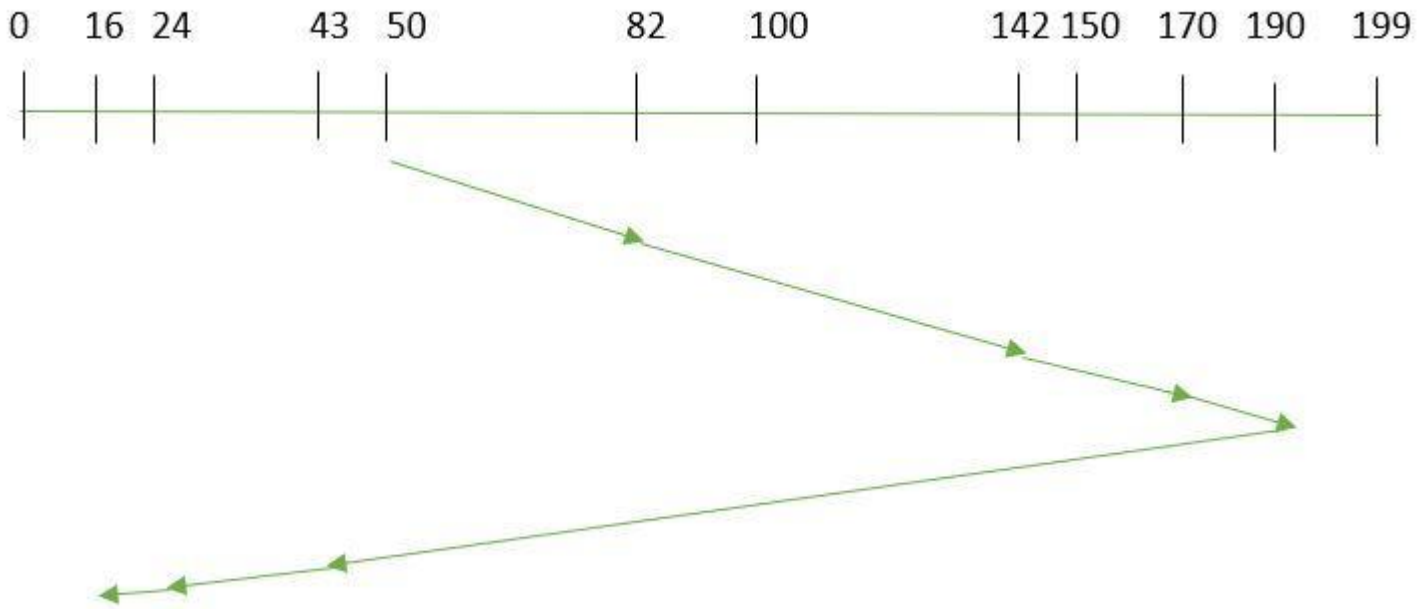
So, the total overhead movement (total distance covered by the disk arm) is calculated as:




```
=(199-50) + (199-0)
+ (43-0) = 391
```

LOOK

- LOOK Algorithm is similar to the SCAN disk scheduling algorithm except for the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced in front of the head and then reverses its direction from there only. Thus it prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.



- Suppose the requests to be addressed are-82,170,43,140,24,16,190. And the Read/Write arm is at 50, and it is also given that the disk arm should move **“towards the larger value”.**
- So, the total overhead movement (total distance covered by the disk arm) is calculated as:
- = (190-50) + (190-16) = 314