

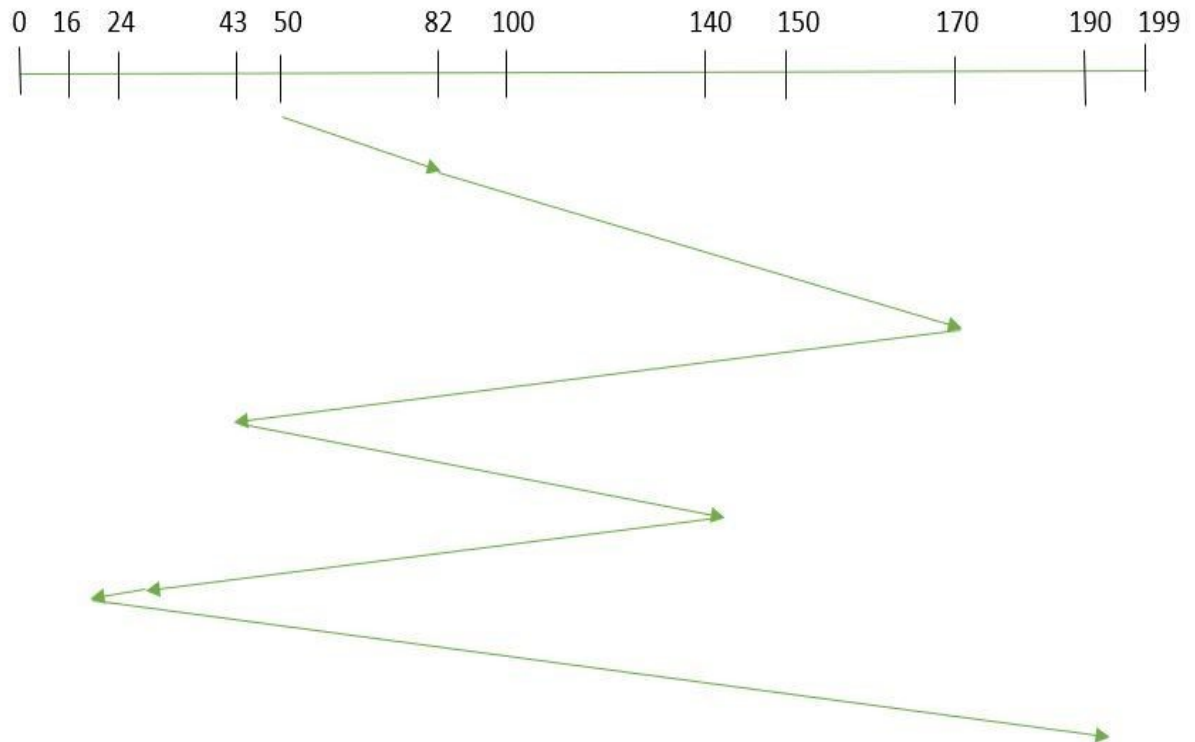
DISK Scheduling Algorithm example:

1. FCFS

Example:

Suppose the order of request is- (82,170,43,140,24,16,190)

And current position of Read/Write head is : 50



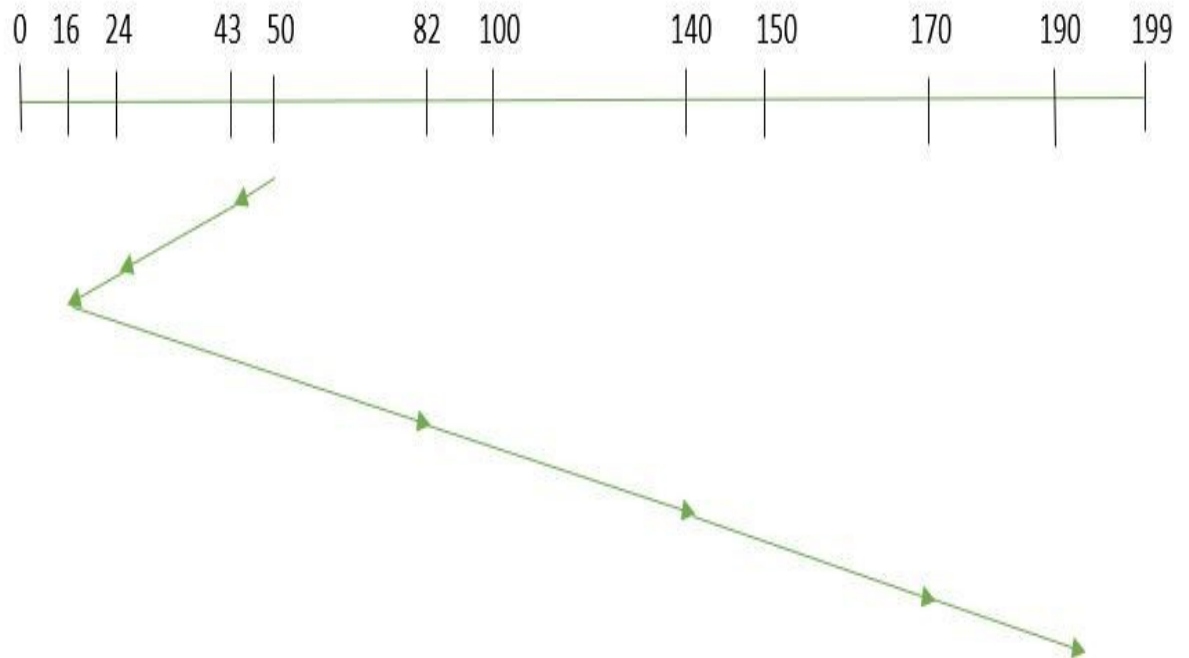
So, total seek time:

$$\begin{aligned} &= (82-50) + (170-82) + (170-43) + (140-43) + (140-24) + (24-16) + (190-16) \\ &= 642 \end{aligned}$$

2. **SSTF**: In SSTF (Shortest Seek Time First), requests having 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.

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 seek time:

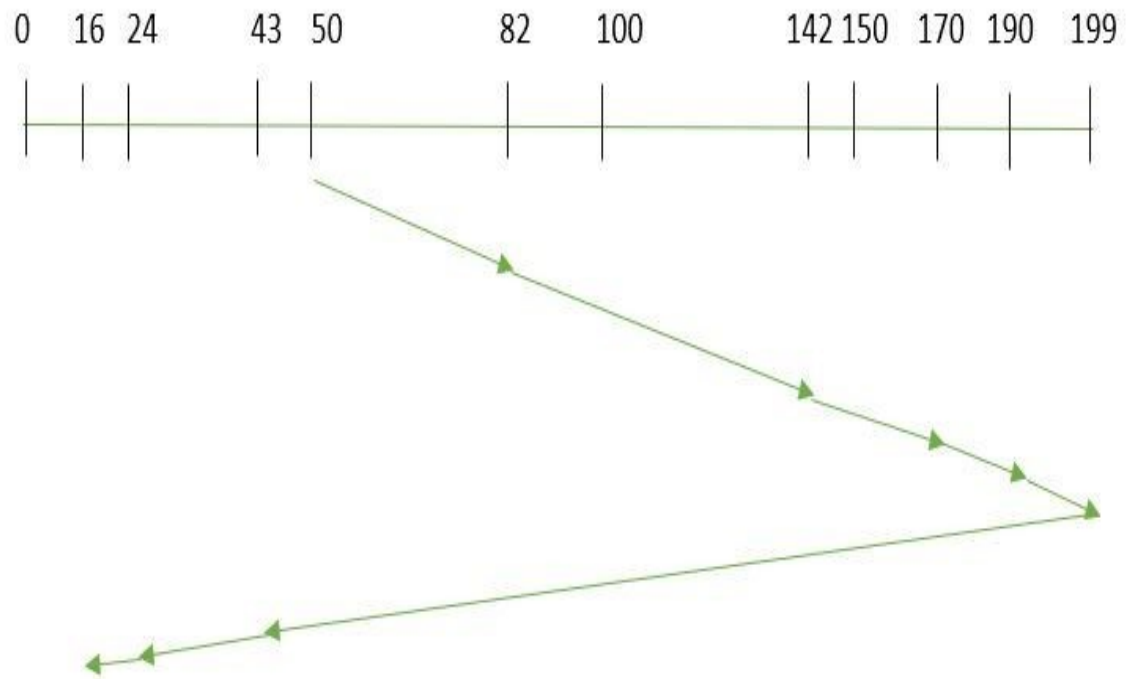
$$\begin{aligned}
 &= (50-43) + (43-24) + (24-16) + (82-16) + (140-82) + (170-140) + (190-170) \\
 &= 208
 \end{aligned}$$

3. **SCAN:** In SCAN algorithm 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. So, this algorithm works as an elevator and hence also known as **elevator algorithm**.

Example:

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”**.

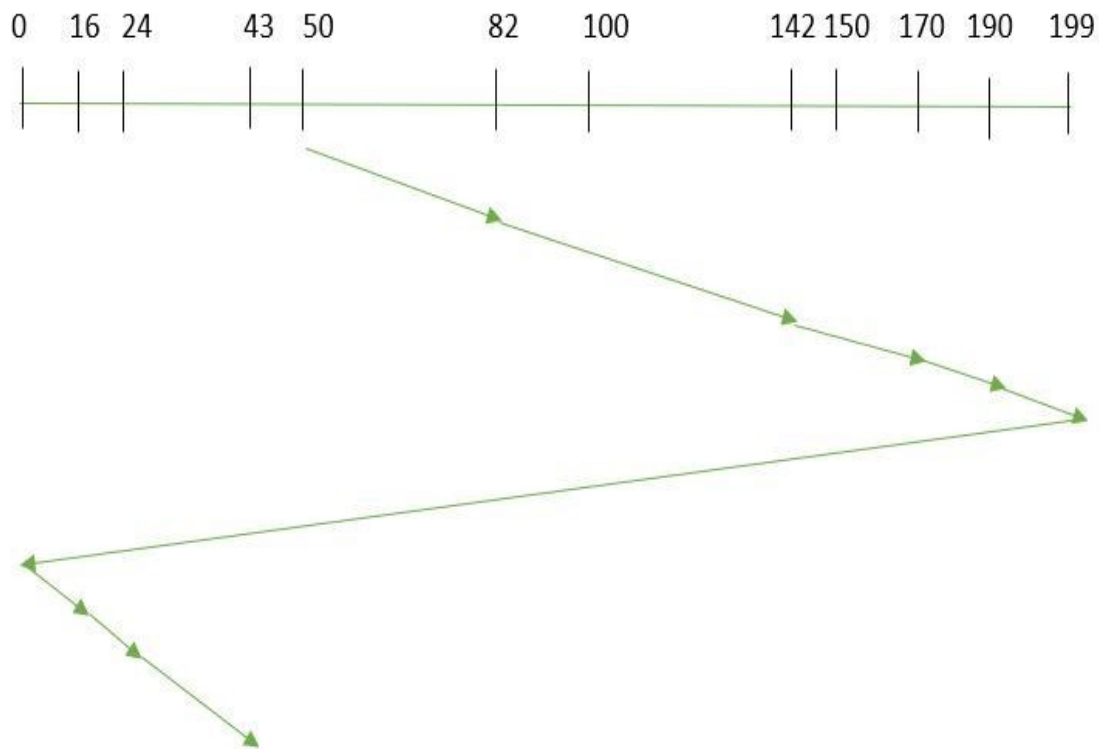


Therefore, the seek time is calculated as:

$$\begin{aligned} &= (199 - 50) + (199 - 16) \\ &= 332 \end{aligned}$$

4. **CSCAN**: **Example**:

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”**.



Seek time is calculated as:

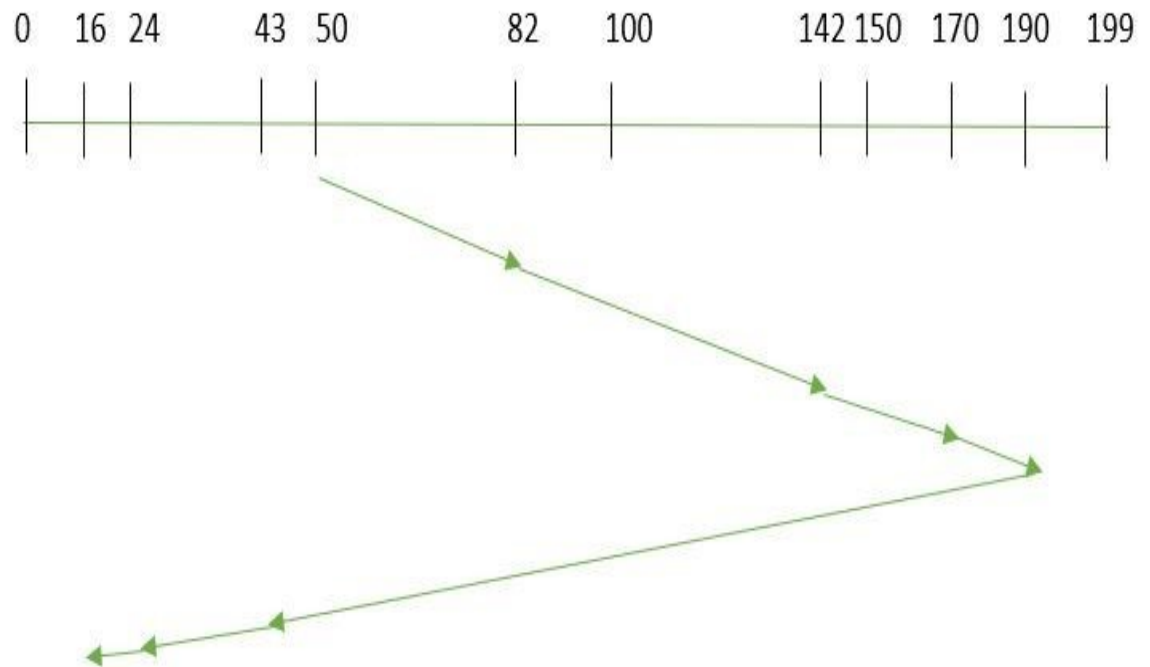
$$= (199-50) + (199-0) + (43-0) \\ = 391$$

5. **LOOK:** It 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.

Example:

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 seek time is calculated as:

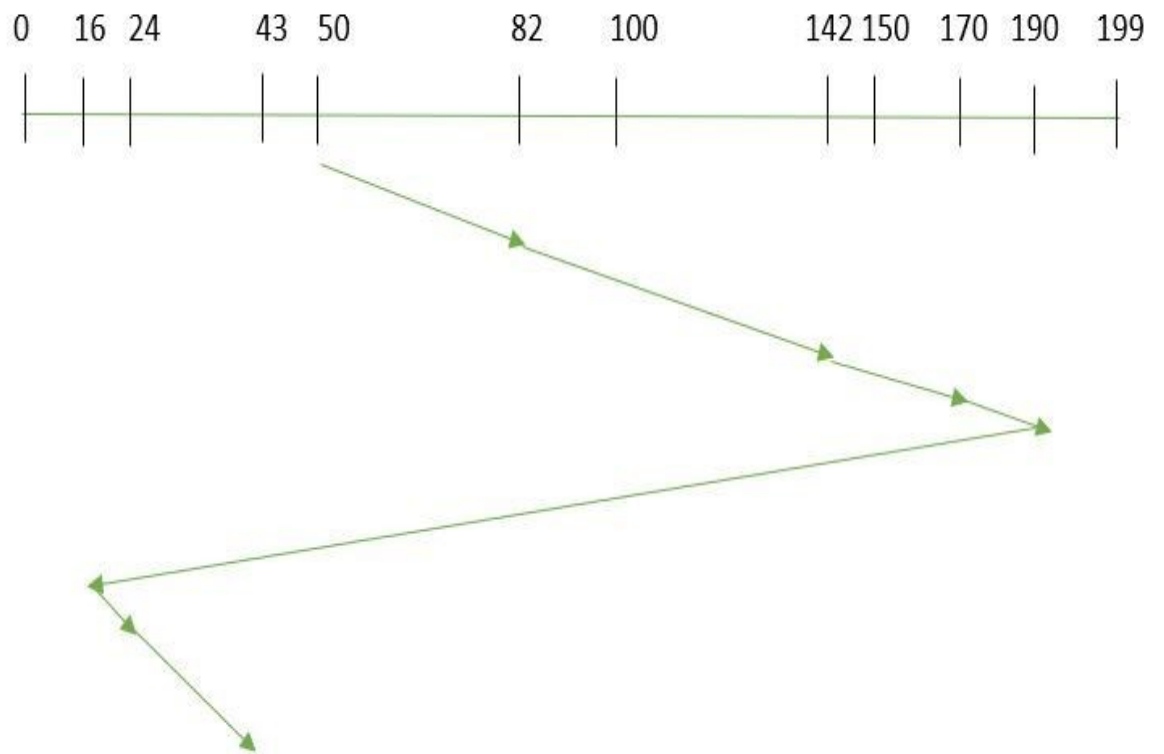
$$\begin{aligned} &= (190 - 50) + (190 - 16) \\ &= 314 \end{aligned}$$

6. **CLOOK:** As LOOK is similar to SCAN algorithm, in similar way, CLOOK is similar to CSCAN disk scheduling algorithm. In CLOOK, the disk arm in spite of going to the end goes only to the last request to be serviced in front of the head and then from there goes to the other end's last request. Thus, it also prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

Example:

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 seek time is calculated as:

$$\begin{aligned} &= (190 - 50) + (190 - 16) + (43 - 16) \\ &= 341 \end{aligned}$$