**CO5**

**STORAGE MANAGEMENT**

**DISK SCHEDULING**

**Disk scheduling**is done by operating systems to schedule I/O requests arriving for accessing the disk. Disk scheduling is also known as I/O Scheduling.

The list of various disks scheduling algorithm is given below.

1)FCFS scheduling algorithm

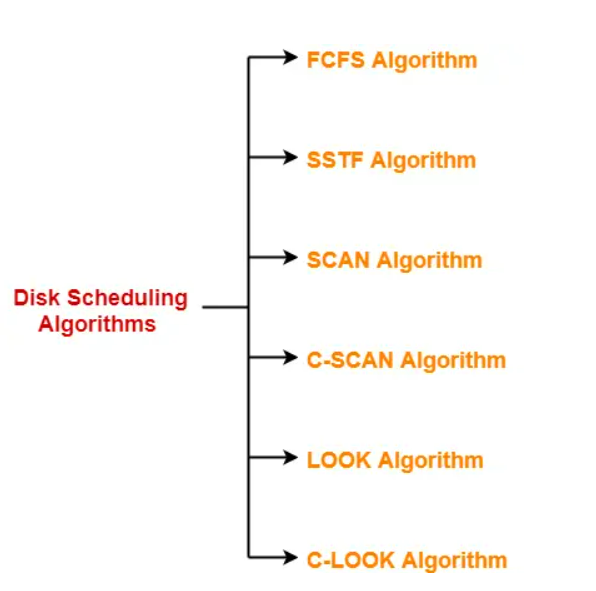
2)SSTF (shortest seek time first) algorithm

3)SCAN scheduling

4)C-SCAN scheduling

5)LOOK Scheduling

6)C-LOOK scheduling



**1.FCFS Scheduling:**

The **FCFS (First Come, First-Served)** scheduling algorithm is the simplest disk scheduling algorithm of all. But it does not provide the fastest service. As the name suggests, this algorithm fulfils requests in the order they arrive in the disk queue.

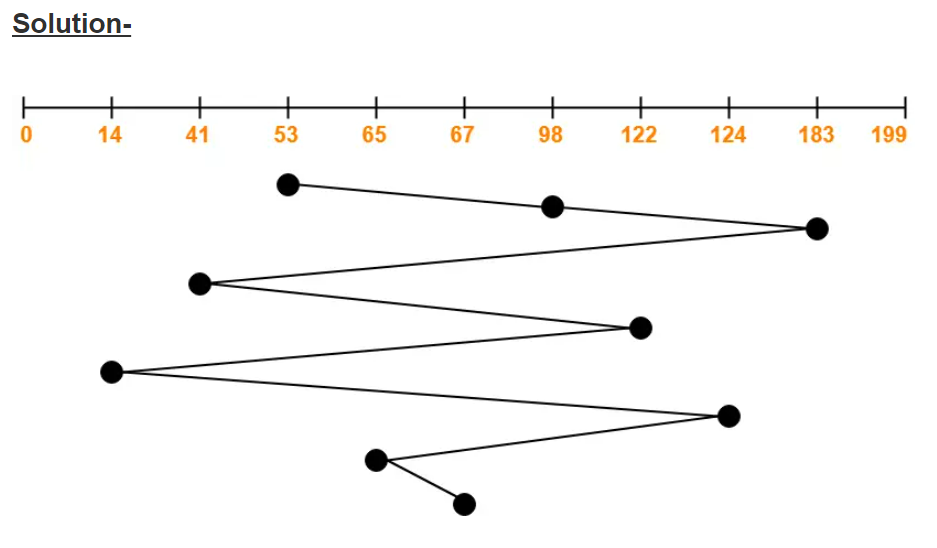
**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 72, 160, 33, 130, 14, 6, 180. The FCFS scheduling algorithm is used. The head is initially at cylinder number 50. The cylinders are numbered from 0 to 199. Find out total head movements (in number of cylinders) incurred while servicing these requests and seek time?

### 

### **Seek Time**

Seek time is the time taken in locating the disk arm to a specified track where the read/write request will be satisfied.

**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The FCFS scheduling algorithm is used. The head is initially at cylinder number 53. The cylinders are numbered from 0 to 199. Find out total head movements (in number of cylinders) incurred while servicing these requests and seek time?



Total head movements incurred while servicing these requests

= (98 – 53) + (183 – 98) + (183 – 41) + (122 – 41) + (122 – 14) + (124 – 14) + (124 – 65) + (67 – 65)

= 45 + 85 + 142 + 81 + 108 + 110 + 59 + 2

= 632

**Advantages-**

* It is simple, easy to understand and implement.
* It does not cause starvation to any request.

**Disadvantages-**

* It results in increased total seek time.
* It is inefficient.

## 2. SSTF Scheduling:

**Advantages-**

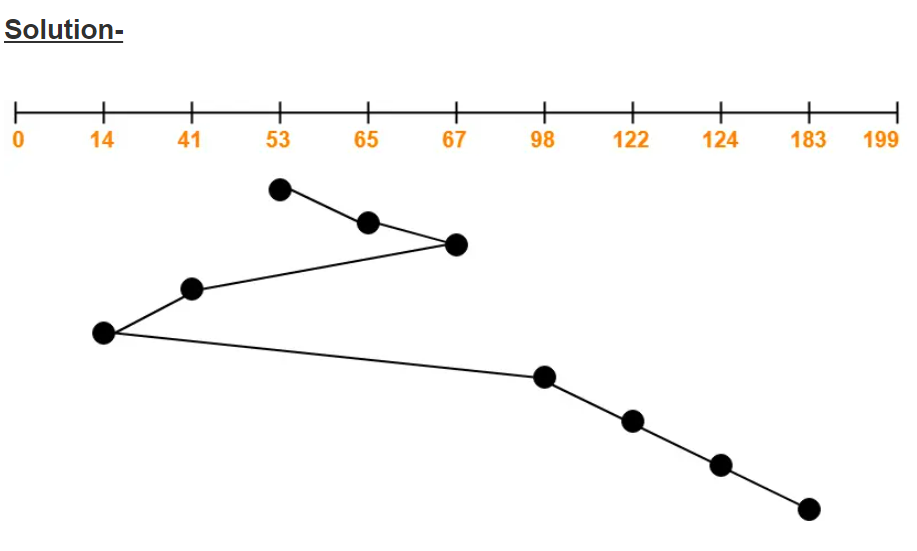
* It reduces the total seek time as compared to [**FCFS**](https://www.gatevidyalay.com/disk-scheduling-disk-scheduling-algorithms/).
* It provides increased throughput.
* It provides less average response time and waiting time.

**Disadvantages-**

* There is an overhead of finding out the closest request.
* The requests which are far from the head might starve for the CPU.
* It provides high variance in response time and waiting time.
* Switching the direction of head frequently slows down the algorithm.

**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 72, 160, 33, 130, 14, 6, 180. The FCFS scheduling algorithm is used. The head is initially at cylinder number 50. The cylinders are numbered from 0 to 199. Find out total head movements (in number of cylinders) incurred while servicing these requests and seek time?

**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The SSTF scheduling algorithm is used. The head is initially at cylinder number 53. The cylinders are numbered from 0 to 199. Find out total head movements (in number of cylinders) incurred while servicing these requests and seek time?



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (67 – 41) + (41 – 14) + (98 – 14) + (122 – 98) + (124 – 122) + (183 – 124)

= 12 + 2 + 26 + 27 + 84 + 24 + 2 + 59

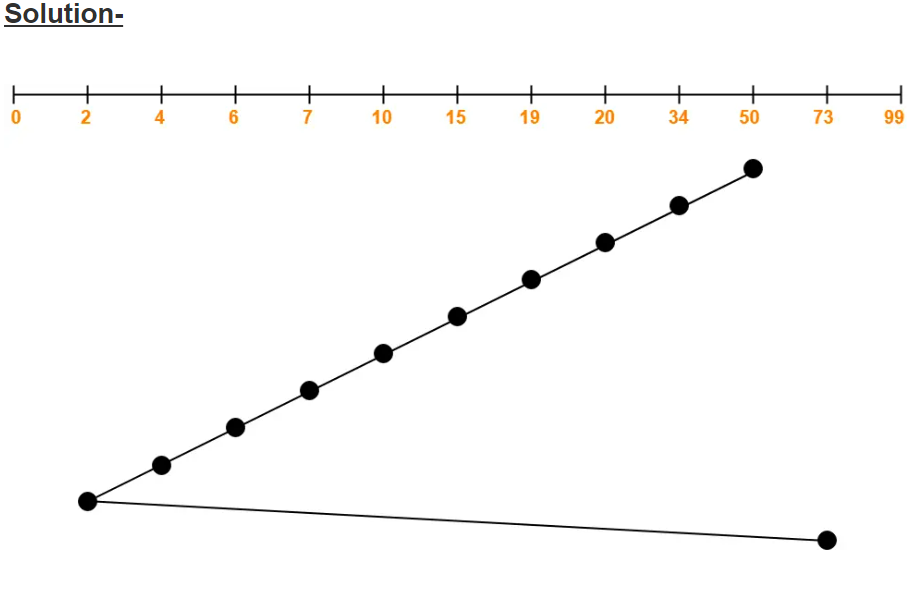
= 236

**Example:**

Consider a disk system with 100 cylinders. The requests to access the cylinders occur in following sequence-

4, 34, 10, 7, 19, 73, 2, 15, 6, 20

Assuming that the head is currently at cylinder 50, what is the time taken to satisfy all requests if it takes 1 ms to move from one cylinder to adjacent one and shortest seek time first policy is used?



Total head movements incurred while servicing these requests

= (50 – 34) + (34 – 20) + (20 – 19) + (19 – 15) + (15 – 10) + (10 – 7) + (7 – 6) + (6 – 4) + (4 – 2) + (73 – 2)

= 16 + 14 + 1 + 4 + 5 + 3 + 1 + 2 + 2 + 71

= 119

Time taken for one head movement = 1 msec. So,

Time taken for 119 head movements

= 119 x 1 msec

= 119 msec

## 3. SCAN Scheduling:

The **SCAN algorithm** is also called the **Elevator algorithm**. As the name suggests, this algorithm scans all the cylinders of the disk back and forth. Head starts from one end of the disk and move towards the other end servicing all the requests in between. After reaching the other end, head reverses its direction and move towards the starting end servicing all the requests in between. The same process repeats.

**Advantages-**

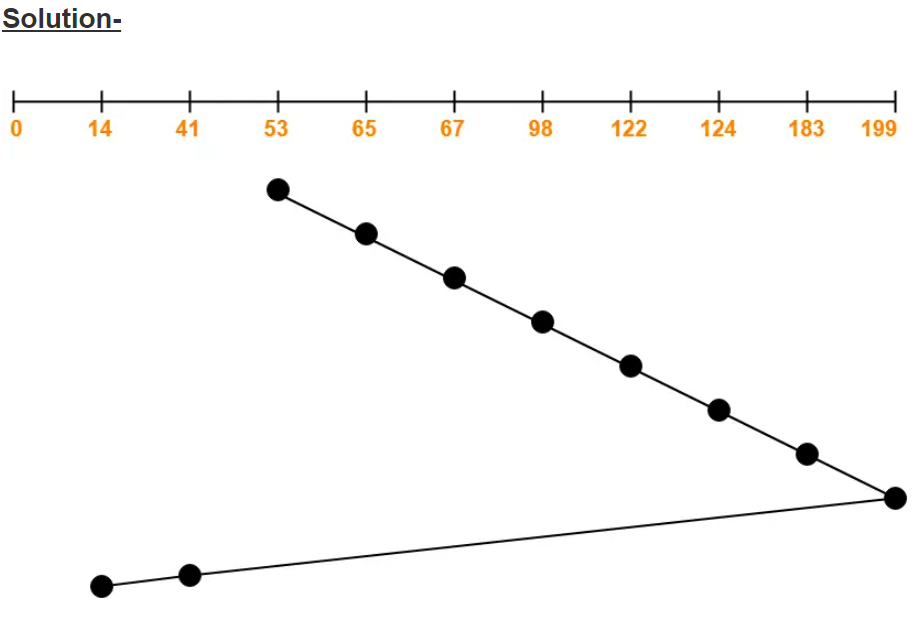
* It is simple, easy to understand and implement.
* It does not lead to starvation.
* It provides low variance in response time and waiting time.

**Disadvantages-**

* It causes long waiting time for the cylinders just visited by the head.
* It causes the head to move till the end of the disk even if there are no requests to be serviced.

**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 72, 160, 33, 130, 14, 6, 180. The FCFS scheduling algorithm is used. The head is initially at cylinder number 50. The cylinders are numbered from 0 to 199. Find out total head movements (in number of cylinders) incurred while servicing these requests and seek time?

**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The SCAN scheduling algorithm is used. The head is initially at cylinder number 53 moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is \_\_\_\_\_\_\_.



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (199 – 183) + (199 – 41) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 16 + 158 + 27

= 331

**Alternatively**,

Total head movements incurred while servicing these requests

= (199 – 53) + (199 – 14)

= 146 + 185

= 331

## 4. C-SCAN Scheduling

The main drawback of SCAN scheduling is the waiting time is not uniform. Some requests are waiting much time and some requests are serviced immediately. We can overcome this drawback with **C-SCAN (Circular Scan)** scheduling algorithm. This algorithm is designed to provide a more uniform wait time. In this algorithm, the head moves from one end to another end of the disk and services the request along the way. When the head reaches the other end, it immediately returns to the beginning of the disk without servicing any requests on the return trip. Same process repeats.

**Advantages-**

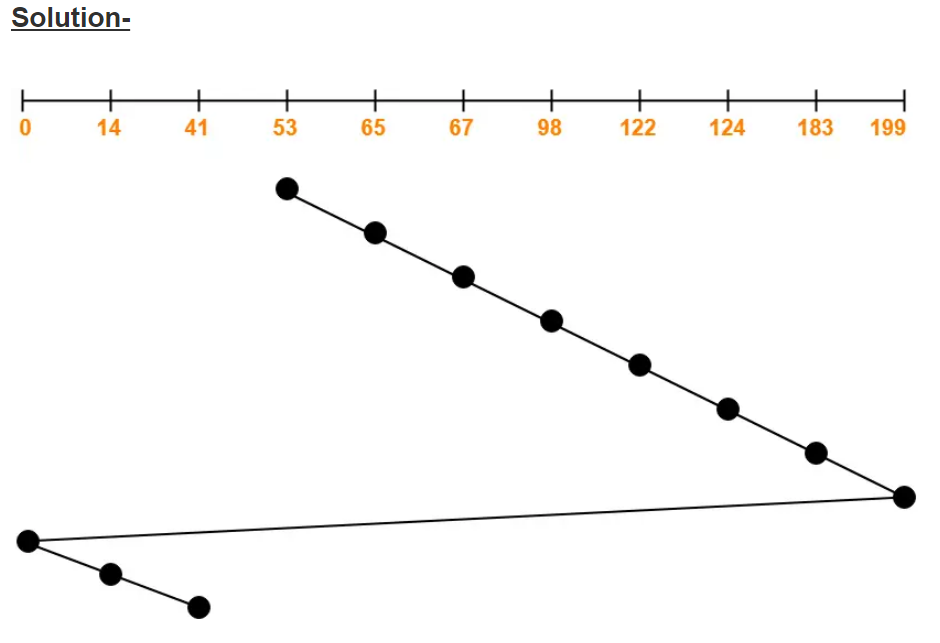
* The waiting time for the cylinders just visited by the head is reduced as compared to the SCAN Algorithm.
* It provides uniform waiting time.
* It provides better response time.

**Disadvantages-**

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

**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 72, 160, 33, 130, 14, 6, 180. The FCFS scheduling algorithm is used. The head is initially at cylinder number 50. The cylinders are numbered from 0 to 199. Find out total head movements (in number of cylinders) incurred while servicing these requests and seek time?

**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The SCAN scheduling algorithm is used. The head is initially at cylinder number 53 moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is \_\_\_\_\_\_\_.



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (199 – 183) + (199 – 0) + (14 – 0) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 16 + 199 + 14 + 27

= 386

**Alternatively**,

Total head movements incurred while servicing these requests

= (199 – 53) + (199 – 0) + (41 – 0)

= 146 + 199 + 41

= 386

## 5. Look Scheduling

In the SCAN, and C-SCAN algorithms the disk arm moves across the full width of the disk. But in this algorithm, the arm goes only as far as the final request in each direction. Then it reverses the direction immediately without reaching the end of the disk.

**Advantages-**

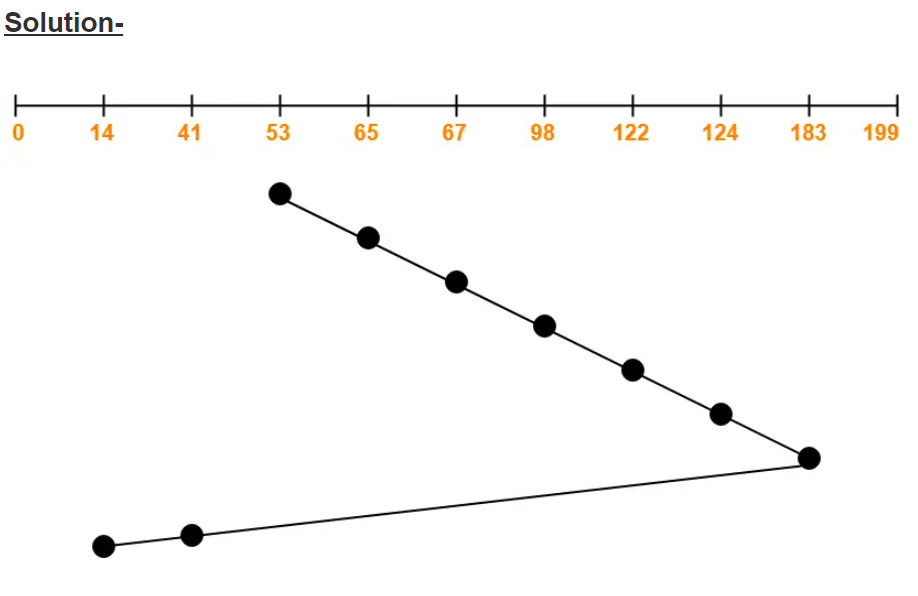
* It does not causes the head to move till the ends of the disk when there are no requests to be serviced.
* It provides better performance as compared to SCAN Algorithm.
* It does not lead to starvation.
* It provides low variance in response time and waiting time.

**Disadvantages-**

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

**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 72, 160, 33, 130, 14, 6, 180. The FCFS scheduling algorithm is used. The head is initially at cylinder number 50. The cylinders are numbered from 0 to 199. Find out total head movements (in number of cylinders) incurred while servicing these requests and seek time?

**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The SCAN scheduling algorithm is used. The head is initially at cylinder number 53 moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is \_\_\_\_\_\_\_.

  
Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (183 – 41) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 142 + 27

= 299

**Alternatively**,

Total head movements incurred while servicing these requests

= (183 – 53) + (183 – 14)

= 130 + 169

= 299

## 6. C- Look Scheduling:

Circular-LOOK Algorithm is an improved version of the [LOOK Algorithm](https://www.gatevidyalay.com/look-algorithm-disk-scheduling-algorithms/). 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. The same process repeats.

**Advantages-**

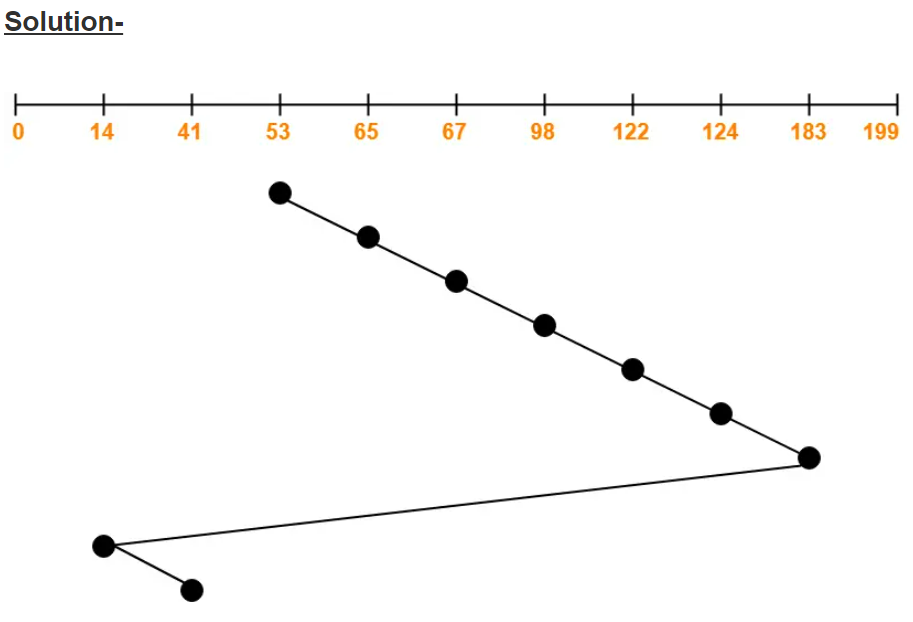
* It does not causes the head to move till the ends of the disk when there are no requests to be serviced.
* It reduces the waiting time for the cylinders just visited by the head.
* It provides better performance as compared to LOOK Algorithm.
* It does not lead to starvation.
* It provides low variance in response time and waiting time.

**Disadvantages-**

* There is an overhead of finding the end requests.

**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 72, 160, 33, 130, 14, 6, 180. The FCFS scheduling algorithm is used. The head is initially at cylinder number 50. The cylinders are numbered from 0 to 199. Find out total head movements (in number of cylinders) incurred while servicing these requests and seek time?

**Example:** Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The SCAN scheduling algorithm is used. The head is initially at cylinder number 53 moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is \_\_\_\_\_\_\_.



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (183 – 14) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 169 + 27

= 326

**Alternatively**,

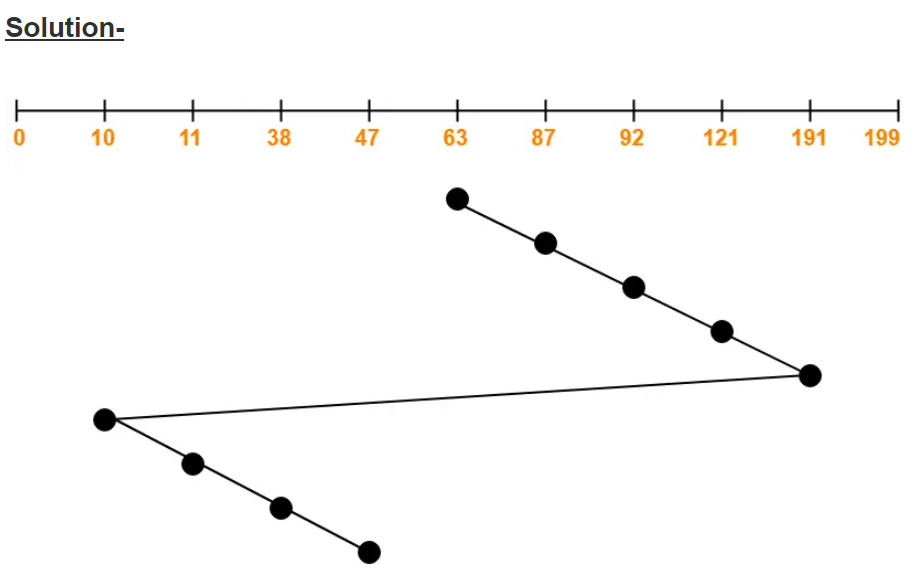
Total head movements incurred while servicing these requests

= (183 – 53) + (183 – 14) + (41 – 14)

= 130 + 169 + 27

= 326

Example: Consider a disk queue with requests for I/O to blocks on cylinders 47, 38, 121, 191, 87, 11, 92, 10. The C-LOOK scheduling algorithm is used. The head is initially at cylinder number 63 moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is \_\_\_\_\_\_\_.



Total head movements incurred while servicing these requests

= (87 – 63) + (92 – 87) + (121 – 92) + (191 – 121) + (191 – 10) + (11 – 10) + (38 – 11) + (47 – 38)

= 24 + 5 + 29 + 70 + 181 + 1 + 27 + 9

= 346

**Alternatively**,

Total head movements incurred while servicing these requests

= (191 – 63) + (191 – 10) + (47 – 10)

= 128 + 181 + 37

= 346