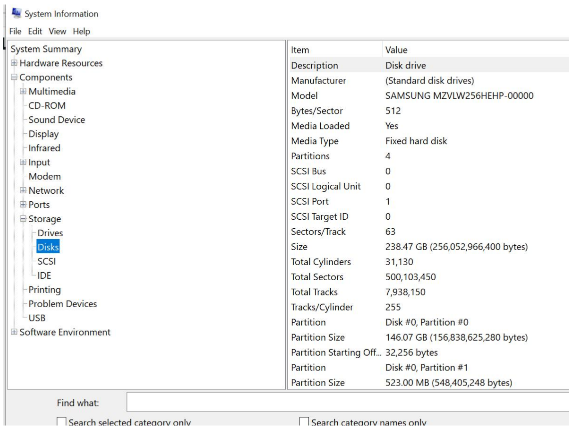
About your Hard Drive

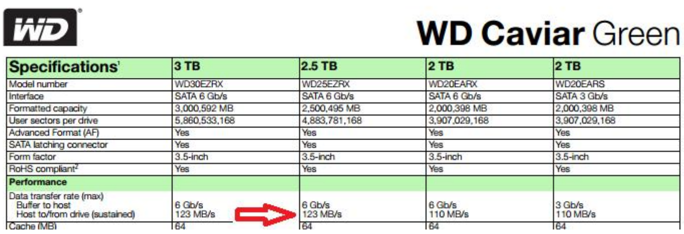
- Windows
Click on **Start** and type in **msinfo32** and press **Enter**.



Week 10

Tutorial

Example 2 - Transfer Time



WD Caviar Green

Specifications	3 TB	2.5 TB	2 TB	2 TB
Model number	WD30EZRX	WD25EZRX	WD20EARX	WD20EARS
Interface	SATA 6 Gb/s	SATA 6 Gb/s	SATA 6 Gb/s	SATA 3 Gb/s
Formatted capacity	3,000,592 MB	2,500,495 MB	2,000,398 MB	2,000,398 MB
User sectors per drive	5,860,533,168	4,883,781,168	3,907,029,168	3,907,029,168
Advanced Format (AF)	Yes	Yes	Yes	Yes
SATA latching connector	Yes	Yes	Yes	Yes
Form factor	3.5-inch	3.5-inch	3.5-inch	3.5-inch
RoHS compliant	Yes	Yes	Yes	Yes
Performance				
Data transfer rate (max) Buffer to host Host to/from drive (sustained)	6 Gb/s 123 MB/s	6 Gb/s 123 MB/s	6 Gb/s 110 MB/s	3 Gb/s 110 MB/s
Cache (MB)	64	64	64	64

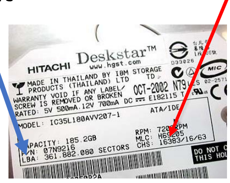
the transfer time will also depend on the block size.
If we assume a 4K block size and a transfer rate of 123 MBs we can use the following equation

Transfer time = block size / Data (sustained) transfer rate = 4 / 123 = 3.175 $*10^{-5}$ s = 0.0317 ms

Example 1

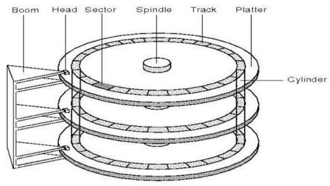
LBA of your Hard Drive

CHS of your Hard Drive



CHS = C x H x S

CHS (Cylinder/Head/Sector) is an earlier form of hard disk addressing.



C: cylinder, the valid range is between 0 and 1023 cylinders.
H: Head, the valid range is between 0 and 254 heads (formerly 0-15).
S: Sector, the valid range is between 1 and 63 sectors.

The capacity of hard disk **CHS = 16383/16/63**

- 16383(cylinders) x 16(heads) x 63(sectors) x 128(bytes) = **2,113,800,192** bytes (2 GB).
- physical sector sizes of 128, 256, 512, 1024, 4096 bytes

Example 3

Consider a disk queue with requests for I/O to blocks on cylinders

98, 183, 37, 122, 14, 124, 65, 67.

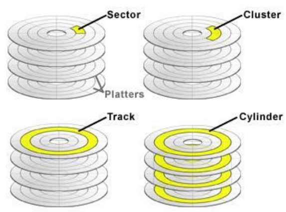
The **SCAN scheduling algorithm** is used. The head is initially at cylinder number 53. The cylinders are numbered from 0 to 199.

Calculate and show with diagram the disk head movement using SCAN-scheduling algorithm.

Hint

It services the requests in the direction of its head movement and continues until it reaches the end of the disk or the last cylinder. After this, it reverses the direction and services the requests in its path.
- the preferred direction of the head movement has to be found first and all the requests in that direction will be serviced.

LBA (Logical block addressing) is what is used on (probably all) modern drives. The drive is logically divided into blocks (512 byte or 4k byte sectors mostly), the first block is block 0, the next one block 1, etc..



LBA = (c · Nheads + h) · Nsectors + (s – 1)

LBA - Logical Block Address
Nheads - Number of heads on a disk heads-per-disk
Nsectors - Numbers of sectors on a track sectors-per-track
c,h,s - is the cylinder, head, sector numbers 24-bits total (10+8+6)

Geometry: (this is defined by the manufacturer)
- Cylinders - 1020
- Heads- 16
- Sectors - 63

In order to convert one style of address to the other, you need to specify the drive geometry:

. number of cylinders
. number of heads (per cylinder)
. number of sectors per track

Example 5

Consider a disk queue with I/O requests on the following cylinders in their arriving order:

54, 97, 73, 128, 15, 44, 110, 34, 45

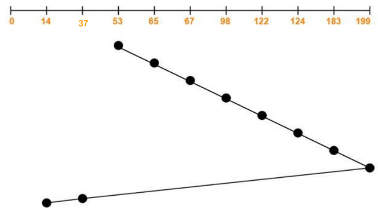
The disk head is assumed to be at Cylinder **23**.

Calculate and show with diagram the total disk head movement using **SSTF-scheduling algorithm**.

Hint

- it calculates the seek time of all the requests in the queue,
- It selects the appropriate request, and
- It schedules it for the execution.

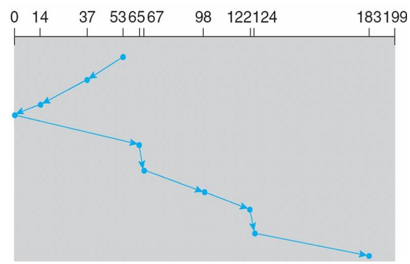
Solution



towards larger cylinder numbers on its servicing pass

53-65-67-98-122-124-183-199-37-14

the total head movement = 331

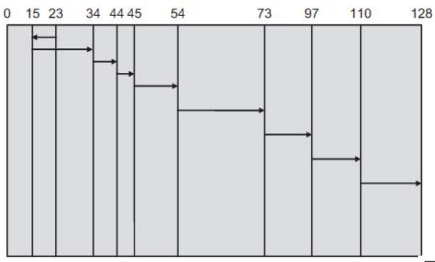


towards smaller cylinder numbers on its servicing pass

53-37-14-0-65-67-98-122-124-183

the total head movement = 236

Solution



Head movement	Total head movement for the request
23–15	8
15–34	19
34–44	10
44–45	1
45–54	9
54–73	19
73–97	24
97–110	13
110–128	18
Total head movement = 121	

Example 4

Consider a disk queue with I/O requests on the following cylinders in their arriving order:

54, 97, 73, 128, 15, 44, 110, 34, 45

The disk head is assumed to be at Cylinder **23**.

Calculate and show with diagram the total disk head movement using **FCFS-scheduling algorithm**.

Example 6

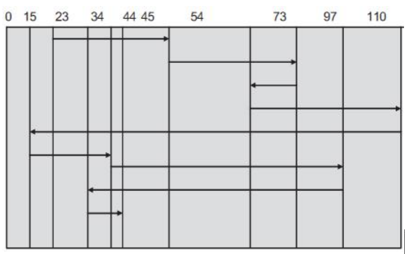
Consider a disk queue with I/O requests on the following cylinders in their arriving order:

6, 10, 12, 54, 97, 73, 128, 15, 44, 110, 34, 45

The disk head is assumed to be at Cylinder **23** and **moving in the direction of decreasing number of cylinders**. The disk consists of total 150 cylinders.

Calculate and show with diagram the disk head movement using **C-SCAN scheduling algorithm**.

Solution

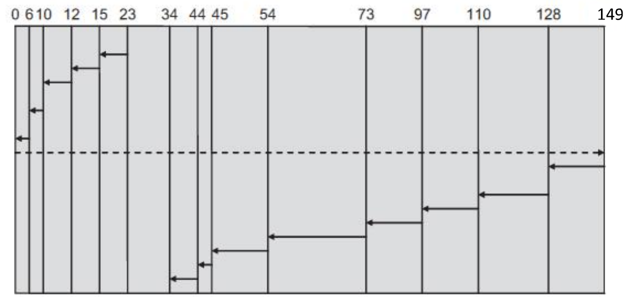


54, 97, 73, 128, 15, 44, 110, 34, 45

I/O request for cylinder	Head movement	Total head movement for the request
54	23–54	31
97	54–97	43
73	97–73	24
128	73–128	55
15	128–15	113
44	15–44	29
110	44–110	66
34	110–34	76
45	34–45	11
Total head movement = 448		

Solution

the head returns to the beginning of the disk/the end of the disk without serving any request in its path



The total head movement = 138
The head jumps 150 cylinders

PRACTICE PROBLEM BASED ON FCFS DISK SCHEDULING ALGORITHM

FCFS Disk Scheduling Algorithm

- As the name suggests, this algorithm **entertains requests in the order they arrive in the disk queue.**
- It is the simplest disk scheduling algorithm.

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.

Problem

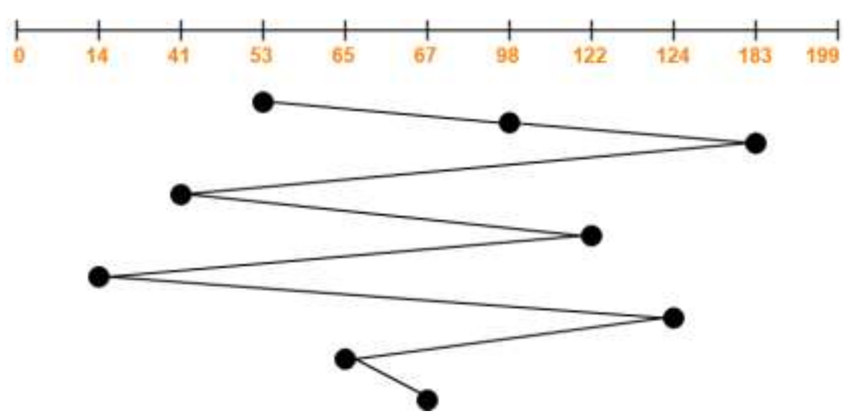
Consider a disk queue with requests for I/O to blocks on cylinders

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.

Write the sequence in which requested tracks are serviced using FCFS and calculate the total head movement (in number of cylinders) incurred while servicing these requests.

Solution



98-183-41-122-14-124-65-67

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

PRACTICE PROBLEM BASED ON SCAN DISK SCHEDULING ALGORITHM

SCAN Disk Scheduling 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.

NOTE

- SCAN Algorithm is also called as *Elevator Algorithm*.
- This is because its working resembles the working of an elevator.

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.

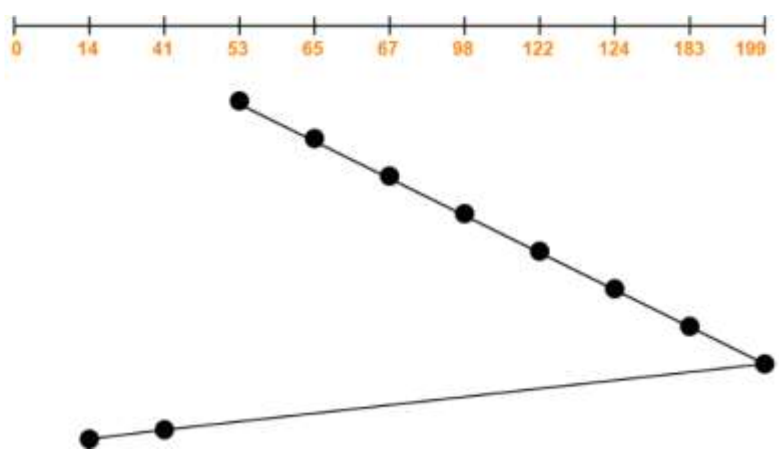
Problem

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.
Write the sequence in which requested tracks are serviced using SCAN and calculate the total head movement (in number of cylinders) incurred while servicing these requests.

Solution



53-65-67-98-122-124-183-199-41-14

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,

53-65-67-98-122-124-183-199-41-14

Total head movements incurred while servicing these requests
= (199 – 53) + (199 – 14)
= 146 + 185
= 331

PRACTICE PROBLEM BASED ON C-SCAN DISK SCHEDULING ALGORITHM

C-SCAN Disk Scheduling Algorithm

- Circular-SCAN Algorithm is an improved version of the SCAN Algorithm.
- Head starts from one end of the disk and move towards the other end servicing all the requests in between, then it jumps to the last cylinder of the opposite direction without servicing any request then it turns back and start moving in that direction servicing the remaining requests.
- After reaching the other end, head reverses its direction and then resets at the outer track to begin again.
- It then returns to the starting end without servicing any request in between.
- The 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.

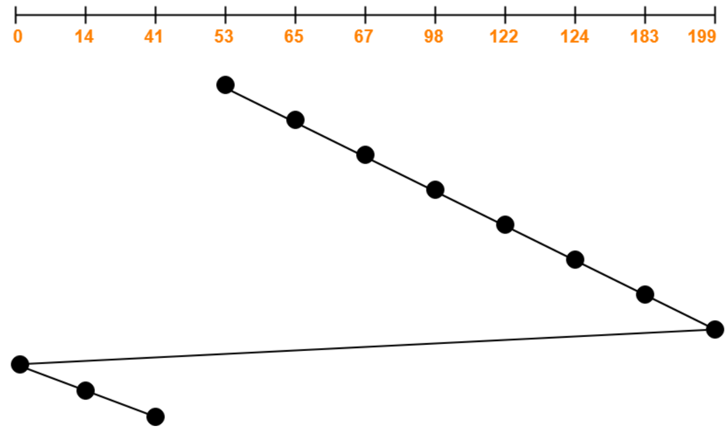
Problem

Consider a disk queue with requests for I/O to blocks on cylinders

98, 183, 41, 122, 14, 124, 65, 67.

The C-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 disk has 200 cylinders).
Write the sequence in which requested tracks are serviced using C-SCAN and calculate the total head movement (in number of cylinders) incurred while servicing these requests.

Solution



53-65-67-98-122-124-183-199-0-14-41
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 + 200 + 14 + 27
= 187

The head jumps from 199-0 = 200

Alternatively,
Total head movements incurred while servicing these requests
= (199 – 53) + (200) + (41 – 0)
= 146 + 200 + 41
= 187

PRACTICE PROBLEM BASED ON LOOK DISK SCHEDULING ALGORITHM

LOOK Disk Scheduling Algorithm

- LOOK Algorithm is an improved version of the SCAN 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 servicing all the requests in between.
- The same process repeats.

NOTE

The main difference between SCAN Algorithm and LOOK Algorithm is-

- SCAN Algorithm scans all the cylinders of the disk starting from one end to the other end even if there are no requests at the ends.
- LOOK Algorithm scans all the cylinders of the disk starting from the first request at one end to the last request at the other end.

Advantages

- It does not cause 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.

Problem

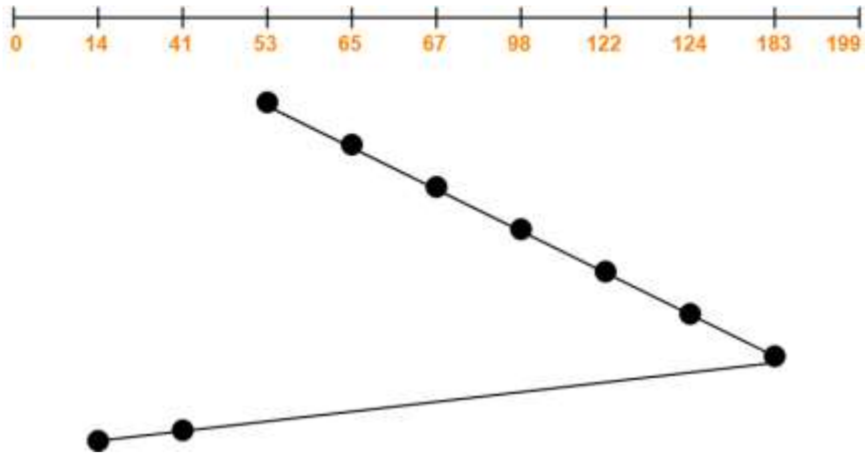
Consider a disk queue with requests for I/O to blocks on cylinders

98, 183, 41, 122, 14, 124, 65, 67.

The LOOK 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.

Write the sequence in which requested tracks are serviced using LOOK and calculate the total head movement (in number of cylinders) incurred while servicing these requests.

Solution



53-65-67-98-122-124-183-41-14

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,

53-65-67-98-122-124-183-41-14

Total head movements incurred while servicing these requests
= (183 – 53) + (183 – 14)
= 130 + 169
= 299

PRACTICE PROBLEMS BASED ON C-LOOK DISK SCHEDULING ALGORITHM

C-LOOK Disk Scheduling Algorithm

- Circular-LOOK Algorithm is an improved version of the LOOK 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.**
- **The same process repeats**.

Advantages

- It does not cause 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.

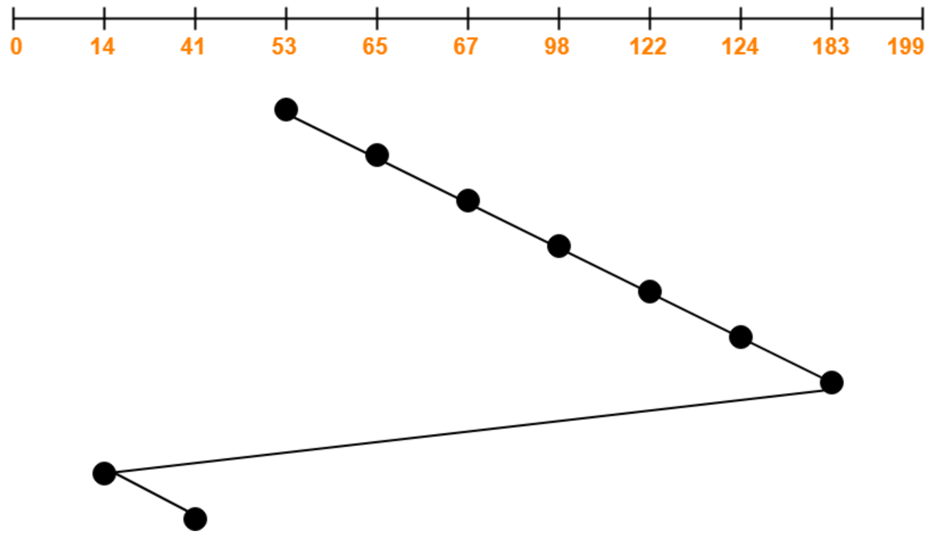
Problem-01:

Consider a disk queue with requests for I/O to blocks on cylinders

98, 183, 41, 122, 14, 124, 65, 67.

The C-LOOK 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.
Write the sequence in which requested tracks are serviced using C-LOOK and calculate the total head movement (in number of cylinders) incurred while servicing these requests.

Solution



53-65-67-98-122-124-183-14-41

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
= 157

The head jumps from 183-14 = 169

Alternatively,
Total head movements incurred while servicing these requests
= (183 – 53) + (183 – 14) + (41 – 14)
= 130 + 169 + 27
= 157

Problem-02:

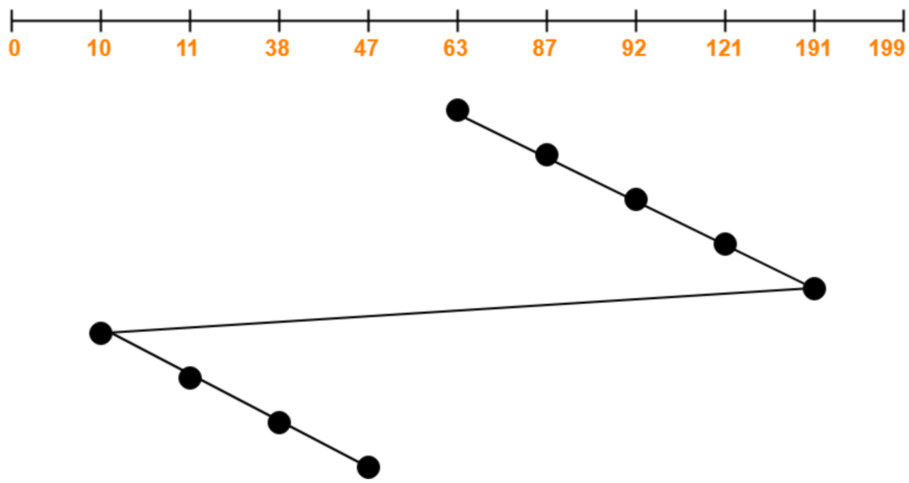
Consider a disk queue with requests for I/O to blocks on cylinders

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.

Write the sequence in which requested tracks are serviced using C-LOOK and calculate the total head movement (in number of cylinders) incurred while servicing these requests.

Solution



63-87-92-121-191-10-11-38-47

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
= 165

The head jumps from 191-10 = 181

Alternatively,
Total head movements incurred while servicing these requests
= (191 – 63) + (191 – 10) + (47 – 10)
= 128 + 181 + 37
= 165

PRACTICE PROBLEMS BASED ON SSTF DISK SCHEDULING ALGORITHM

SSTF Disk Scheduling Algorithm

- SSTF stands for **Shortest Seek Time First**.
- This algorithm services that request next which **requires least number of head movements from its current position regardless of the direction.**
- It breaks the tie in the direction of head movement.

Advantages:

- It reduces the total seek time as compared to FCFS.
- It provides increased throughput.
- It provides less average response time and waiting time.

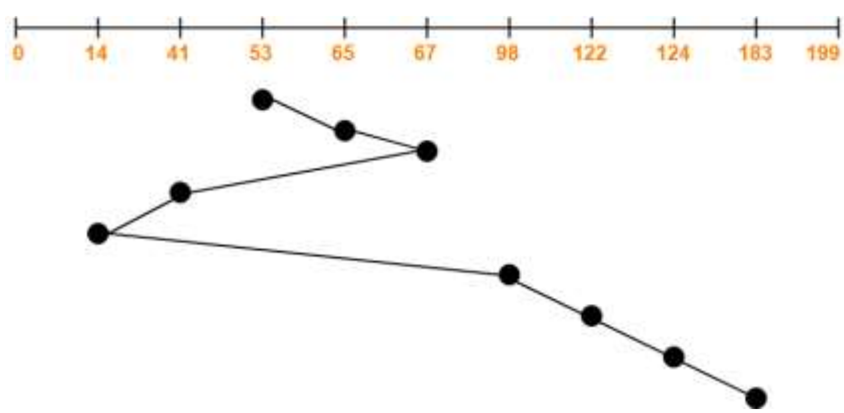
Disadvantages:

- 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.

Problem-01:

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** moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199.
Write the sequence in which requested tracks are serviced using SSTF and calculate the total head movement (in number of cylinders) incurred while servicing these requests.

Solution



53-65-67-41-14-98-122-124-183

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

Problem-02:

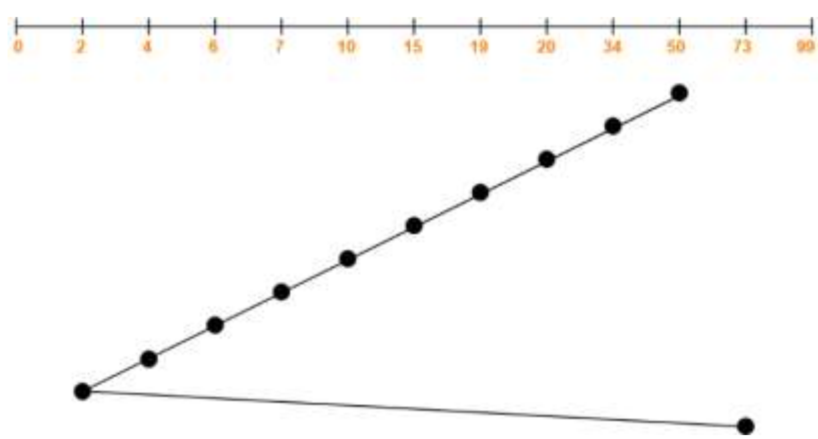
Consider a disk system with 100 cylinders. The requests to access the cylinders occur in following sequence:

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?

A. 95 ms
B. 119 ms
C. 233 ms
D. 276 ms

Solution



50-34-20-19-15-10-7-6-4-2-73

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.
Time taken for 119 head movements
= 119 x 1 msec
= 119 msec

Option (B) is correct.