Chapter: Secondary-Storage Structures

By: Navjot Kaur

Secondary Storage Structure

- Disk is secondary storage that is used to store data
- The data is provided to user programs by means of I/O requests.

Disk Structure

- **Disk is divided into tracks cylinder and sectors.**
- Data is stored on series of magnetic disks called Platters, connected by spindle.
- Surface of platter is divided into circular rings called Tracks
- **Tracks are divided into Sectors.**
- **Sectors are of fixed sized.**
- **Tracks are one arm position form a cylinder.**

Secondary Storage Structure

- **Cylinder** contains one track per platter surface.
- **Sector** stores 512 or 1024 bytes of user data.
- **Sector is identified by:** Cylinder no., track no. with in cylinder and position no. In track.

Cylinder is numbered starting with zero , from outer to inner most cylinder

Tracks are numbered starting with zero from top track to bottom

Sector start with zero ant iclock wise starting from reference position.

Secondary Storage Structure

- Storage Devices can be categorized in to 3 types:
- 1. **Random Access Memory**
- 2. **Serial Storage Device**
  - The data is written on tape.
  - Data is read in a serial / sequential manner
  - **Exp: Magnetic Tape**
- 3. **Direct Access Storage**
  - Exp: Magnetic Disk
  - Floppy disk
  - CD etc.

Secondary Storage Structure

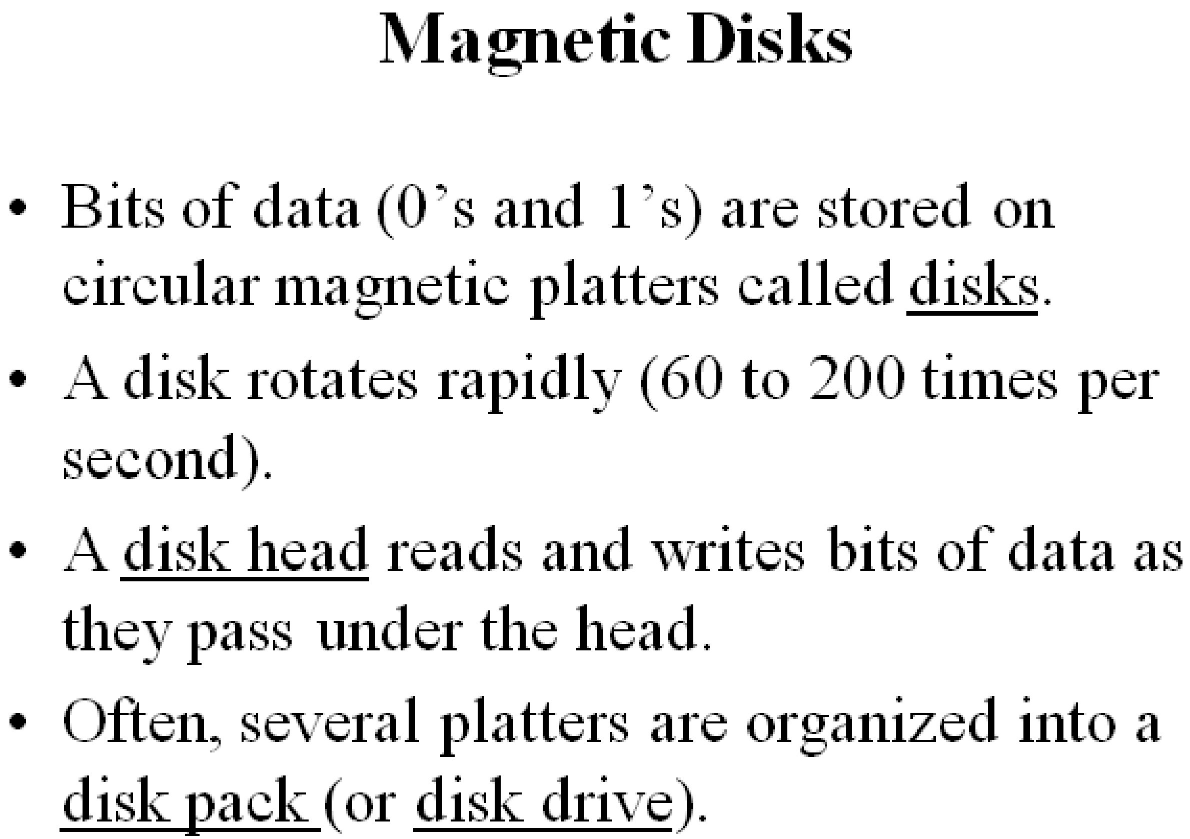
- Some devices are:

Character Devices	Block Devices
Devices in which data is accessed character by character or byte by byte. If we want to write data , it can be written in form of character or byte. Example: Printer	Entire block of data is accessed or written. Example: Magnetic Disk Floppy or CD

Magnetic Tape

- Data is written on tape
- Entire tape is divided into 9 tracks for storing the data
- First 8 tracks will contain 8 bits and 9th track is used to contain parity bit.
- Entire tape pass through read write head
- A Read/Write head is an Array of 9 read/write heads
- Data is stored in the form of records
- Multiple bytes form a record.

Magnetic Disk

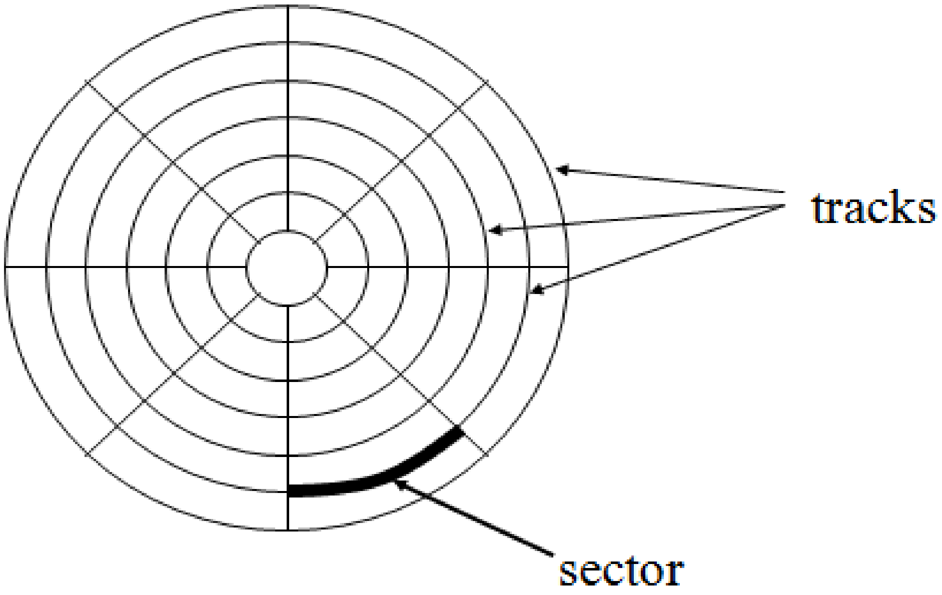


Magnetic Disks

- Bits of data (0's and 1's) are stored on circular magnetic platters called disks.
- A disk rotates rapidly (60 to 200 times per second).
- A disk head reads and writes bits of data as they pass under the head.
- Often, several platters are organized into a disk pack (or disk drive).

Disk Surface

Looking at a surface

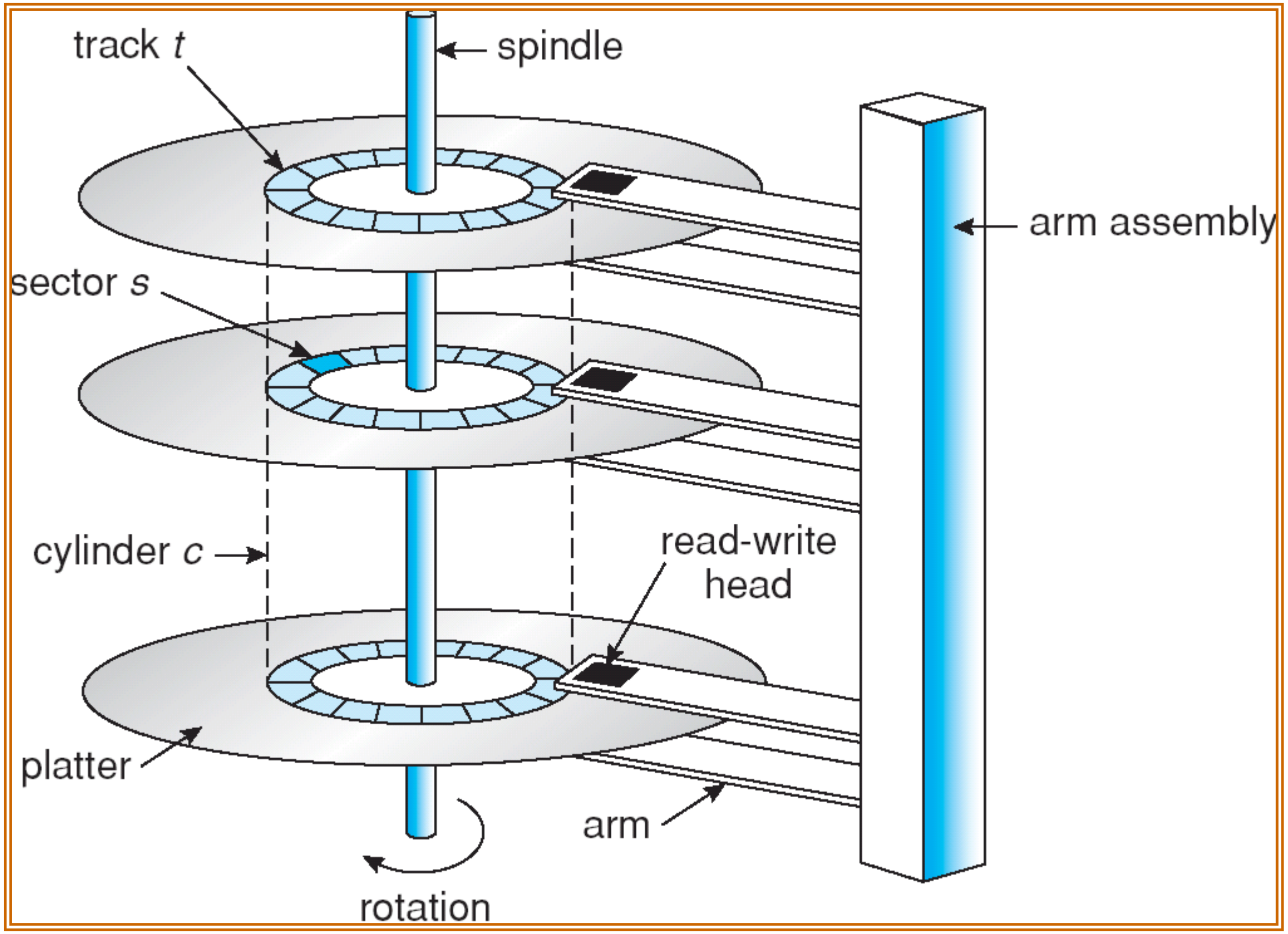


Surface of disk showing tracks and sectors

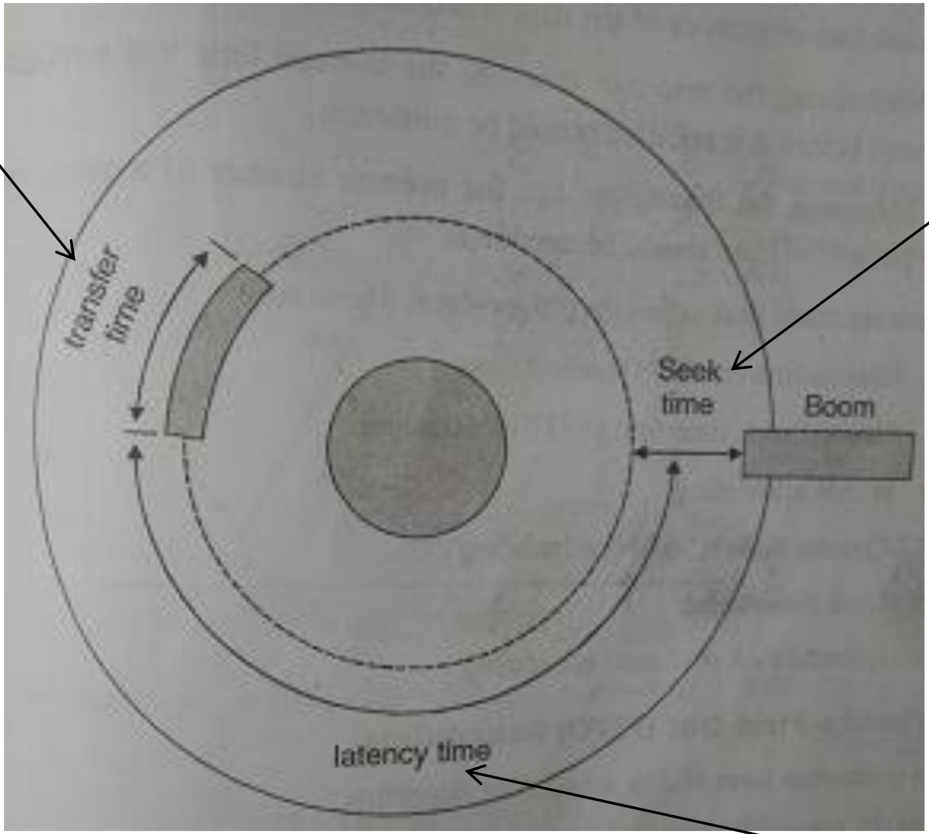
Organization of Disks

- Disk contains concentric **tracks.**
- Tracks are divided into **sectors**
- A **sector** is the smallest addressable unit in a disk.

Moving-head Disk Machanism



Component of Disk



time spent in actually moving data to from the disk surface

Time to move head to desired track

Time taken to reach the portion of disk on which data record is stored

Mass Storage Structure

- **Seek Time: It is the time required to move a head to the required track.**

- **Latency Time: It is the amount of time it takes the portion of disk on which data record is stored to spin under read/write head.**

- **Transfer time: time spent in actually moving data to from the disk surface.**

It is amount of information to be read, the number of bytes per track and rotation speed.

Total Access Time: it is **Seek time + Latency Time + Transfer Time**

Disk Scheduling

Scheduling is mechanism of arranging requests in order to save un-necessary rotations by disk arm and disk pointer.

OS takes I/O requests from queue and process them one by one. Algo. used to select **which I/O request is going to be selected first** is called Disk Scheduling Algorithm

Objective:

1. **Minimize response time**
2. **Maximize Throughput.**

Response Time: Avg. time that a request must wait before it is satisfied

Throughput: Avg. number of requests satisfied per unit of time.

Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth.
- Access time has two major components
  - *Seek time* is the time for the disk are to move the heads to the cylinder containing the desired sector.
  - *Rotational latency* is the additional time waiting for the disk to rotate the desired sector to the disk head.
- Minimize seek time
- Seek time ≈ seek distance

FCFS

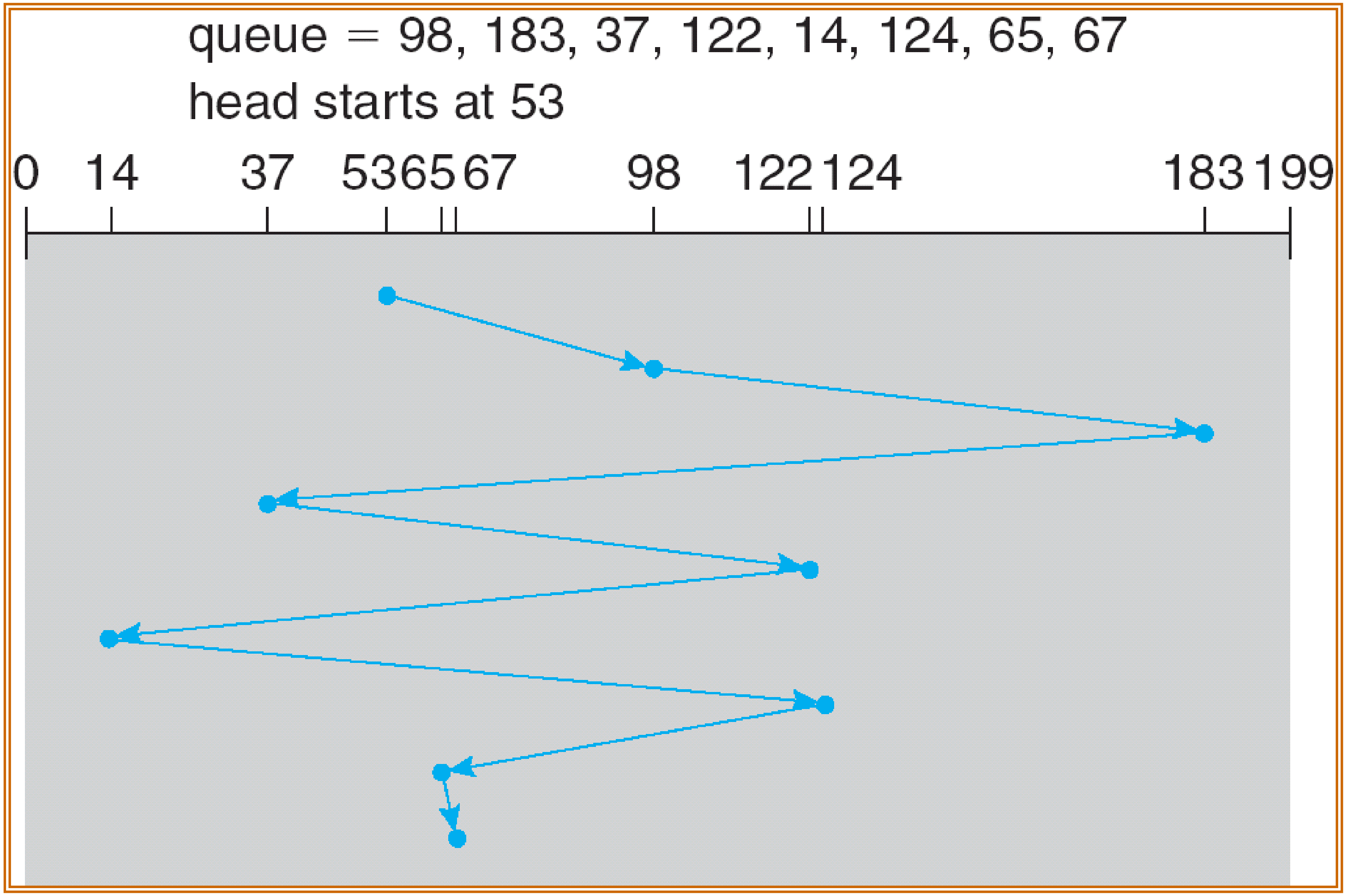
- Several algorithms exist to schedule the servicing of disk I/O requests.
- A disk size of 200 tracks receives the requests for the blocks for I/O operation. No. of blocks requested on disk are:

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

Head pointer/ Disk arm is at 53

FCFS

Illustration shows total head movement of 640 cylinders.



queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53

FCFS

Illustration shows total head movement of 640 cylinders.

Total Head Movements:

|53-98| + |98-183| + |183- 37| + | 37-122| + |122-14| + |14-124| +

|124-65| + |65-67|

= 640

FCFS Example

A disk size of 200 tracks receives the requests for the blocks for I/O operation. No. of blocks requested on disk are: 23, 89, 132, 42, 187, 60. Disk arm is at track 100.

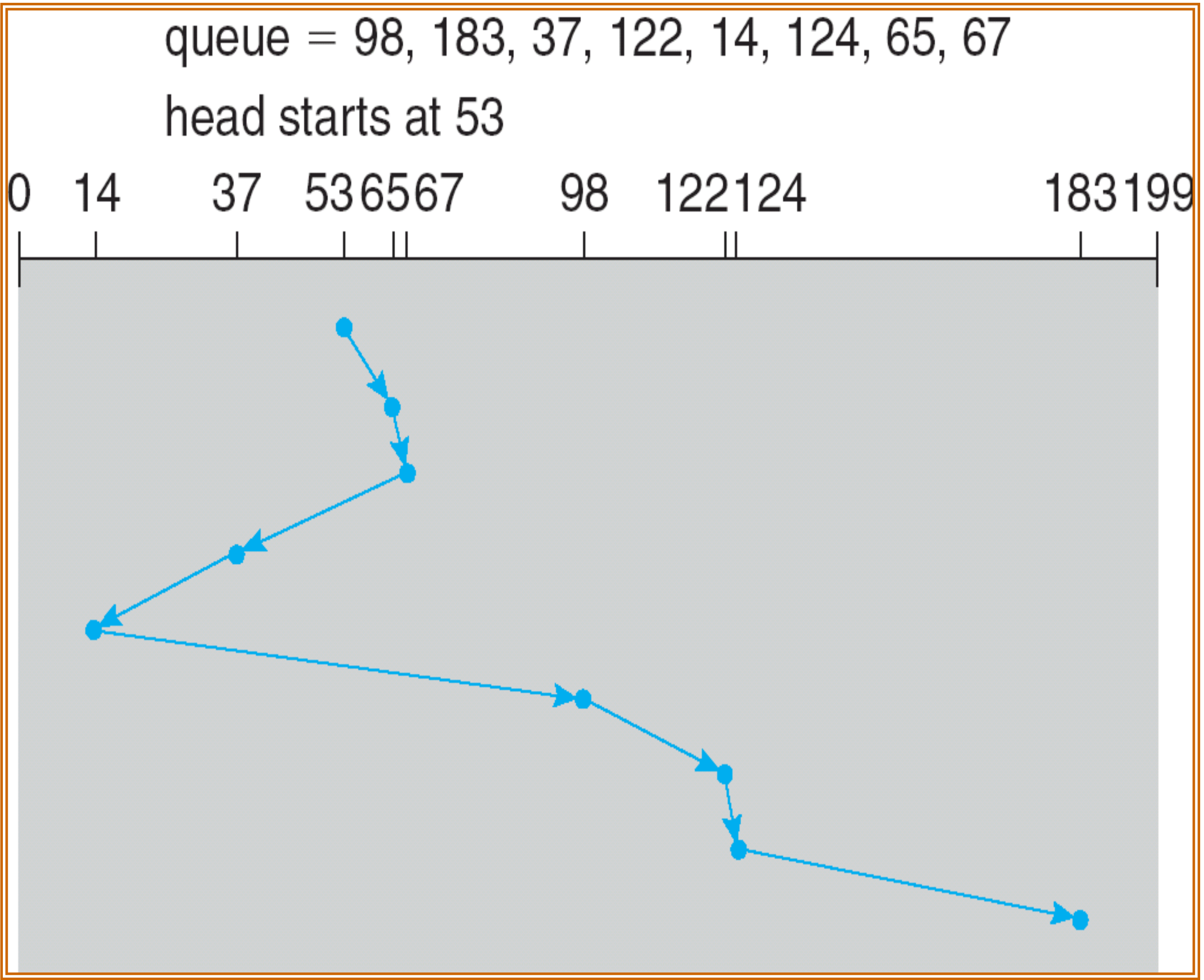
Hint: Total head movements: 548

SSTF- Shortest Seek Time First

- **Selects the request with the minimum seek time from the current head position.**
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests.

- Illustration shows total head movement of 236 cylinders.

SSTF (Cont.)



5) 53-98=45
8) 53-183=130
3) 53-37=16
6)53-122=69
4)53-14=39
7) 53-124=71
1) 53-65=12
2) 53-67=14

SSTF- Shortest Seek Time First

- **Exp: 23,89,132,42,187,60. Disk head is at position 100**

Find total head movements.

Hint: 241

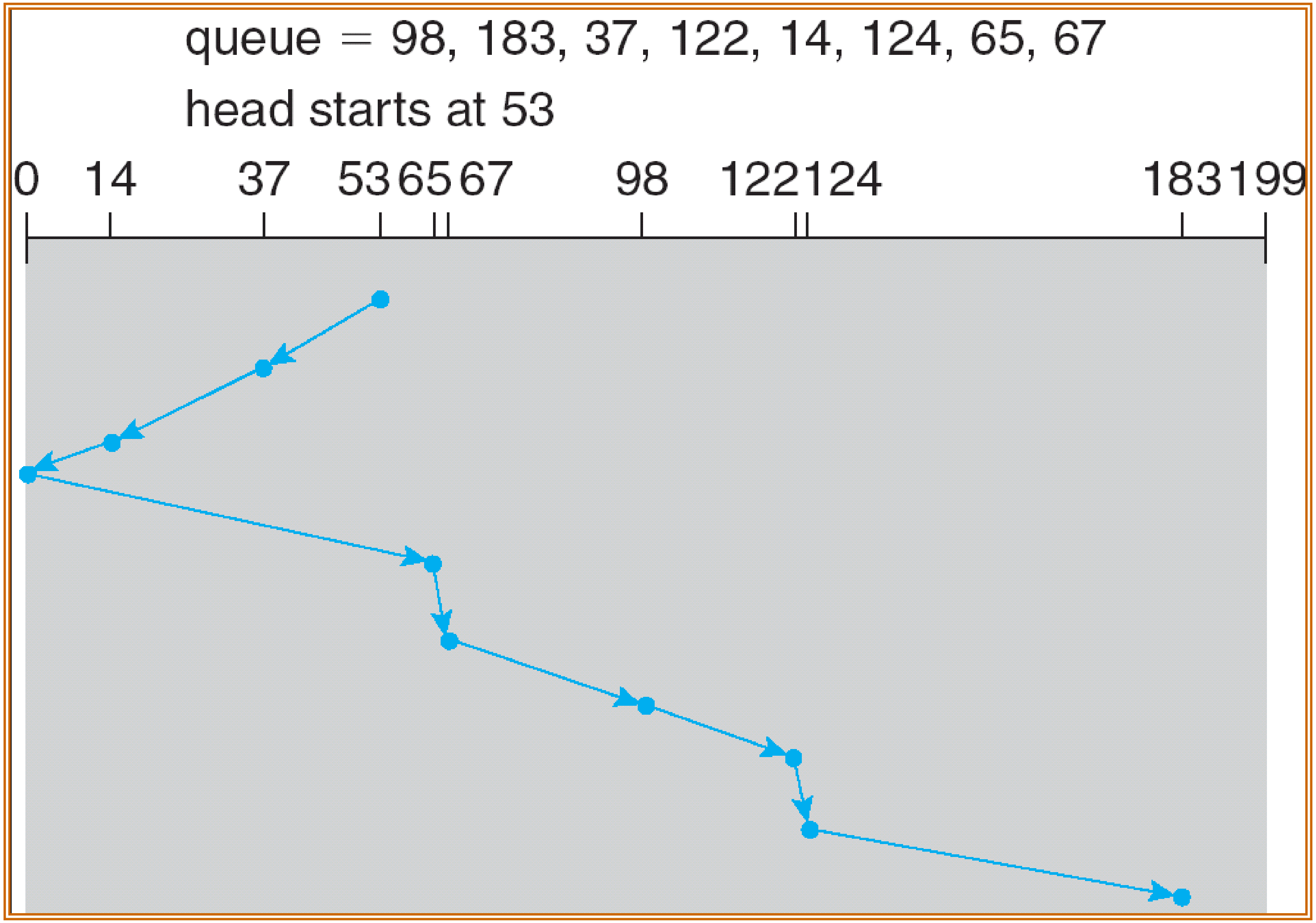
SCAN

- The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.

- A request arriving behind the head will have to wait until the arm moves to the end of the disk, reverses direction and comes back.

- Sometimes called the ***elevator algorithm*****.**
- Illustration shows total head movement of 208 cylinders.
- **Example: Previously head is at 60. so from 60 we go to 53 i.e left**
- **But if previously if head is at 5 then direction will be towards right. By deafult: left direction.**

SCAN (Cont.)


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

SCAN

- Advantage
- Throughput is better than FCFS
- If a request arrive in a queue just in front of the head, it will be serviced immediately.

Disadvantage:

The disk arm always starts from the beginning, no matter other number of requests are present on the other end of disk.

C-Scan

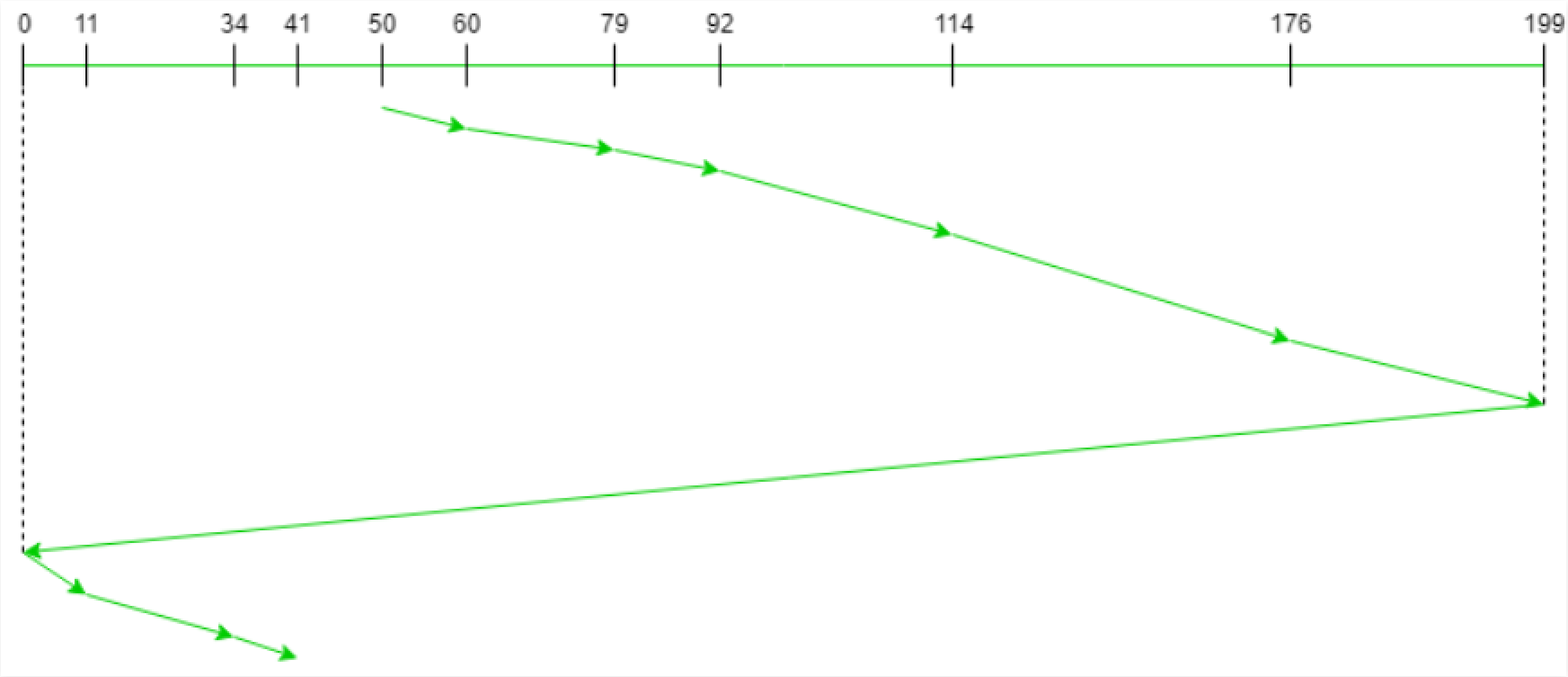
- C-SCAN moves the head from one end servicing all the requests to the other end.
- However, as soon as the head reaches the other end, it immediately returns to the beginning of the disk without servicing any requests on the return trip and starts servicing again once reaches the beginning.
- This is also known as the “Circular Elevator Algorithm

C-Scan

- It is an enhancement of SCAN algorithm.
- Head moves from one end of disk to the other servicing the request along the way.
- When the disk arm reaches the end it quickly returns to the other end without fulfilling any request in the way
- It restricts servicing request from one direction only.
- My cylinder are rising from 0 to n-1 and wise versa. Next will be 0
- It will move in increasing order of cylinder number.
- Head will go to extreme ends.
- **Move towards increasing direction of cylinder number**

C-Scan Example

- Request sequence = {176, 79, 34, 60, 92, 11, 41, 114}
- Initial head position = 50
- Direction = right(We are moving from left to right)



C-Scan

Suppose a disk is having 200 cylinder from 0 to 199. disk is currently searching for disk head 53. previous request was at head 60. the queue of pending request is in FIFO order.

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

Find total head movements.

Ans:183

C-Scan

- Consider request queue: 23,89,132,42,187,60

Head starts at number:100

Hint: head movements: |100-89|+|89-60|+|60-42|+|42-23|+|23-0|+ |0-199|+|199-187|+|187-132|

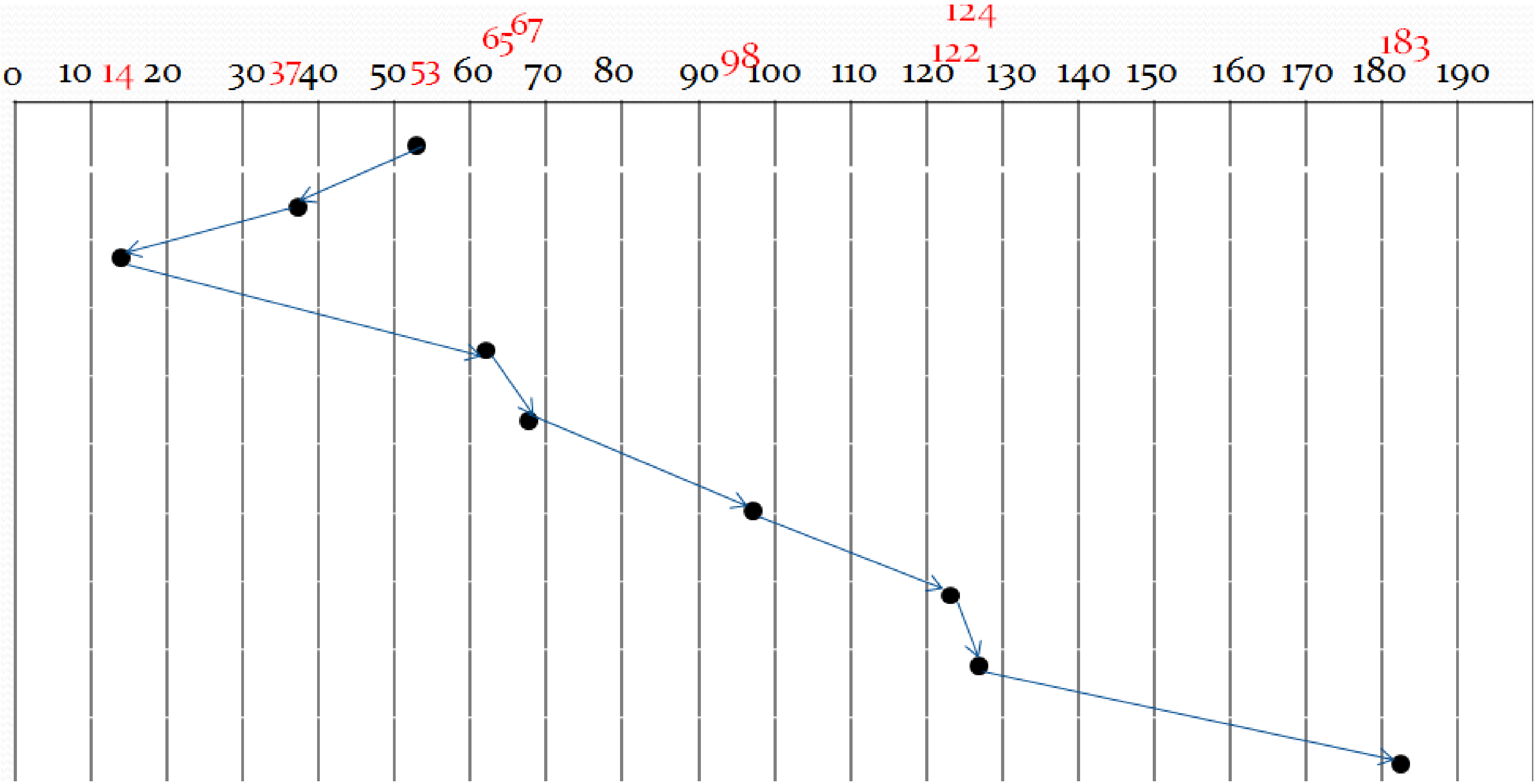
= 311

If previous head position is not given then see the total number of blocks from current head, then pick direction towards maximum number of blocks.

LOOK or C-LOOK

- Same as SCAN
- Arm Does not go to end.
- Will not go beyond max(1.e 183) and beyond min(i.e. 14)
- Arm only goes as far as the final request in each direction,
- Then it reverses direction immediately, without first going all the way to the end of the disk.

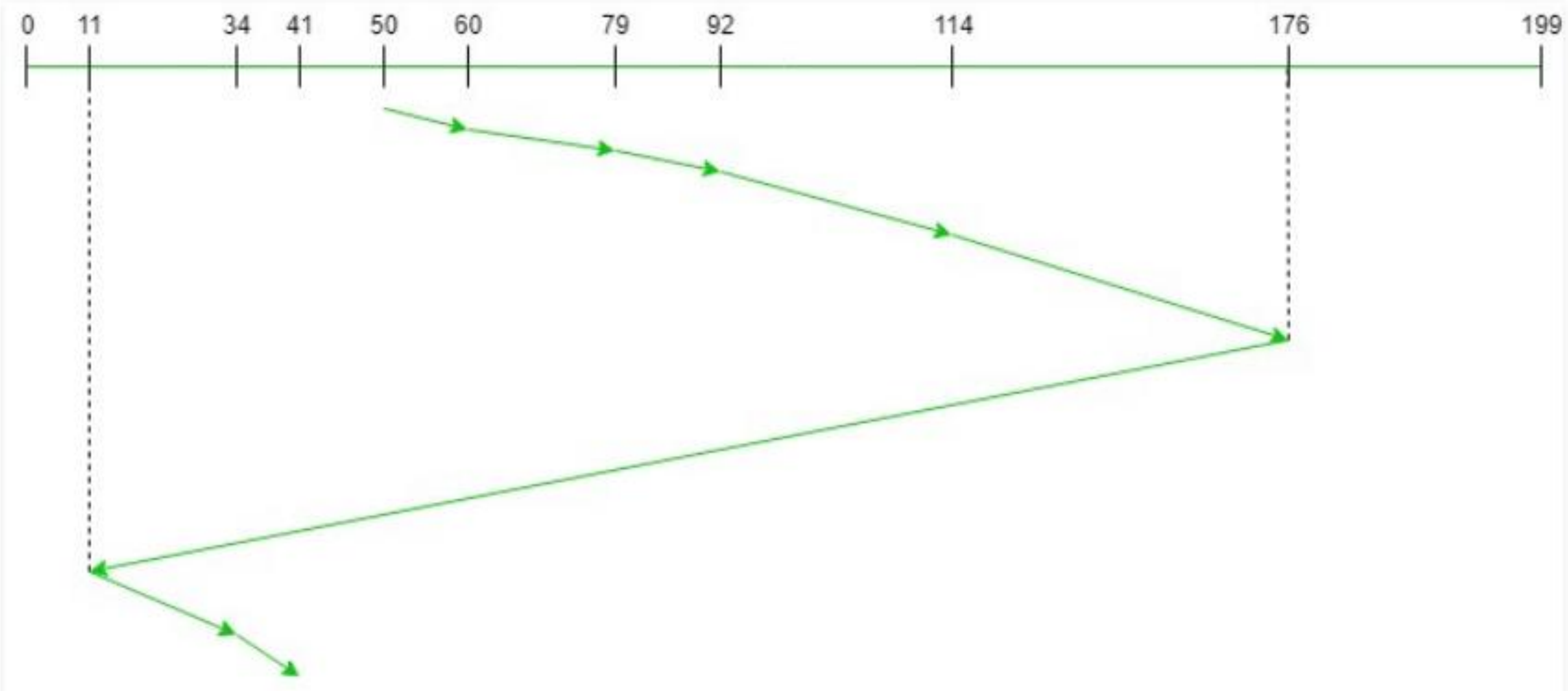
LOOK



Total head movements: 208

C- LOOK

Request sequence = {176, 79, 34, 60, 92, 11, 41, 114}
Initial head position = 50
Direction = right (Moving from left to right)



Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- 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.