

# CS-3103 : Operating Systems : Sec-A (NB) : Disk Management.....

OPERATING  
SYSTEM



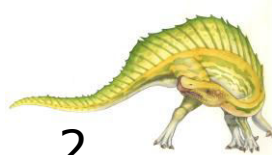
Computer Operating Systems: OS Families for Computers



# Disk Scheduling

---

- The problem of unbounded queue is solved by introducing **scheduling of I/O requests** made on the hard disk.
- Each request is examined and, based on the current position of the R/W head of the hard disk, a decision is made as to when the data demanded by the requesting process should be served.

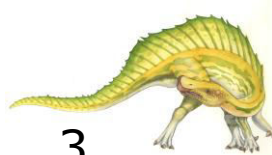




# Disk Scheduling Algorithms

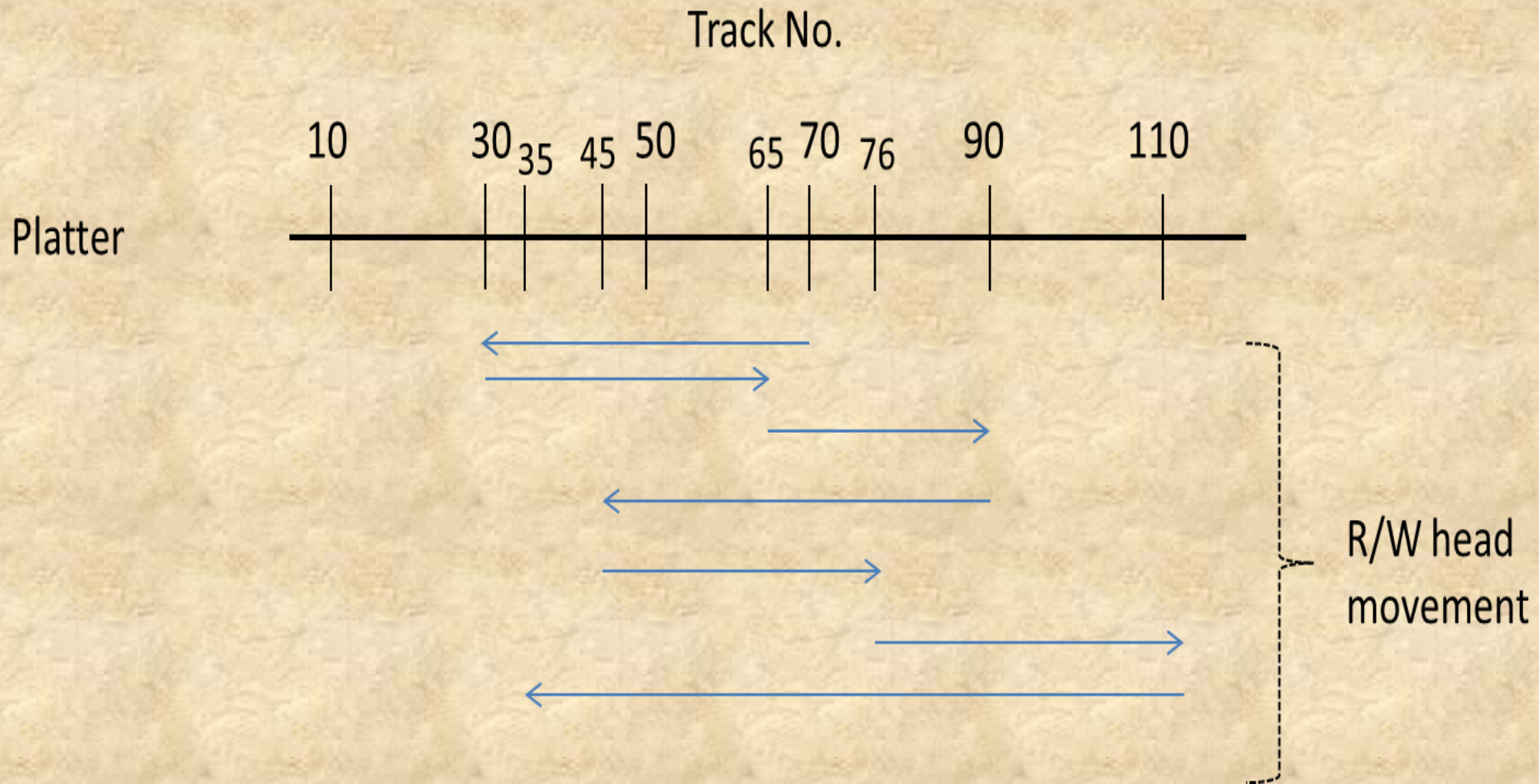
---

- Disk performance has two parameters:
  - **Access time:** Seek time + Rotational latency + Data transfer time
  - **Disk bandwidth:** It is the data transfer capacity of the disk to and from main memory.  $\text{Disk bandwidth} = \frac{\text{Data in bytes transferred}}{\text{Total time taken}}$



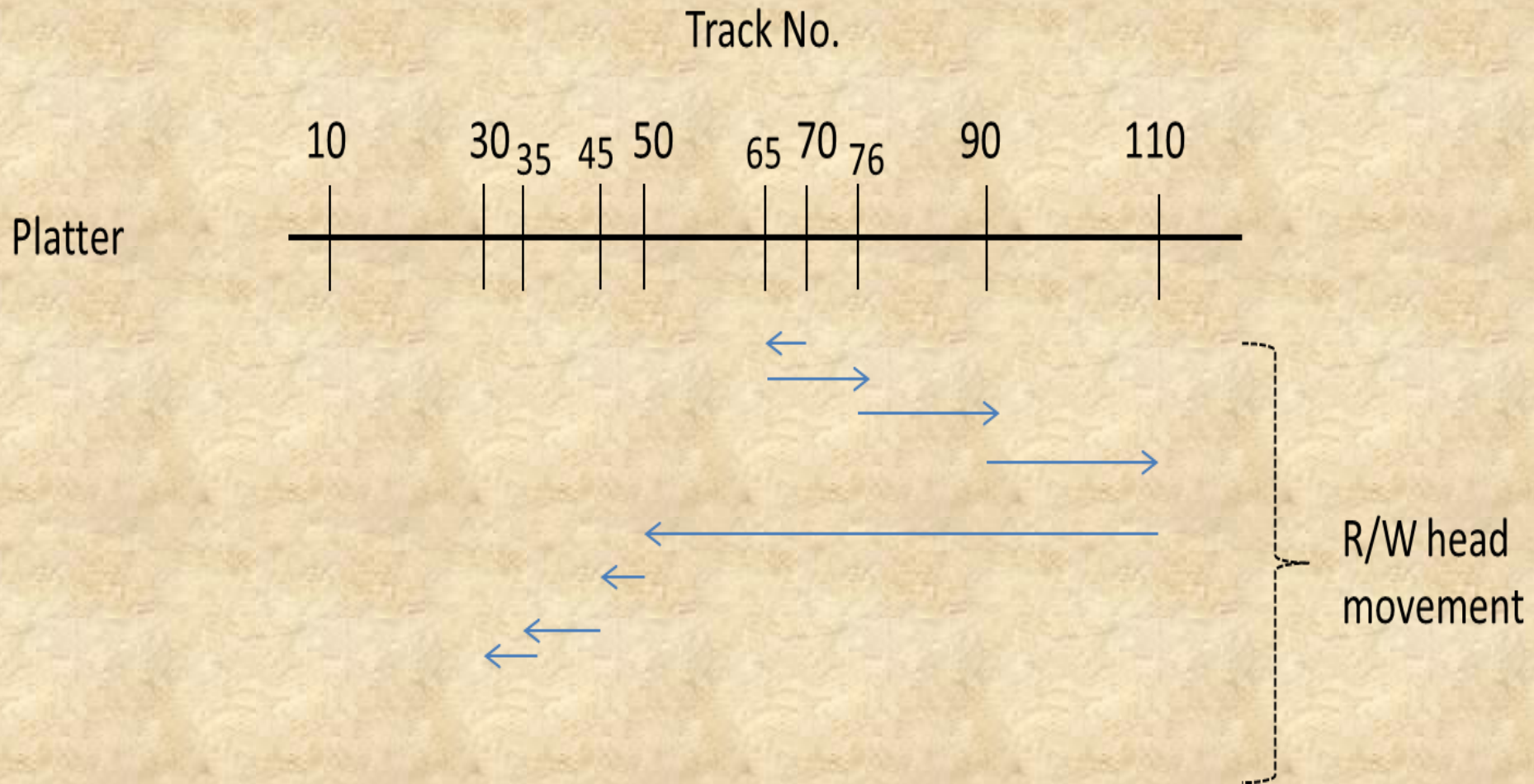
# First Come First Served (FCFS)

- The I/O requests are served in the order of their arrival.
- Example: Let the disk I/O request arrive in the sequence no. of tracks 30 65 90 45 76 110 35. The R/W arm movement as per the FCFS policy is:



# Shortest Seek Time First (SSTF)

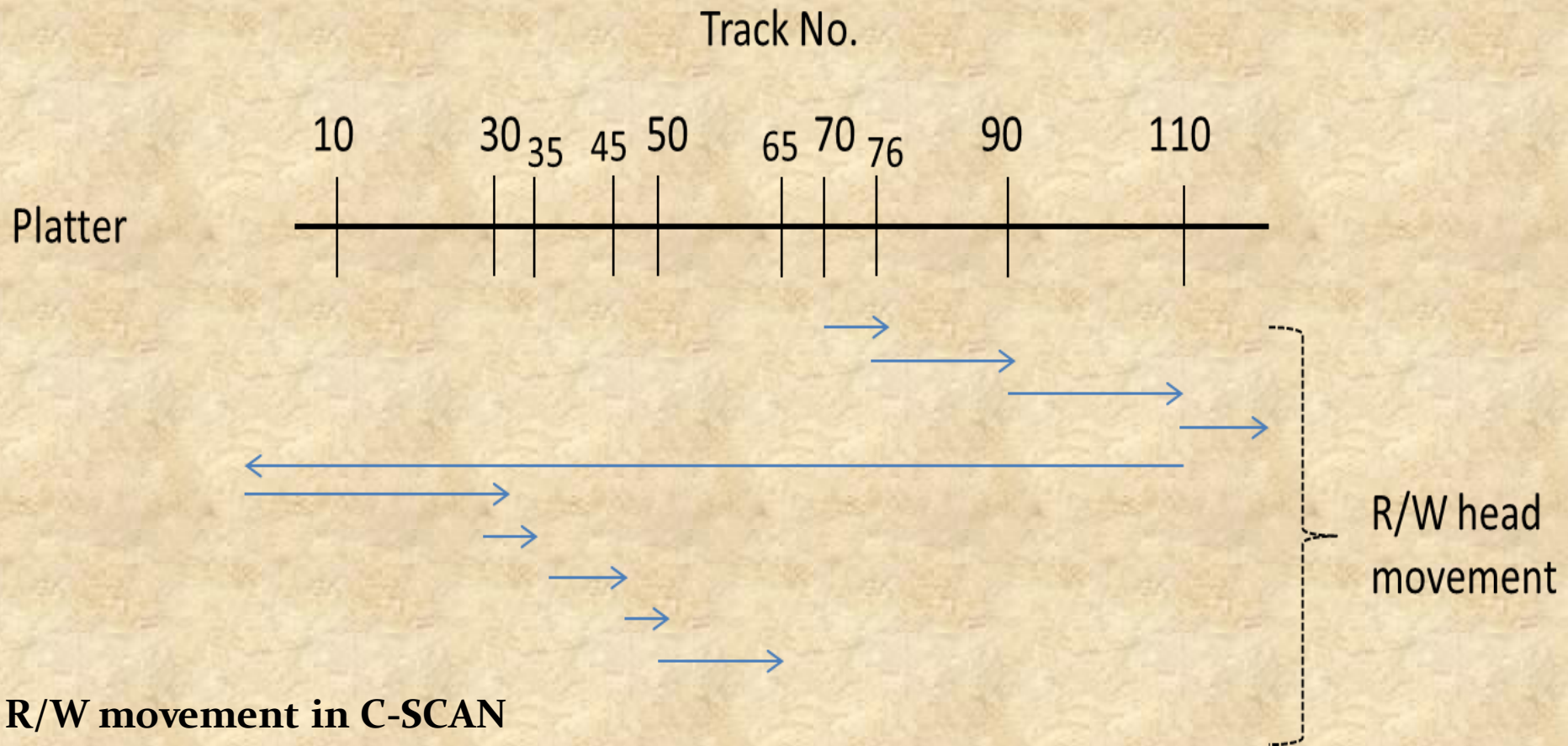
- In this strategy, the next I/O operation whose sector is closest to the current position of the R/W head is selected.
- Example: Let the disk I/O request arrive in the sequence no. of tracks 30 65 90 45 76 110 35.





# Circular SCAN (C-SCAN) Algorithm

- The head moves from one end of the disk to the other, servicing pending I/O requests during the onward scan. After reaching the other end, the head immediately returns to the beginning of the disk. It does not serve the pending requests during the reverse scan.



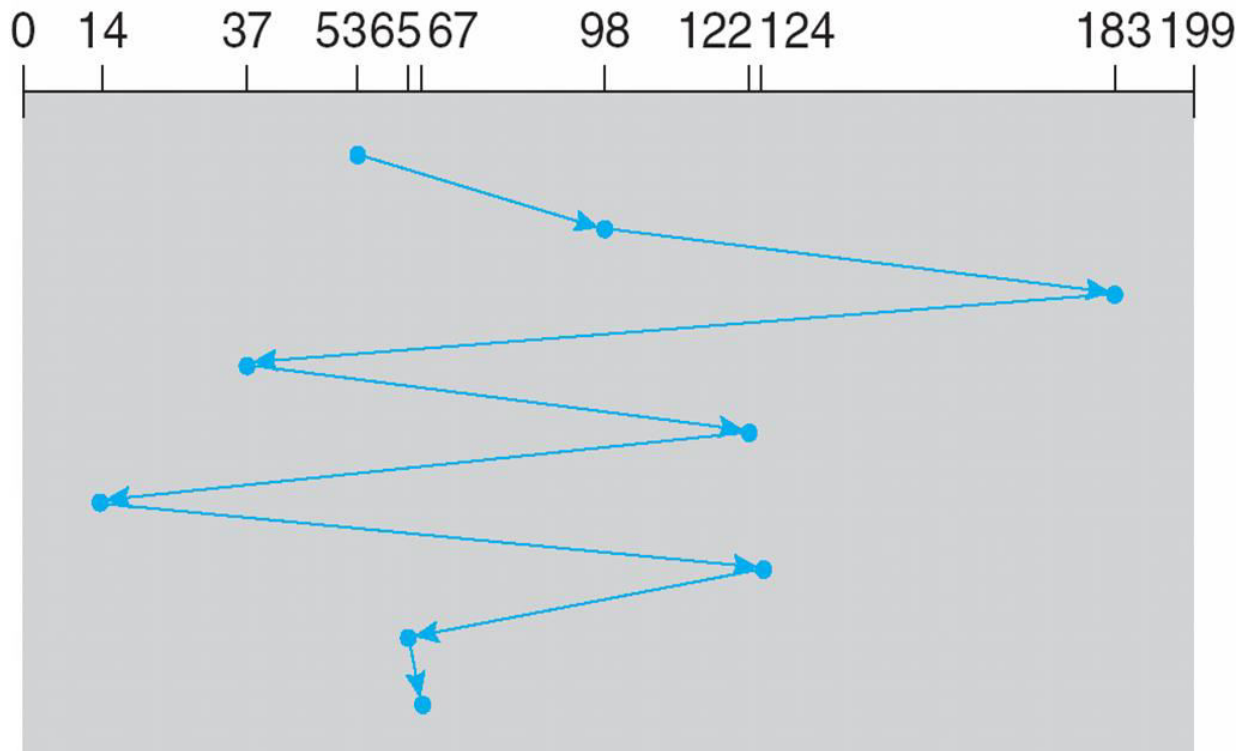


# FCFS

Illustration shows total head movement of 640 cylinders

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

head starts at 53





# 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.
- **SCAN algorithm** Sometimes called the **elevator algorithm**
- Illustration shows total head movement of 208 cylinders
- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest



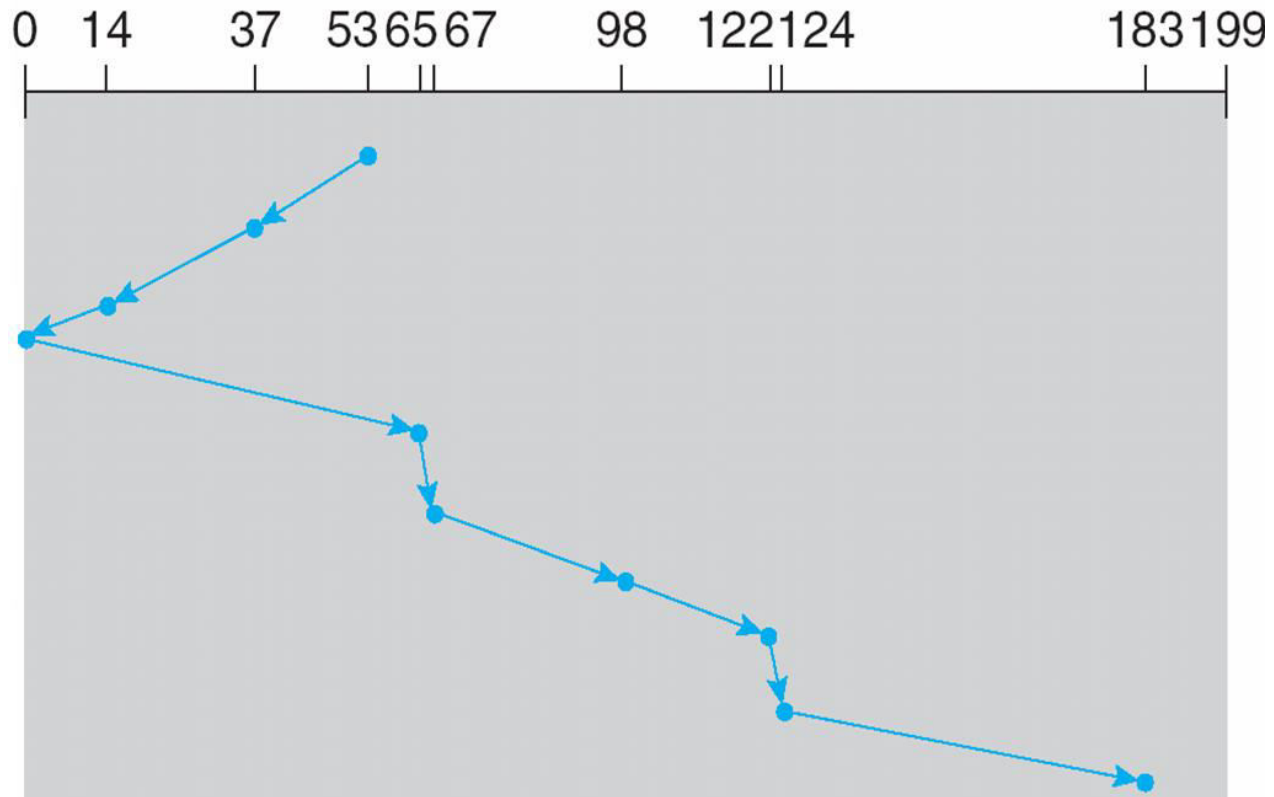




# SCAN (Cont.)

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

head starts at 53





# C-SCAN

---

- Provides a more uniform wait time than SCAN
- The head moves from one end of the disk to the other, servicing requests as it goes
  - When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one
- Total number of cylinders?

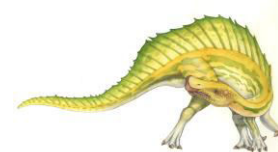
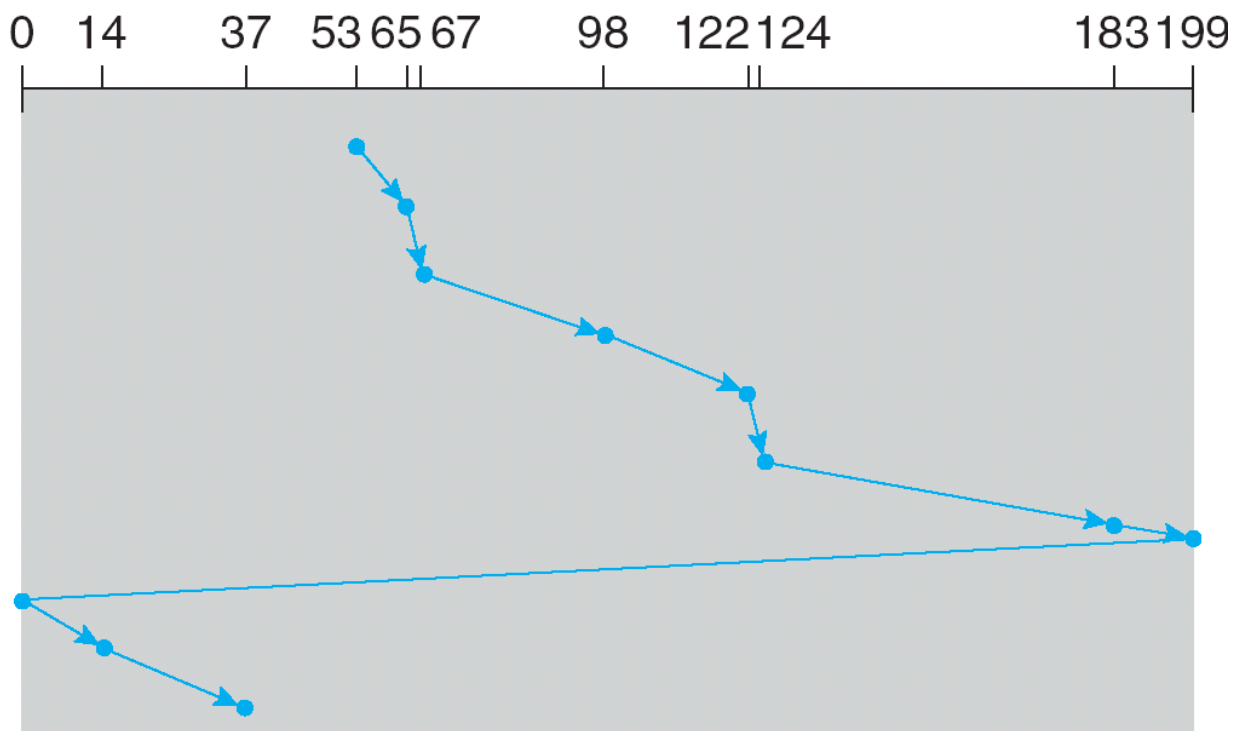




# C-SCAN (Cont.)

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

head starts at 53



# Exercises

Suppose that a disk has 200 cylinders, numbered 0 to 199. The drive is currently serving a request at cylinder 53, and the previous request was at cylinder 25. The queue of pending requests, in FIFO order is

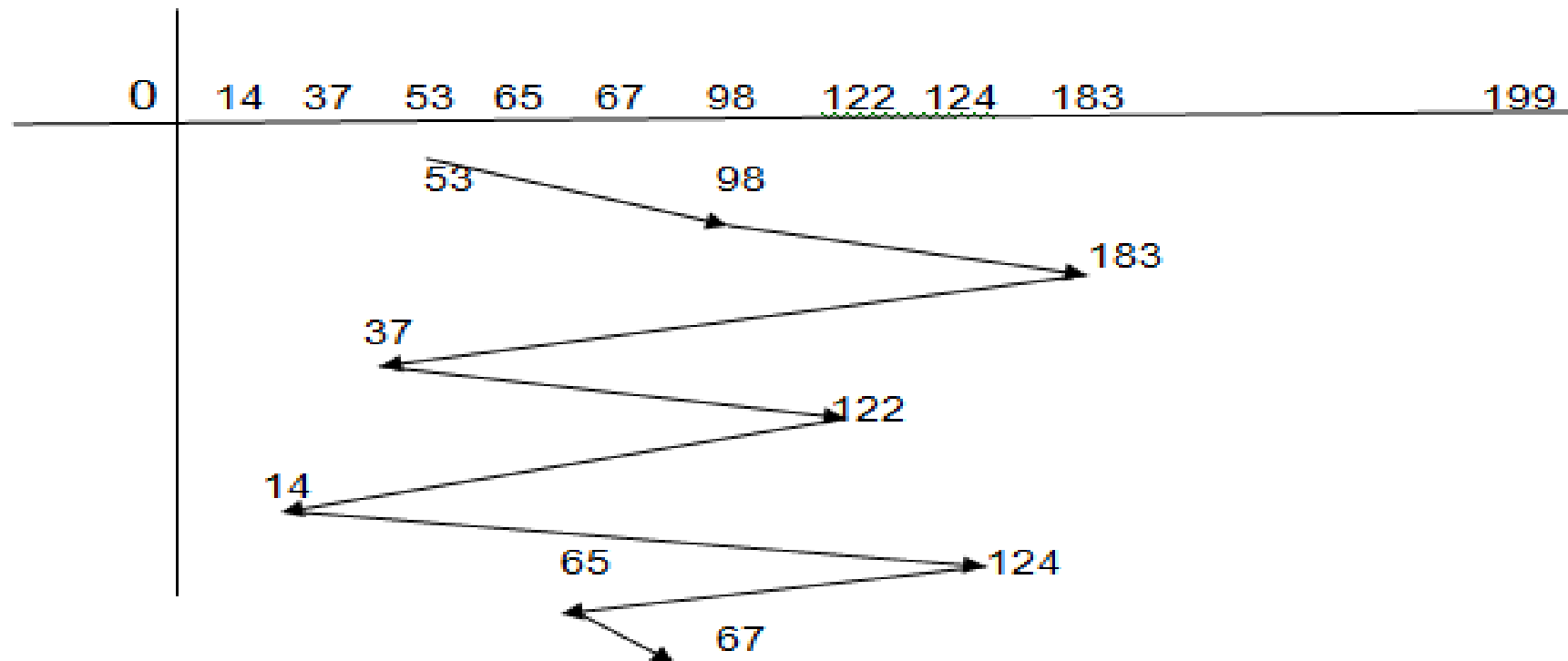
98, 183, 37, 122, 14, 124, 65, 67

Starting from the current head position, show the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling disciplines?

- I. FCFS
- II. SSTF
- III. C-Scan

# Answers

## I. FCFS



***The total distance:***

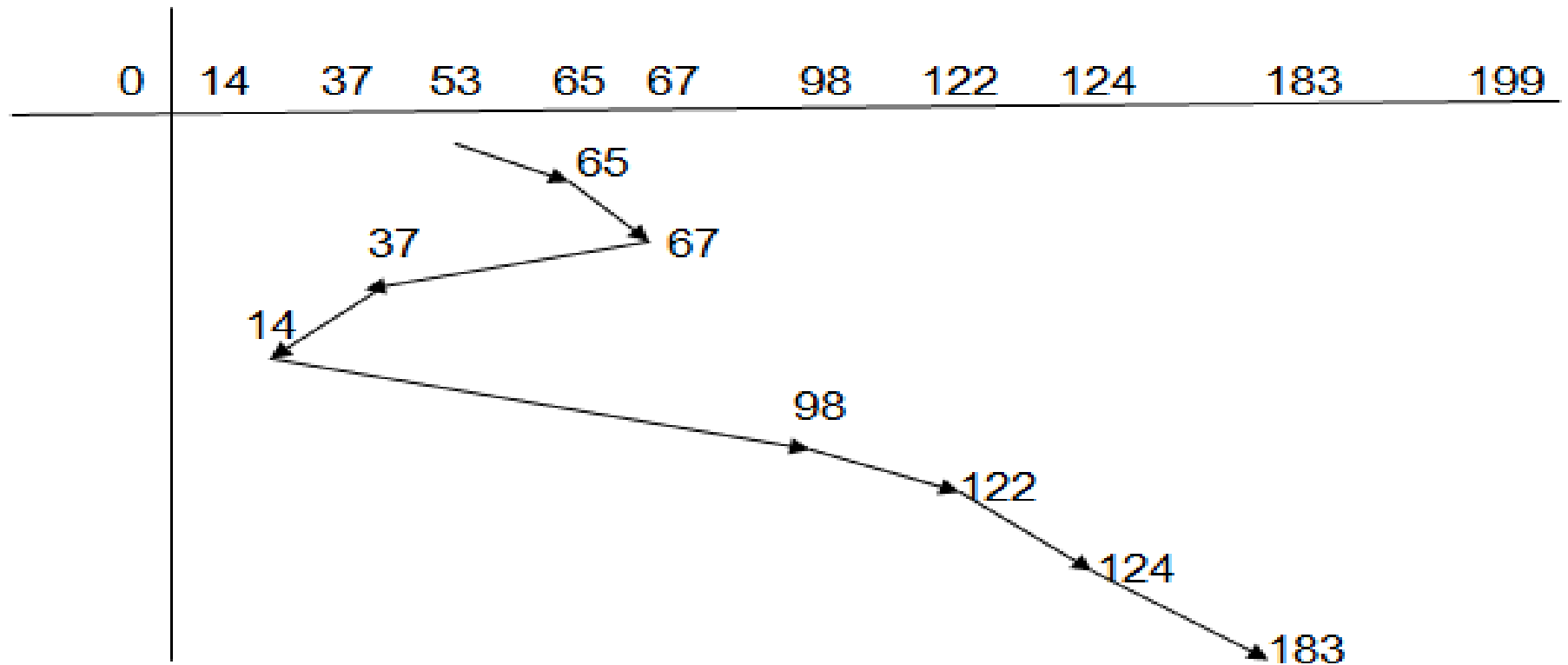
$$(98-53)+(183-98)+(183-37)+(122-37)+(122-14)+(124-14)+(124-65)+(67-65)= \mathbf{640 \text{ Cylinders}}$$

***In another way:***

**FCFS:** 53, 98, 183, 37, 122, 14, 124, 65, 67

# Answers

## ii. SSTF



***The total distance:***

$$(65-53)+(67-65)+(67-37)+(37-14)+(98-14)+(122-98)+(124-122)+(183-124)= \mathbf{236 \text{ Cylinders}}$$

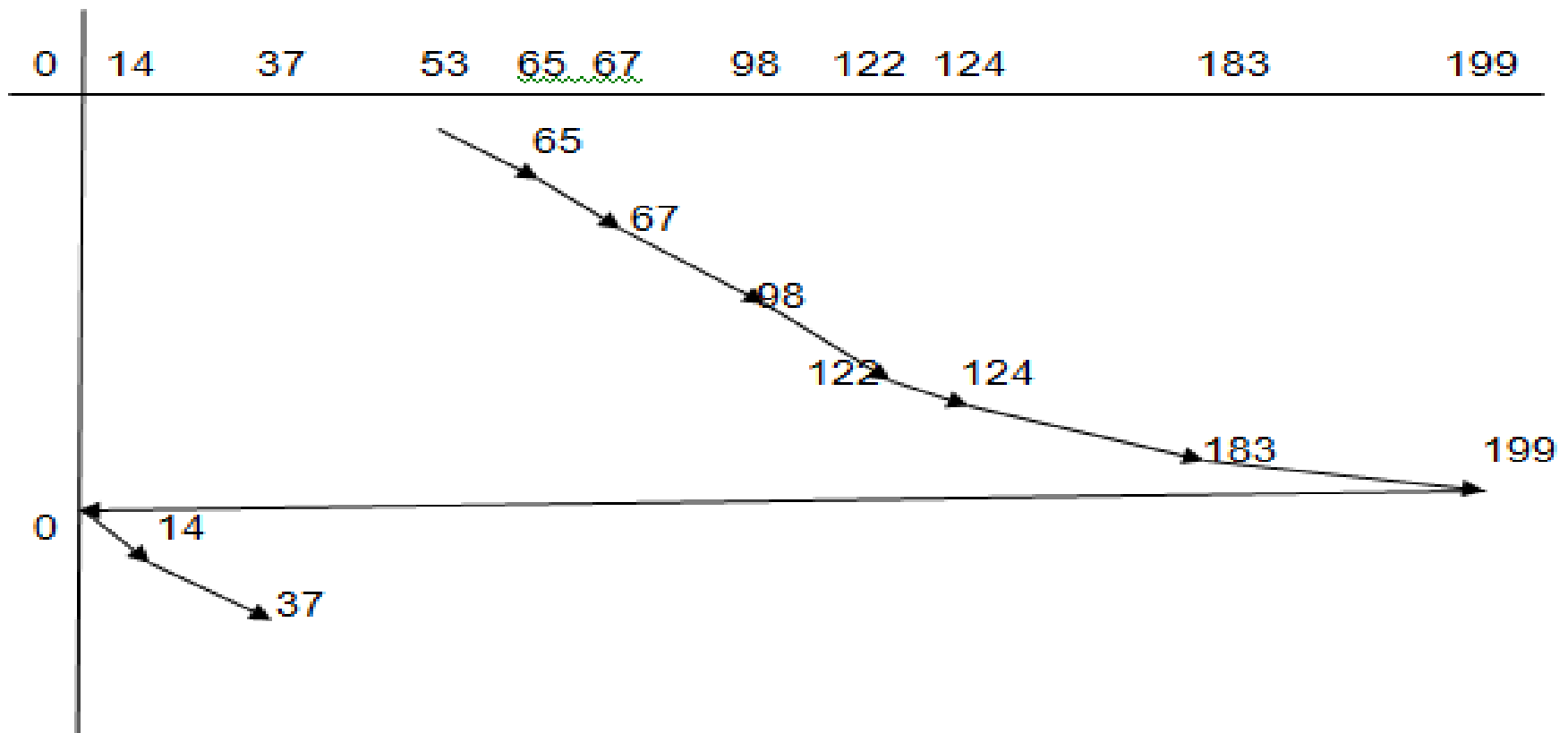
***In another way:***

**SSTF:** 53, 65, 67, 37, 14, 98, 122, 124, 183



# Answers

## iii. C-Scan



***The total distance***

$$(53-65)+(67-65)+(98-67)+(122-98)+(124-122)+(183-124)+(199-183)+(14-0)+(37-14)=$$

**In another way:**

**C-Scan:** 53 , 65 , 67 , 98 , 122 , 124 , 183 , 199 , 0 , 14 , 37