CSE 421/521 - Operating Systems
Fall 2018

LECTURE - XX

MASS STORAGE & IO - I

Tevfik Koşar

University at Buffalo

November 13th, 2018

Overview of Mass Storage Structure

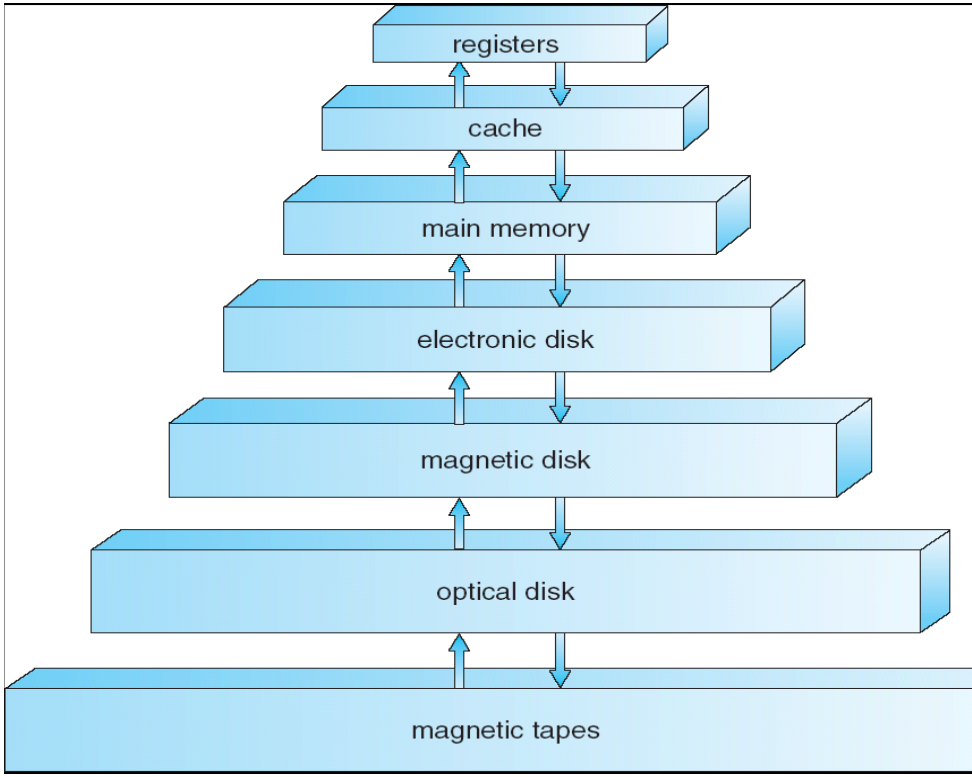
- Magnetic disks provide bulk of secondary storage of modern computers
  - Drives rotate at 90 to 300 times per second
  - Transfer rate 1-2 Gbps
  - Lots of mechanical components
  - Slow performance for random access
  - Better performance for streaming / sequential access
- Solid State Disks becoming more popular
  - architecture similar to memory
  - an order of magnitude (or more) faster than magnetic disks
  - more expensive than magnetic disks
  - Good performance in random reads

Overview of Mass Storage Structure (Cont.)

- Magnetic tape
  - Relatively permanent and holds large quantities of data
  - Random access ~1000 times slower than disk
  - Once data under head, transfer rates comparable to disk
  - Much cheaper than disk
  - Mainly used for backup, long-term storage of infrequently-used data
  - Hundreds of TB typical storage
  - Common technologies are 4mm, 8mm, 19mm, LTO-2 and SDLT

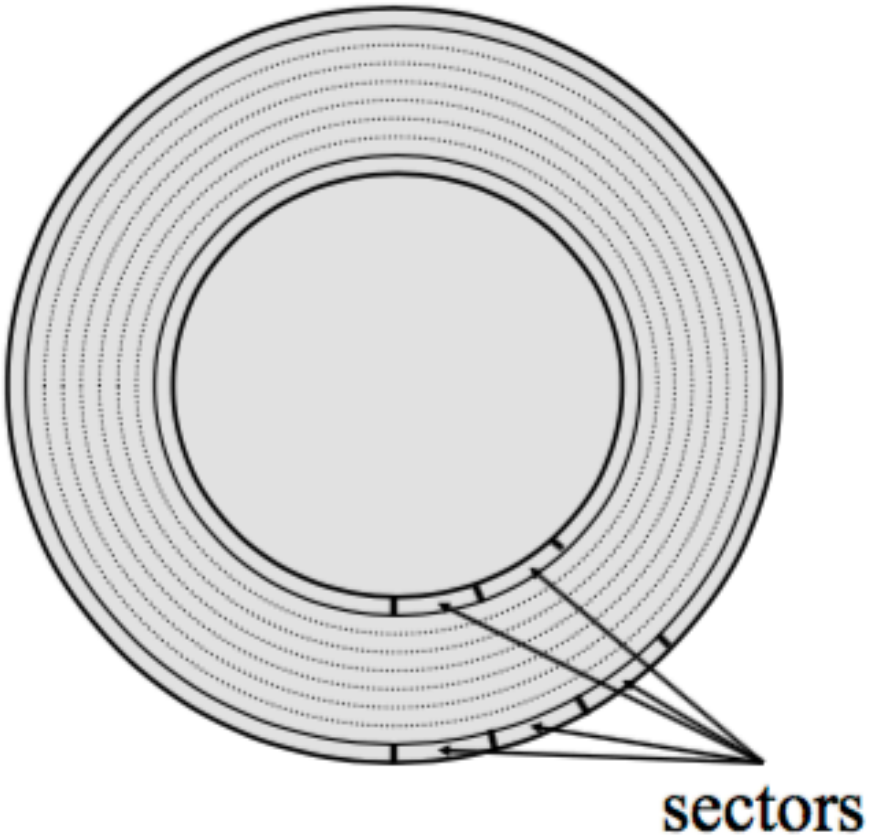
Hierarchical Storage Management (HSM)

- A hierarchical storage system extends the storage hierarchy beyond primary memory and secondary storage to incorporate tertiary storage — usually implemented as a jukebox of tapes or removable disks.
- Usually incorporate tertiary storage by extending the file system.
  - Small and frequently used files remain on disk.
  - Large, old, inactive files are archived to the jukebox.
- HSM is usually found in supercomputing centers and other large installations that have enormous volumes of data.



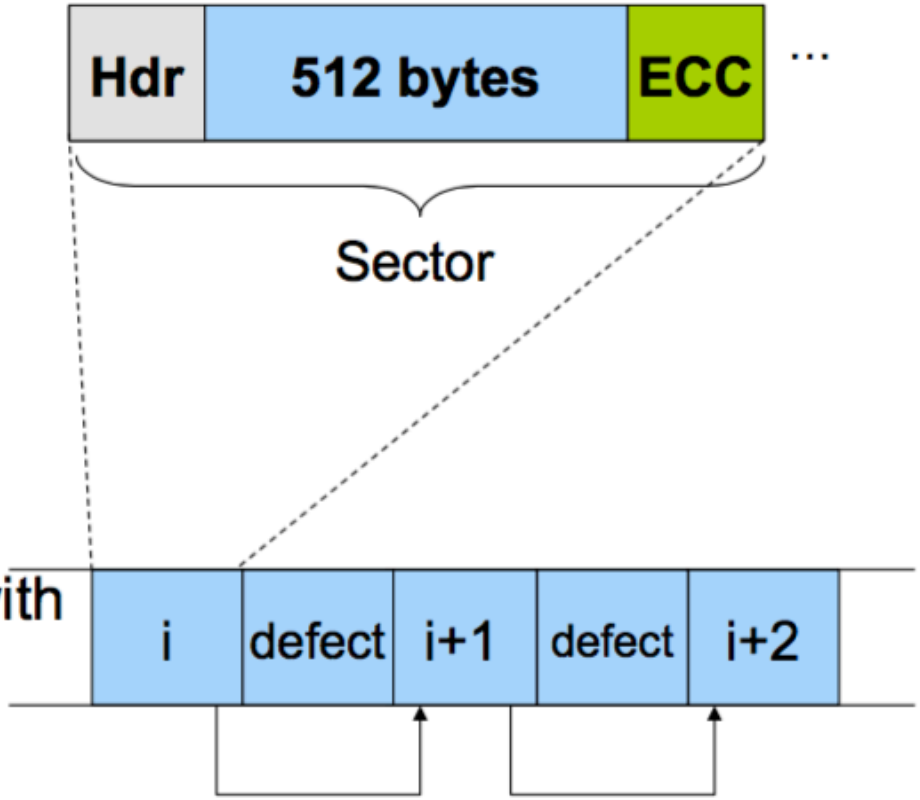
Disk Organization

- Disk surface
  - Circular disk coated with magnetic material
- Tracks
  - Concentric rings around disk surface, bits laid out serially along each track
- Sectors
  - Each track is split into arc of track (min unit of transfer)

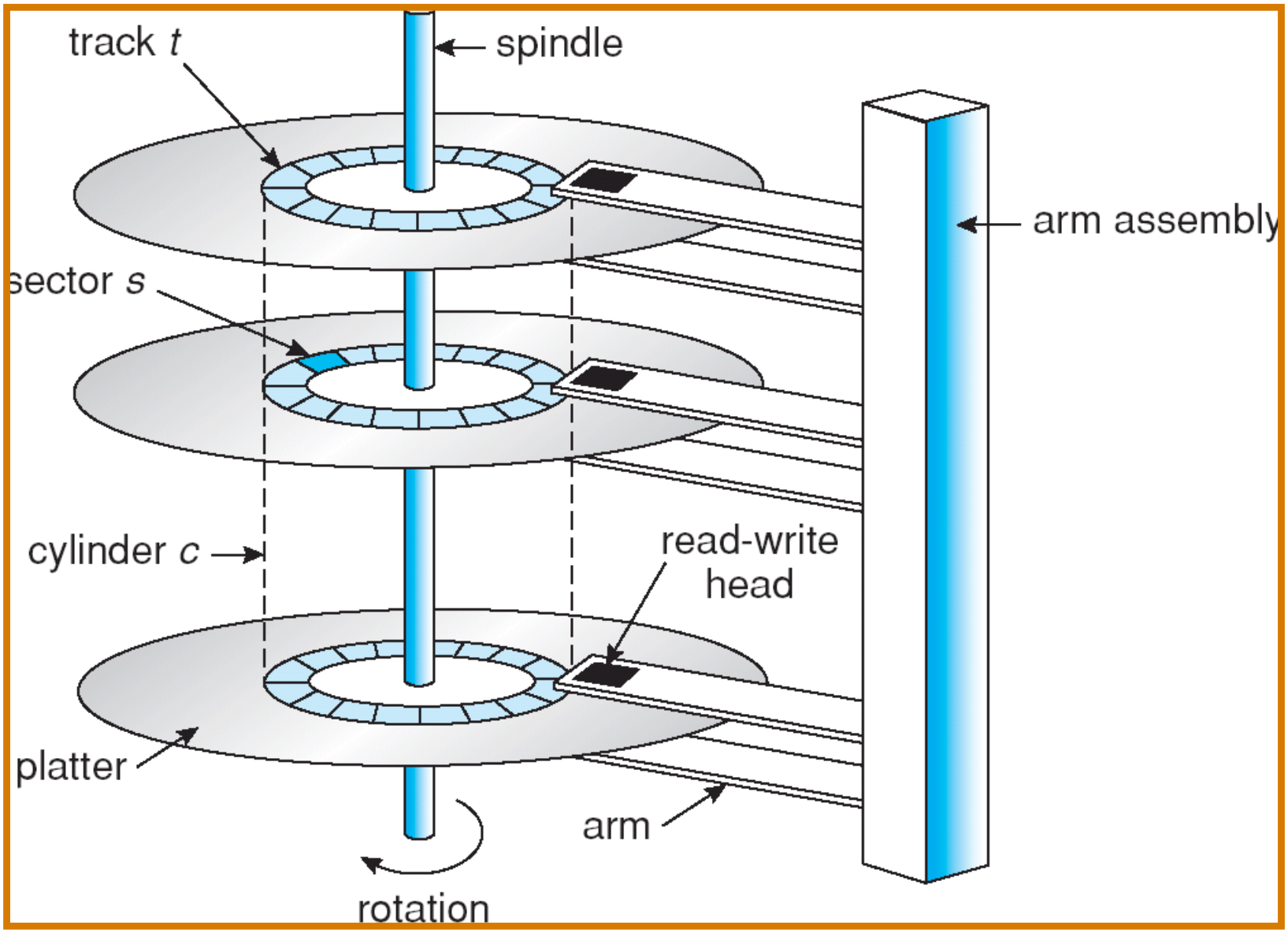


Disk Sectors

- Where do they come from?
  - Formatting process
  - Logical maps to physical
- What is a sector?
  - Header (ID, defect flag, …)
  - Real space (e.g. 512 bytes)
  - Trailer (ECC code)
- What about errors?
  - Detect errors in a sector
  - Correct them with ECC
  - If not recoverable, replace it with a spare
  - Skip bad sectors in the future



Moving-head Disk Mechanism



Disk Performance Overheads

- **Disk latency** = positioning time + transfer time
- **Transfer time** is time to transfer data onto/off disk, and **transfer rate** is the rate at which data flows.
- **Positioning time** (**random-access time**) is time to move disk arm to desired cylinder (**seek time**) and time for desired sector to rotate under the disk head (**rotational latency**)
- **Head crash** results from disk head making contact with the disk surface

Example: Toshiba Disk

Size	
Platters/Heads	2/4
Capacity	320 GB
Performance	
Spindle speed	7200 RPM
Average seek time read/write	10.5 ms/ 12.0 ms
Maximum seek time	19 ms
Track-to-track seek time	1 ms
Transfer rate (surface to buffer)	54–128 MB/s
Transfer rate (buffer to host)	375 MB/s
Buffer memory	16 MB
Power	
Typical	16.35 W
Idle	11.68 W

Question

- How long to complete 500 random disk reads, in FIFO order?
  - Seek: average 10.5 msec
  - Rotation: average 4.15 msec
  - Transfer: 5-10 usec
- 500 * (10.5 + 4.15 + 0.01)/1000 = 7.3 seconds

Question

- How long to complete 500 sequential disk reads?
  - Seek: average 10.5 msec
  - Rotation: average 4.15 msec
  - Transfer: 5-10 usec

✦**Total: 10.5 + 4.15 + 500 x 0.01 = 19.65 ms**

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
- Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.

Disk Scheduling (Cont.)

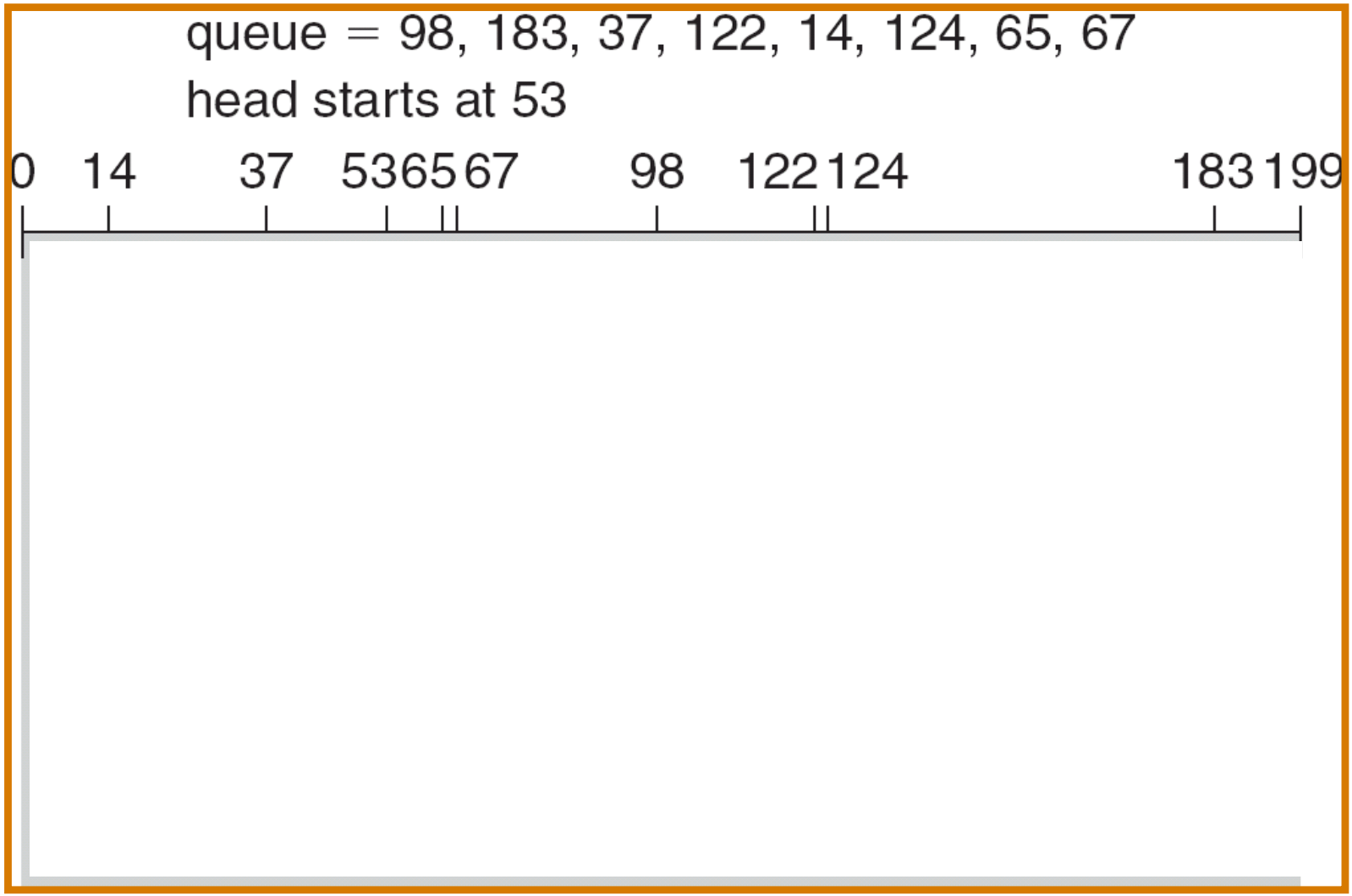
- Several algorithms exist to schedule the servicing of disk I/O requests.

- We illustrate them with a request queue (0-199).

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

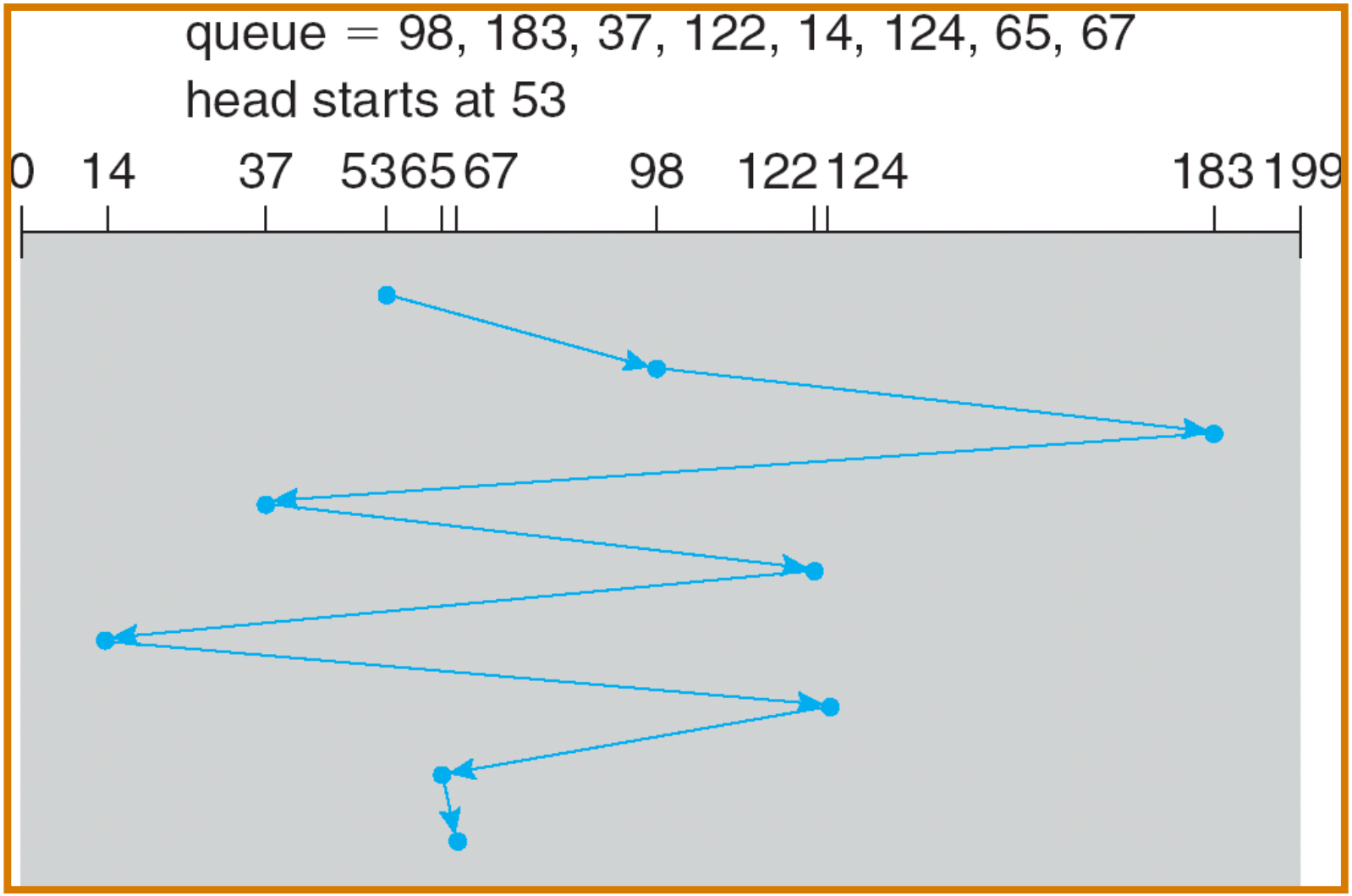
  Head pointer at 53

FCFS


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

FCFS

Illustration shows total head movement of 640 cylinders.



FCFS (Cont.)

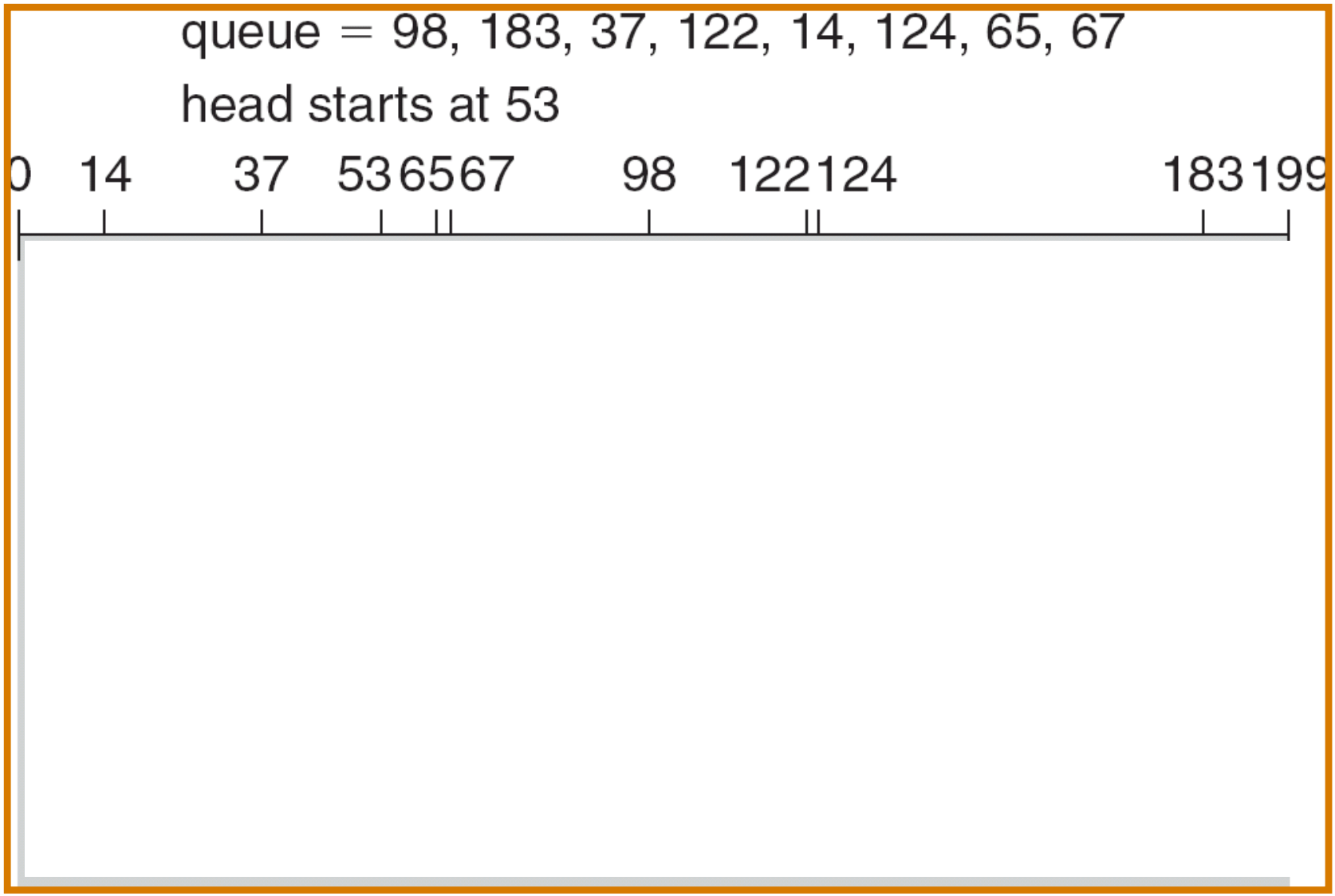
- Pros:
  - Fairness among request
  - In the order applications expect

- Cons:
  - Arrival may be on random spot on the disk
  - Long seeks —> poor performance

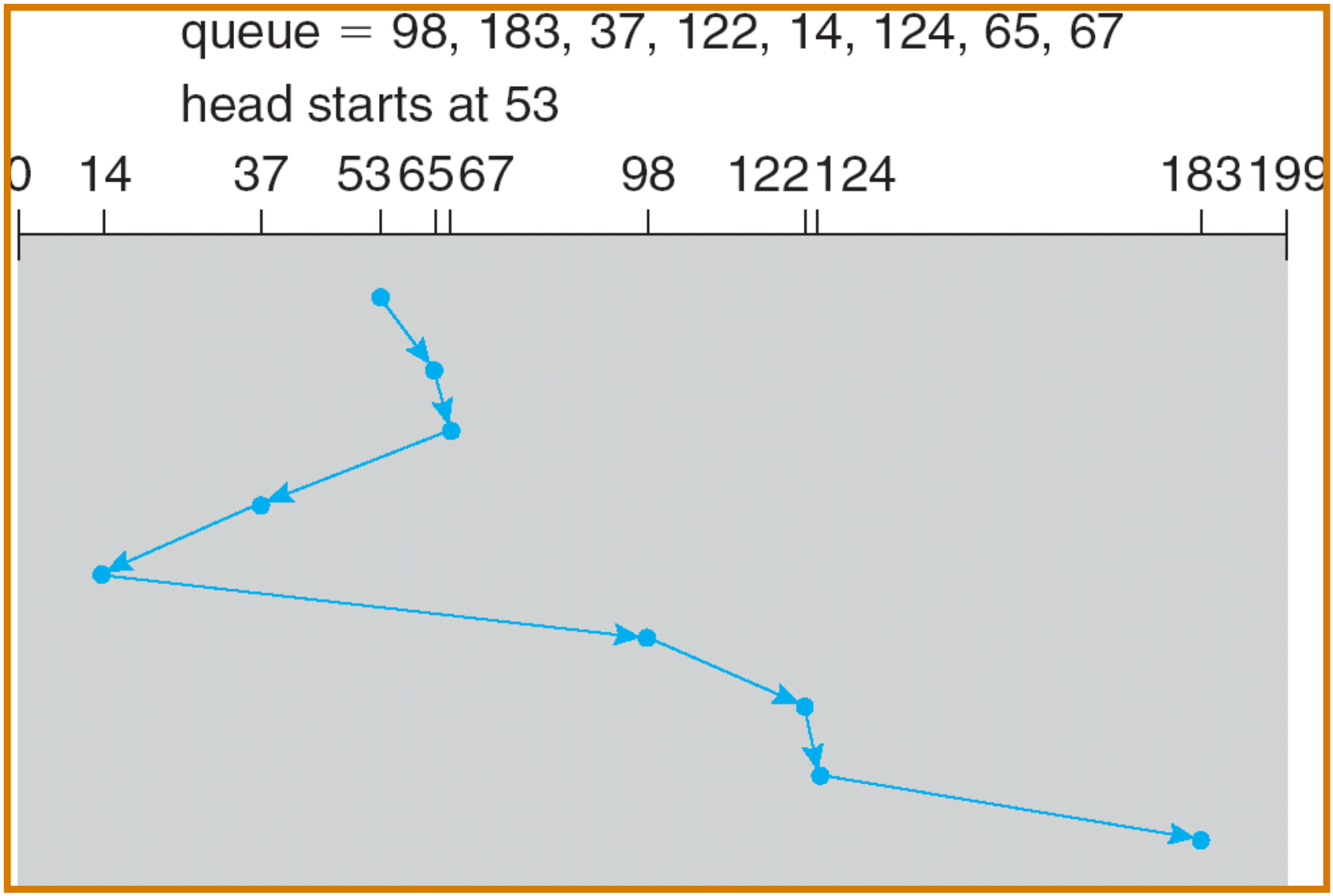
SSTF

- SSTF: Shortest Seek Time First Algorithm
- Selects the request with the minimum seek time from the current head position.
- Tries to minimize the seek time.

SSTF (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

SSTF (Cont.)



- Illustration shows total head movement of 236 cylinders.

SSTF (Cont.)

- Pros:
  - Tries to minimize the seek time

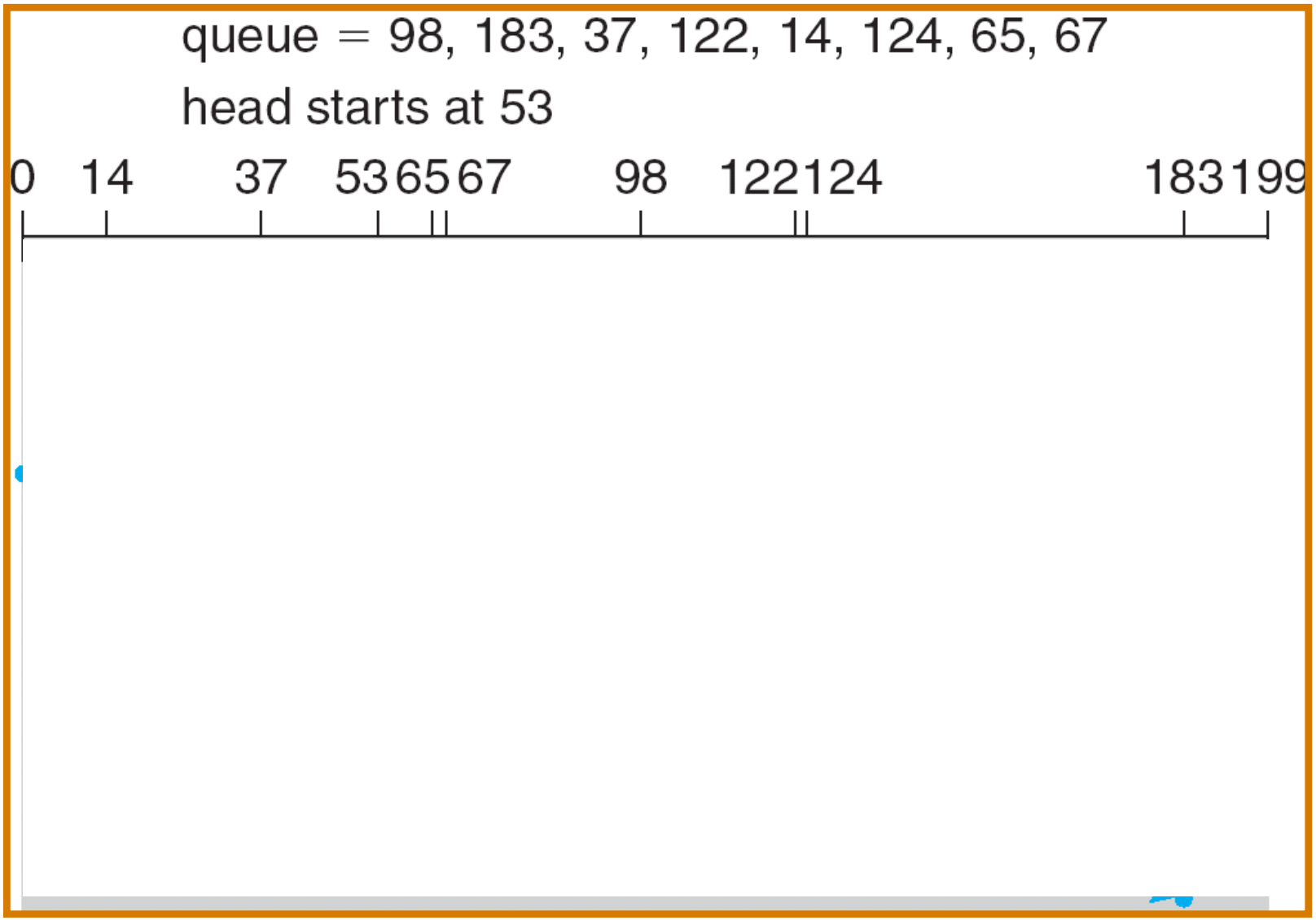
- Cons:
  - SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests.

SCAN

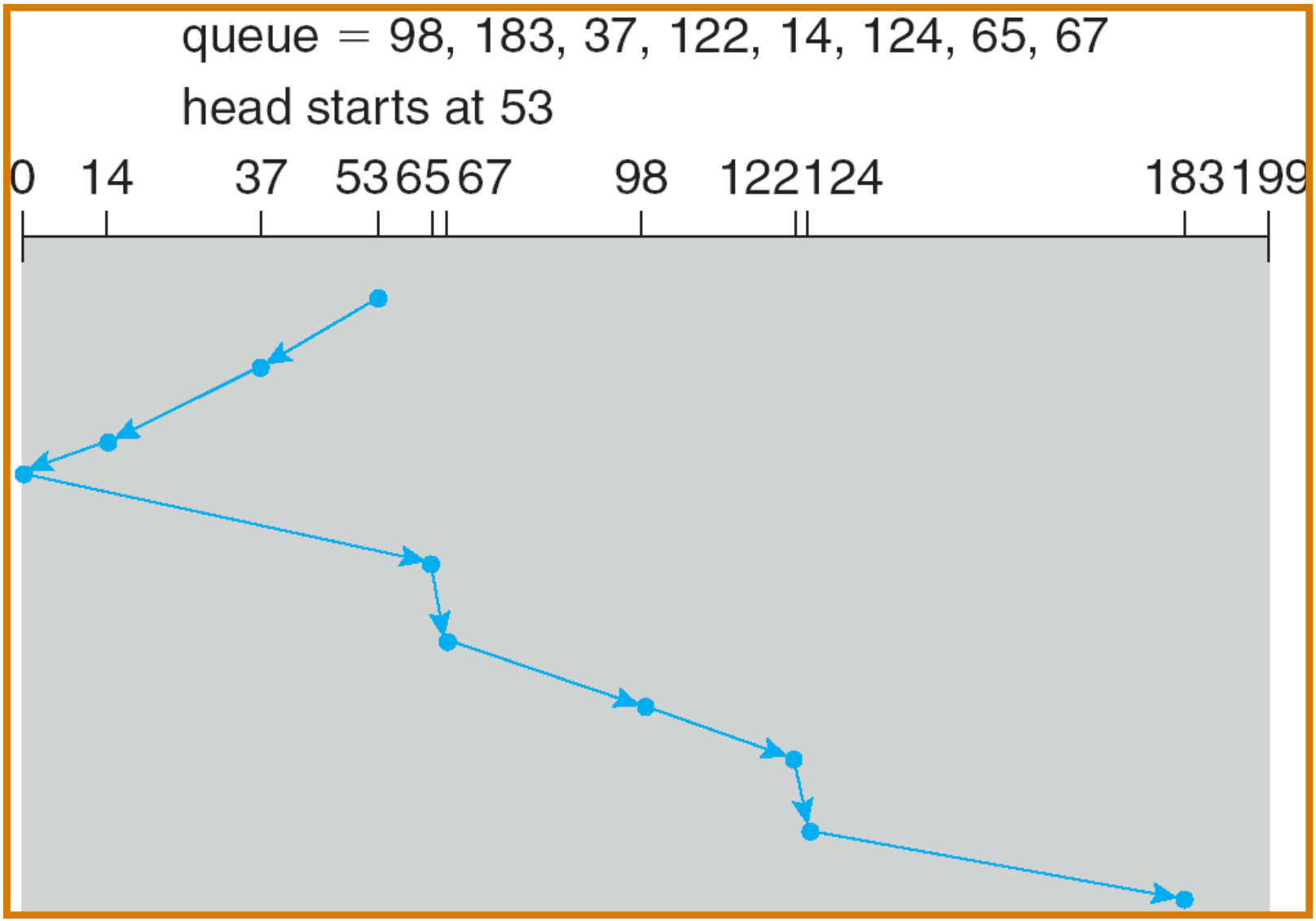
- The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.

- Sometimes called the ***elevator*** *algorithm*.

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 (Cont.)



- Illustration shows total head movement of 236 cylinders.

SCAN (Cont.)

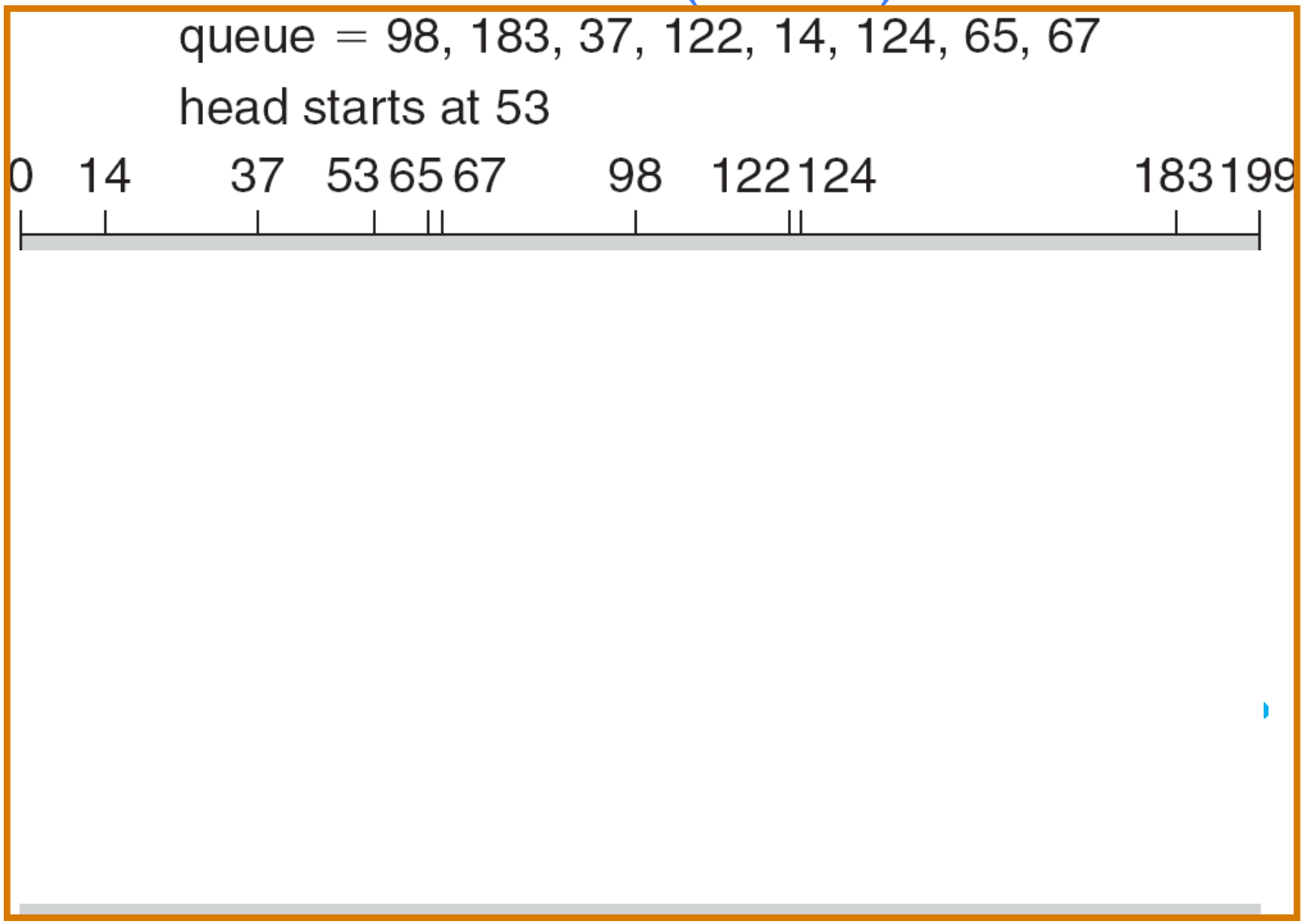
- Pros:
  - Bounded service time for each request
- Cons:
  - Request at the other end will take a while

C-SCAN

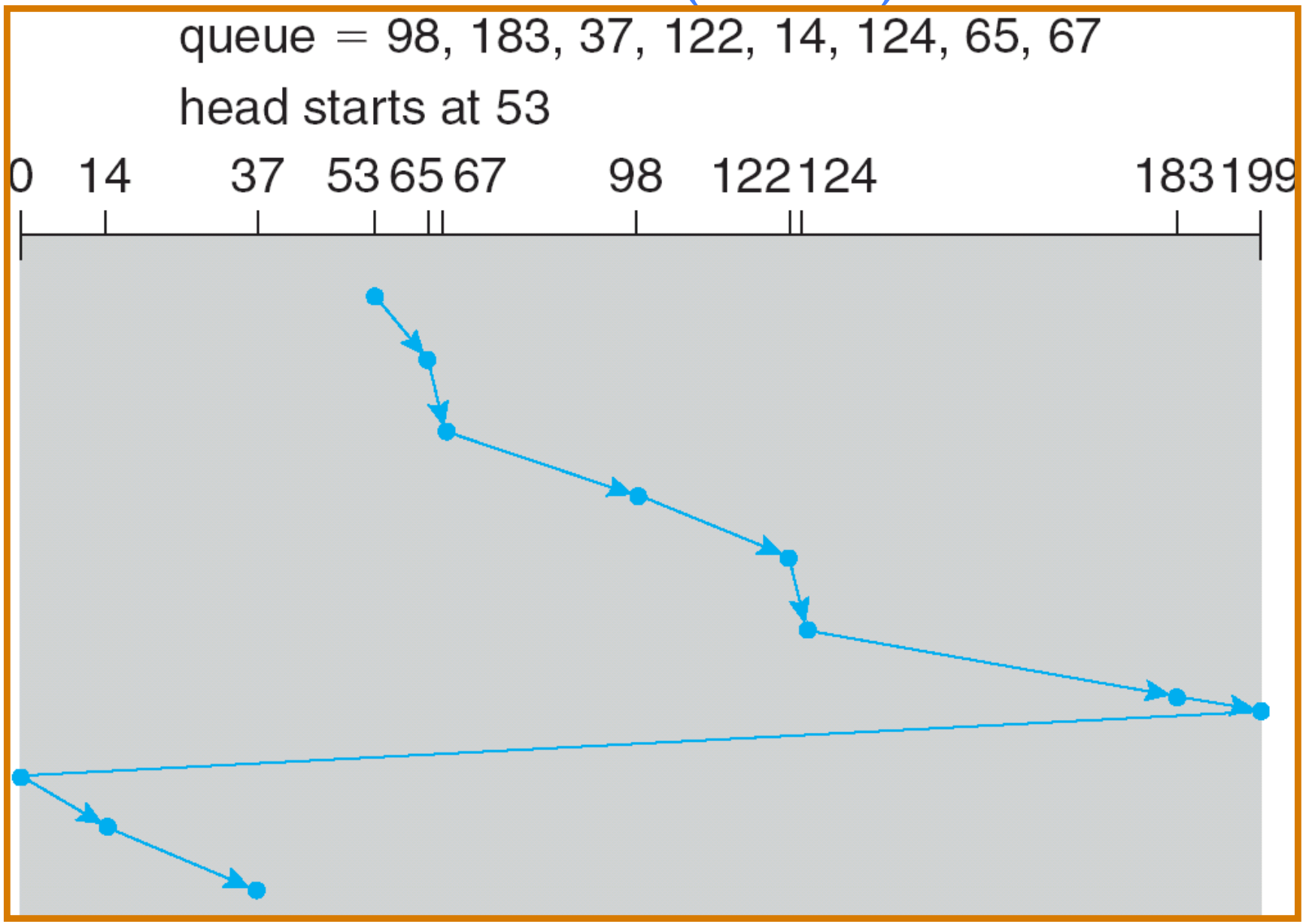
- The head moves from one end of the disk to the other. servicing requests as it goes. When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip.

- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one.

C-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

C-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

C-SCAN (Cont.)

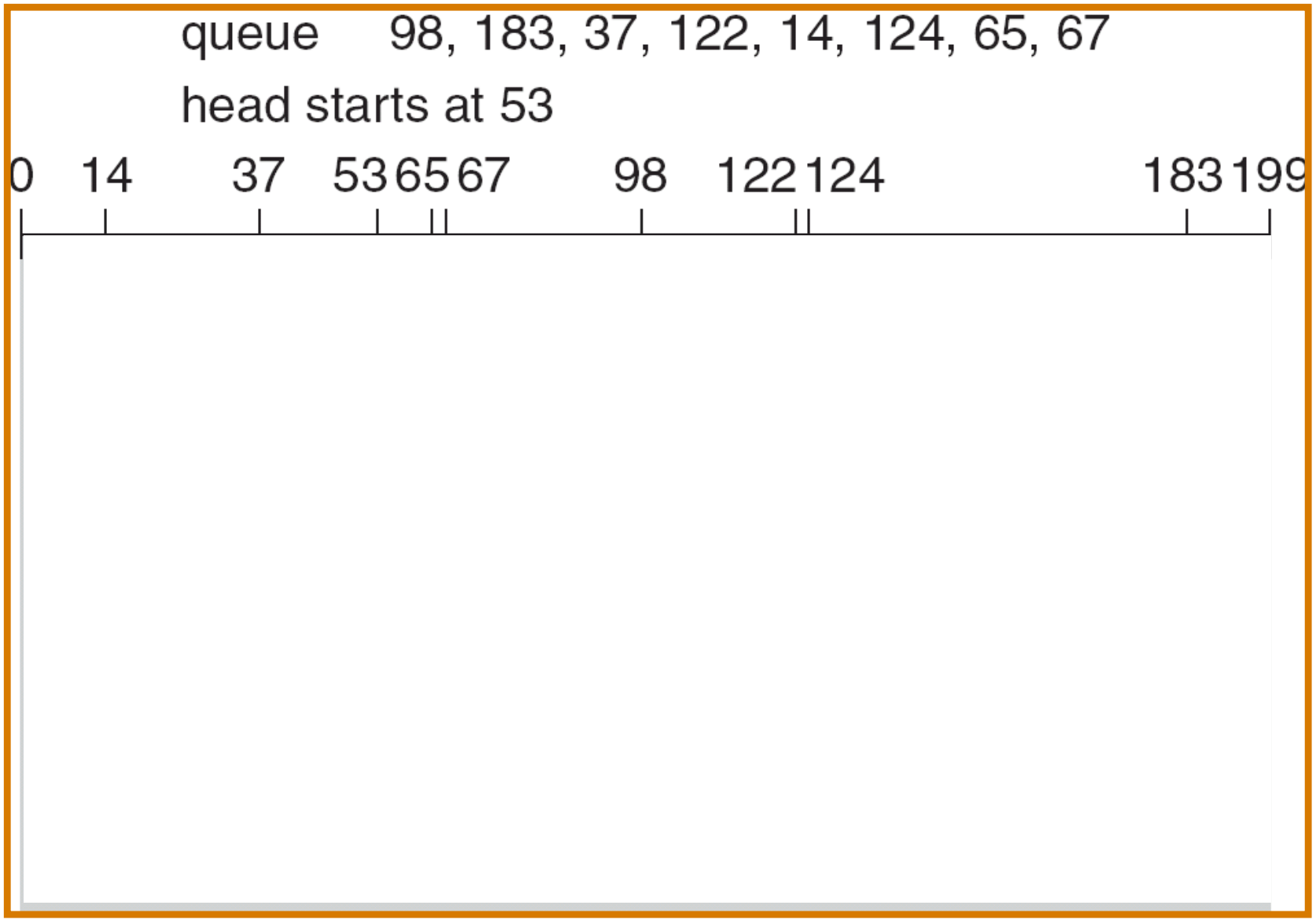
- Pros:
  - Provides a more uniform waiting time than SCAN
- Cons:
  - Does nothing on the return

LOOK

- Improved version of SCAN

- Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk.

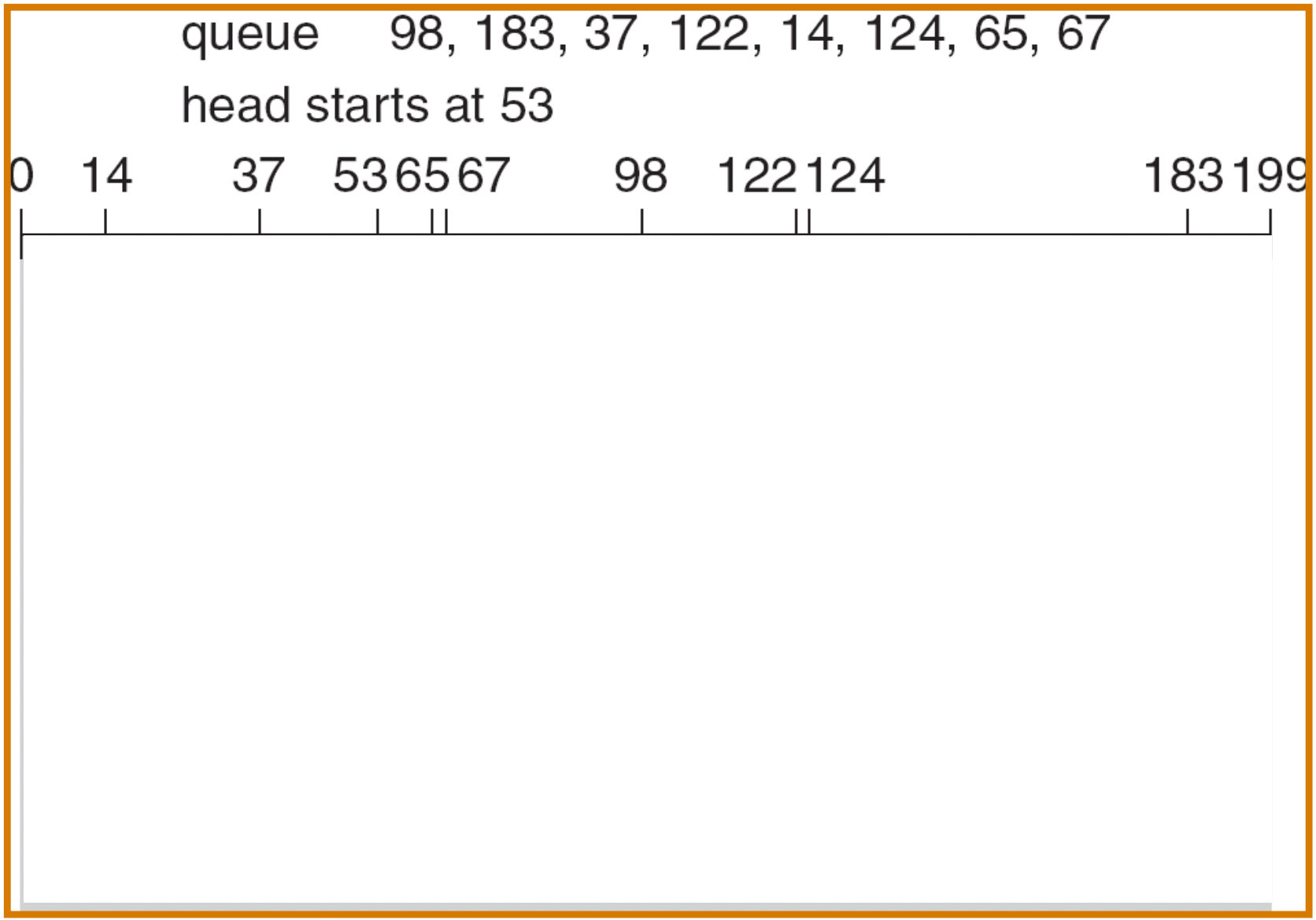
LOOK (Cont.)


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

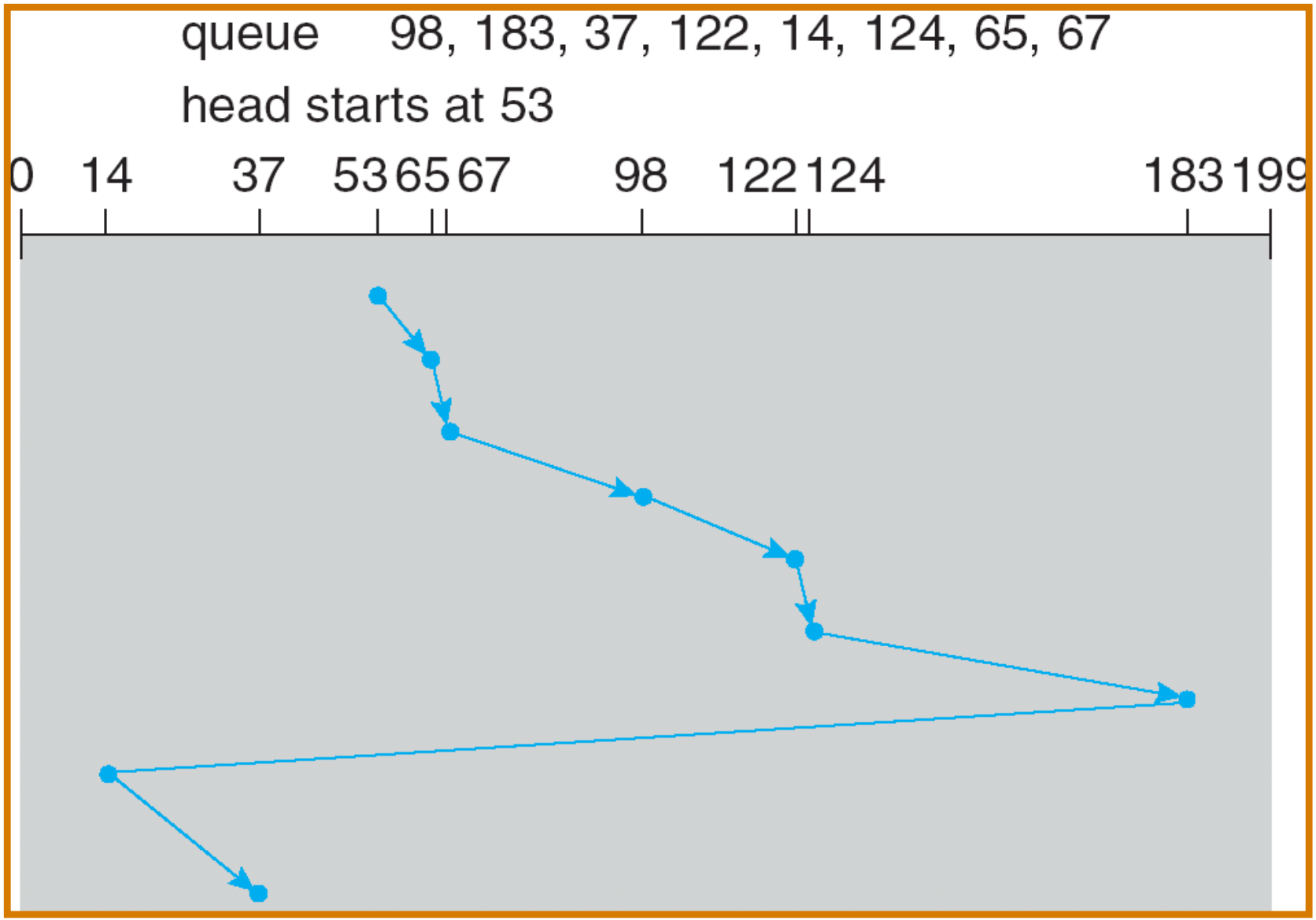
C-LOOK

- Improved version of C-SCAN
- Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk.

C-LOOK (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

C-LOOK (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

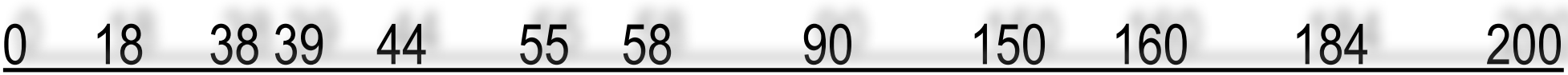
Exercise

- ✓ assume sequence of requested tracks in order received by disk scheduler: 55, 58, 39, 18, 90, 160, 150, 38, 184
- ✓ assume disk head initially located at track #100
- ✓ disk has moved to track # 100 from track #44
- ✓ there are 200 total tracks on disk

- ✓ a) Show disk scheduling for FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK
- ✓ b) Compute total head movements for each

Exercise

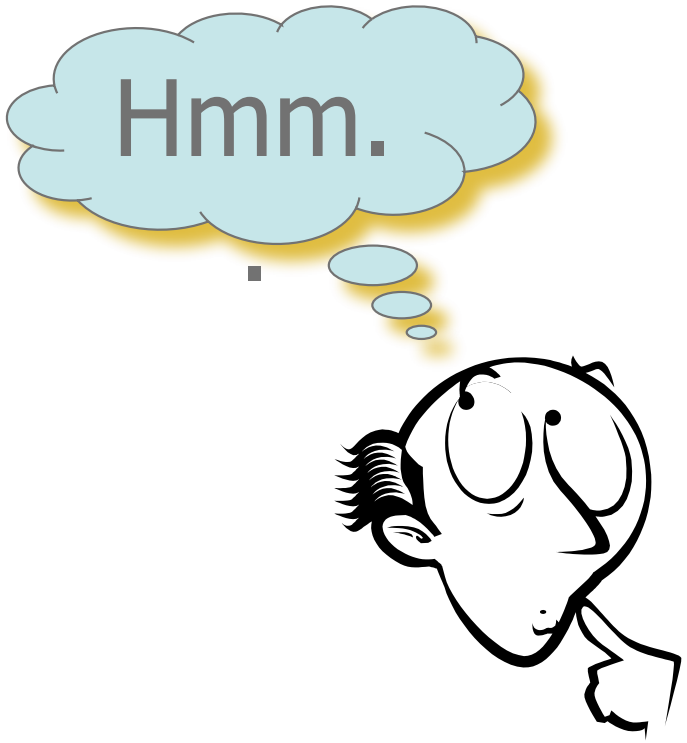
- ✓ assume sequence of requested tracks in order received by disk scheduler: 55, 58, 39, 18, 90, 160, 150, 38, 184



Selecting a Disk-Scheduling Algorithm

- FCFS is a fair algorithm, bu may perform very poorly.
- SSTF is common since increases performance over FCFS, but may cause starvation.
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk, prevent starvation.
- LOOK and C-LOOK optimize SCAN and S-SCAN further.
- 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.

Any Questions?



Acknowledgements

- “Operating Systems Concepts” book and supplementary material by A. Silberschatz, P. Galvin and G. Gagne

- “Operating Systems: Internals and Design Principles” book and supplementary material by W. Stallings

- “Modern Operating Systems” book and supplementary material by A. Tanenbaum

- Z. Shao from Yale; J.P. Singh from Princeton