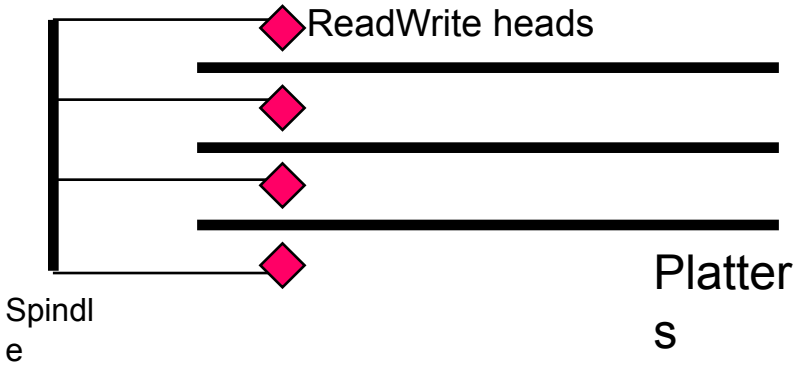
Mass Storage Management

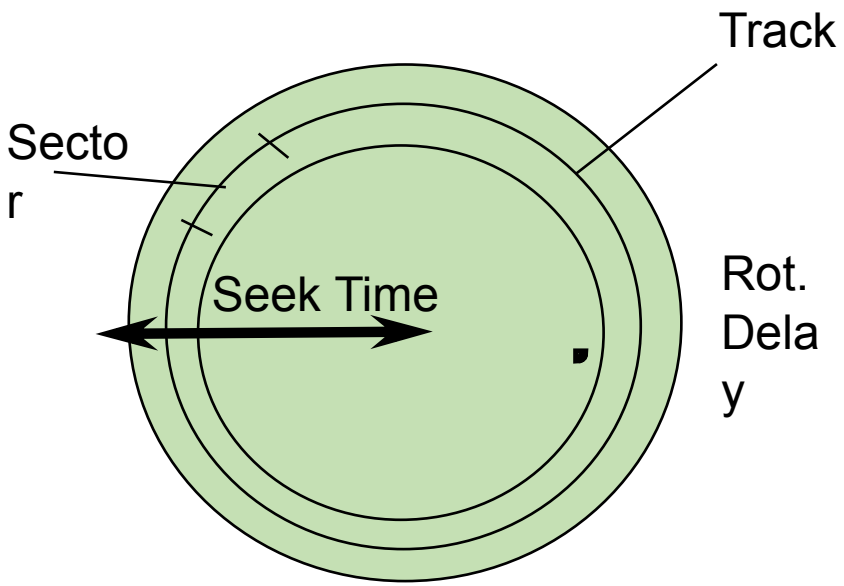
Dept. of Computer Science

Hanyang University

Physical Disk Structure

- Disk are made of
- thin metallic platters
- with a read/write head flying over it
- To read from disk, we must specify:
  - cylinder #
  - surface #
  - sector #
  - transfer size
  - memory address
- Transfer time includes
  - Seek time
  - Rotational delay
  - Transfer time





Disk Structure

- Disk drives are addressed as large

  1-dimensional arrays of *logical blocks*,

  where the logical block is the smallest unit of transfer

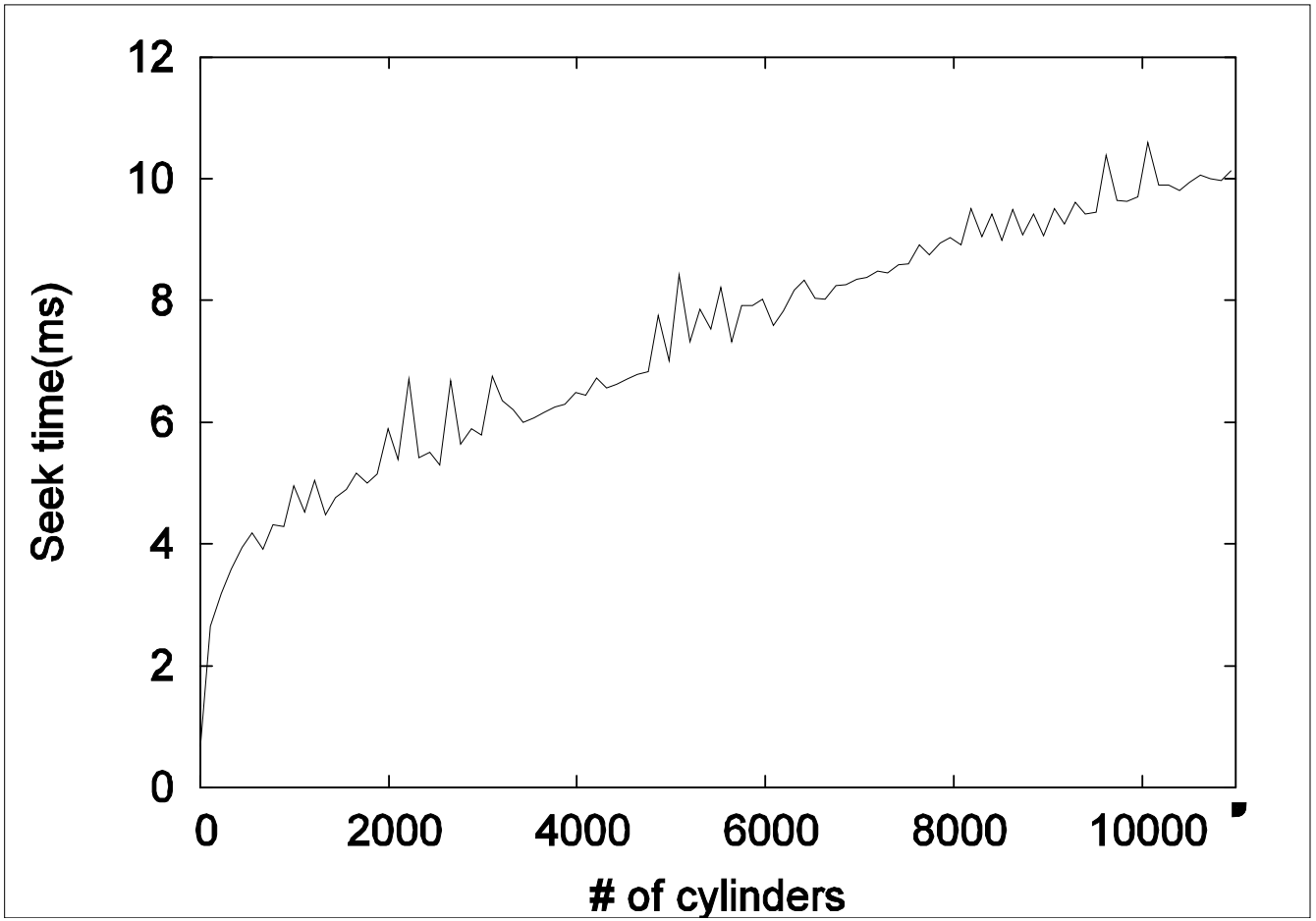
- The 1-dimensional array of logical blocks is mapped into the sectors of the disk
  - Sector 0 is the 1st sector of the 1st track on the outermost cylinder
  - Mapping generally proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost □ In reality, much different from product to product!

Disk Scheduling

- 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

$$\frac{\textit{the total number of bytes transferred}}{\textit{the total time between the request and the completion}}$$

Seek Time Benchmark



Seek time model: $a + b\sqrt{d}$ (a, b: c

Disk Scheduling (Cont.)

- Several algorithms exist to schedule the servicing of disk I/O requests
- We illustrate them with a request queue (0-199)

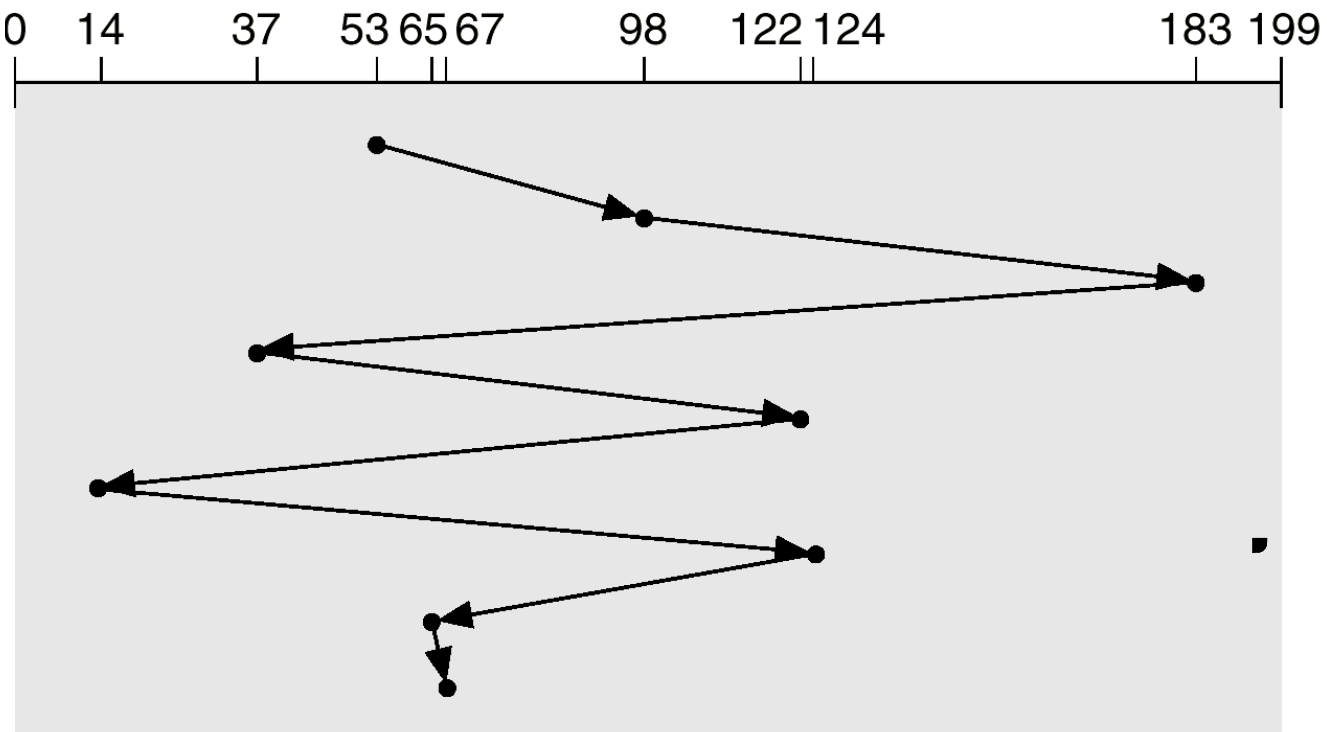
98, 183, 37, 122, 14, 124, 65, 67

Current head position: 53

FCFS

Illustration shows total head movement of 640 cylinders.

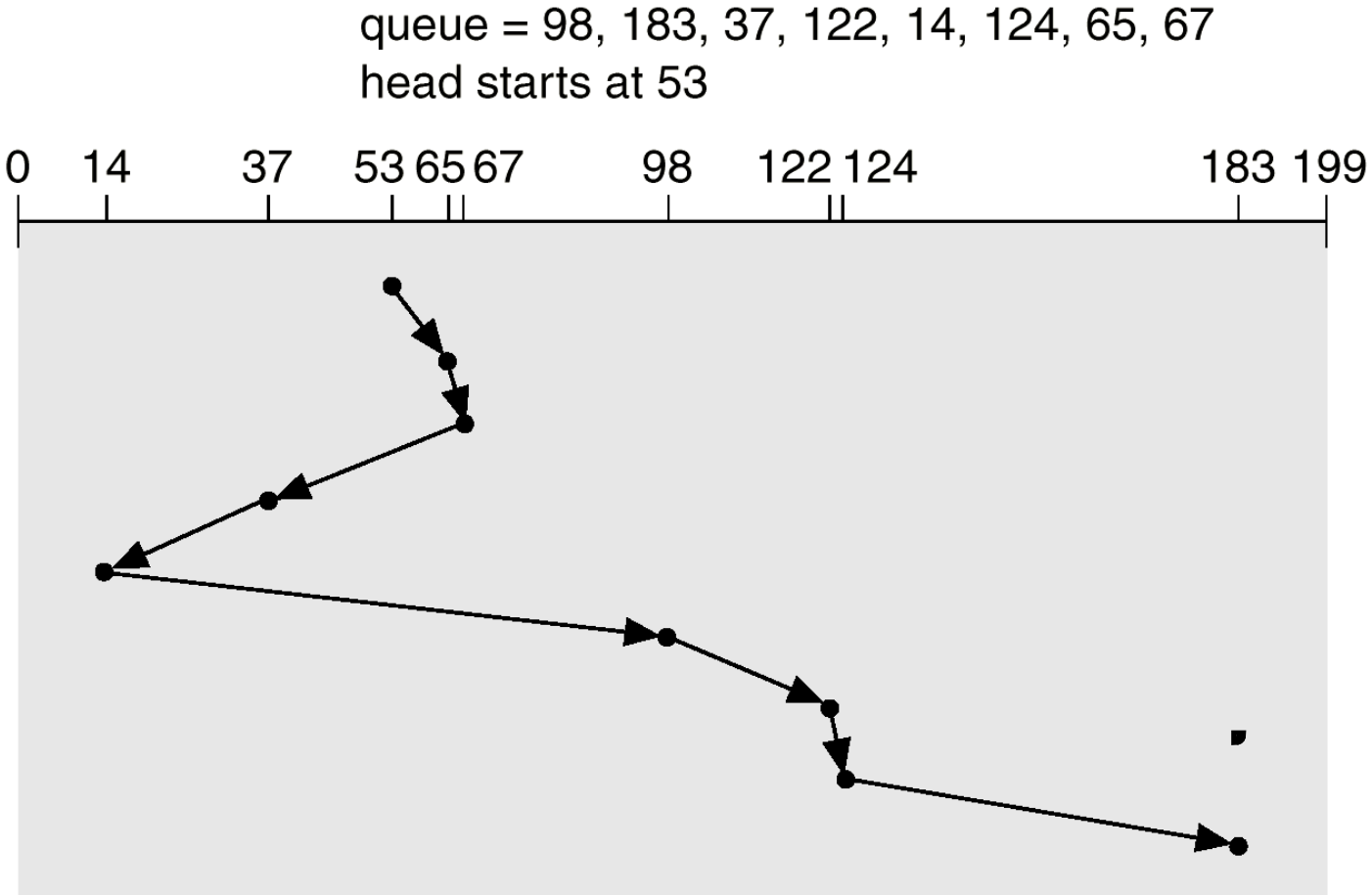
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



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
- unpredictable performance
- Illustration shows total head movement of 236 cylinders

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

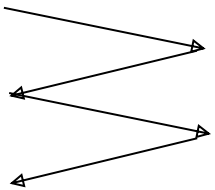
SCAN

- 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*
  - First, service all requests while going up
  - Then service all requests while going down

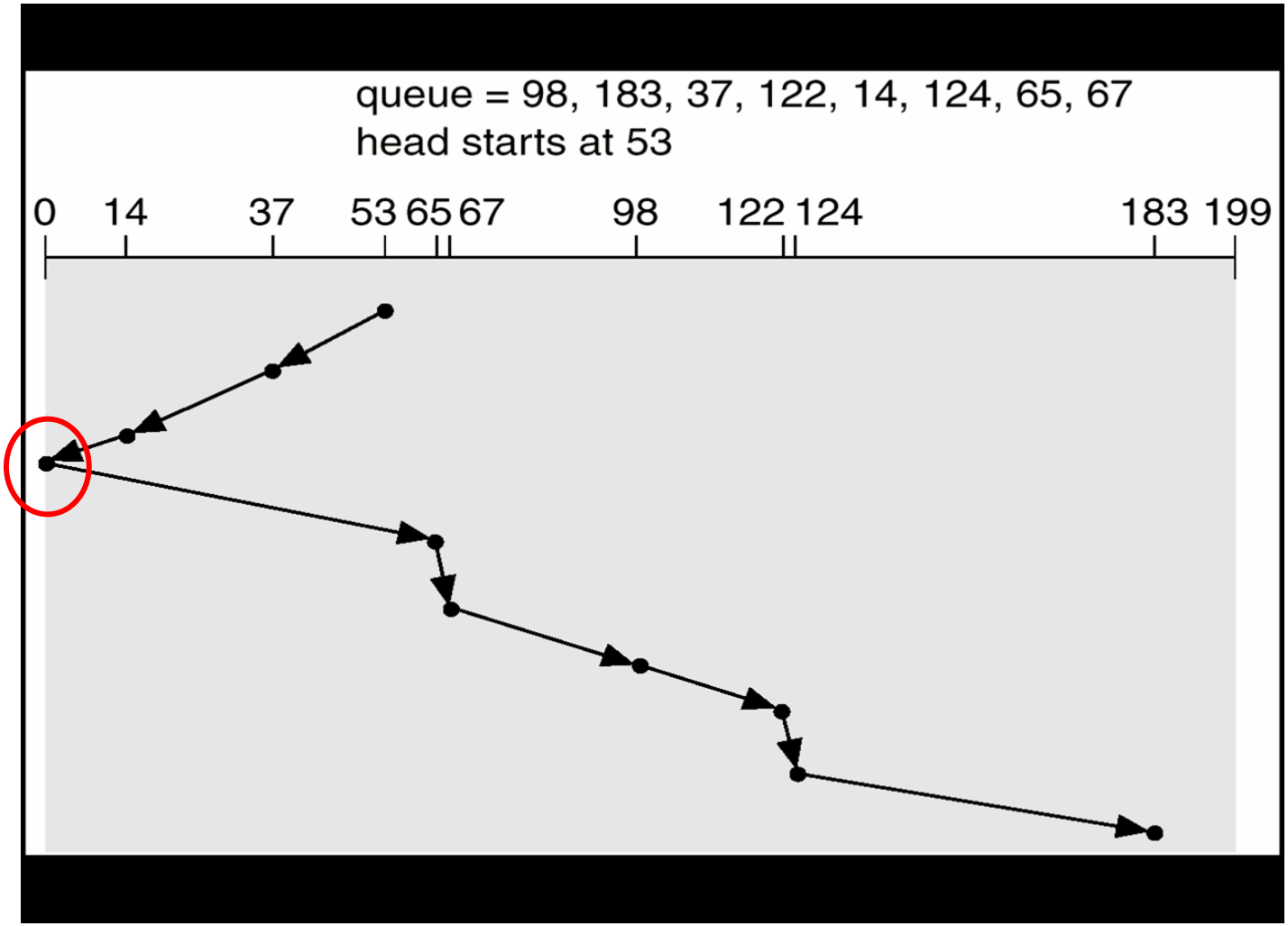


- Illustration shows total head movement of 208 cylinders

- Problem
- Two times of services for inner tracks for each service for outer(inner)most tracks

  □ Unfair wait time

SCAN (Cont.)

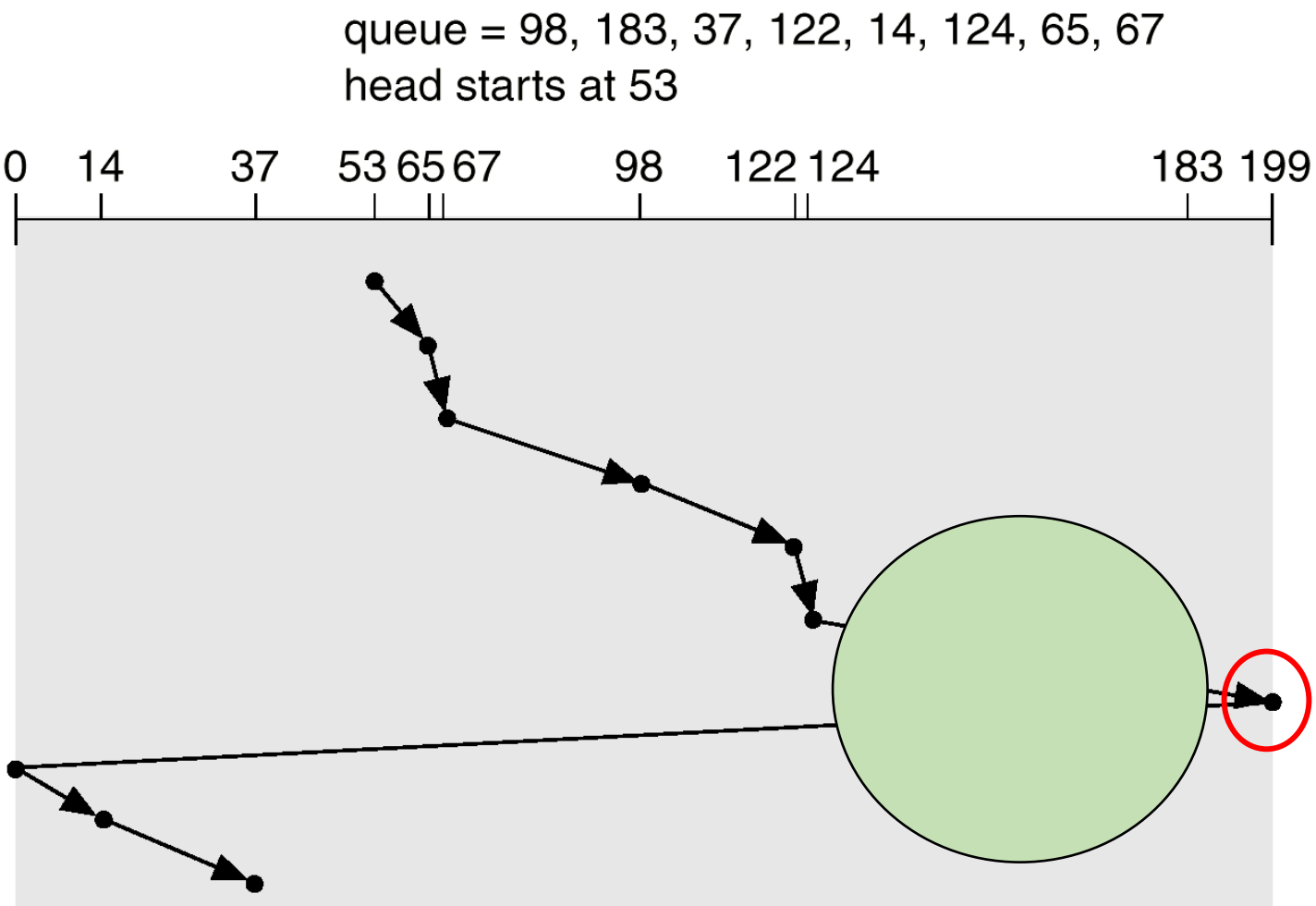

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

C-SCAN

- 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

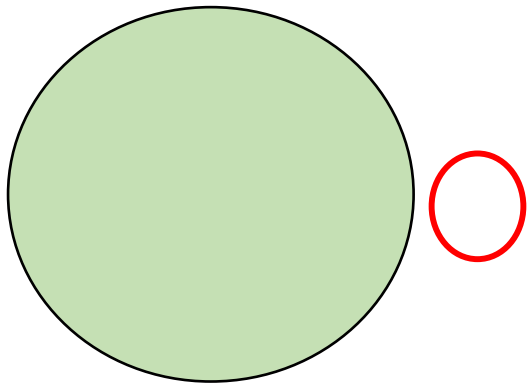
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one
- Problem
- Keep moving forward despite no pending request exists in that direction

C-SCAN (Cont.)

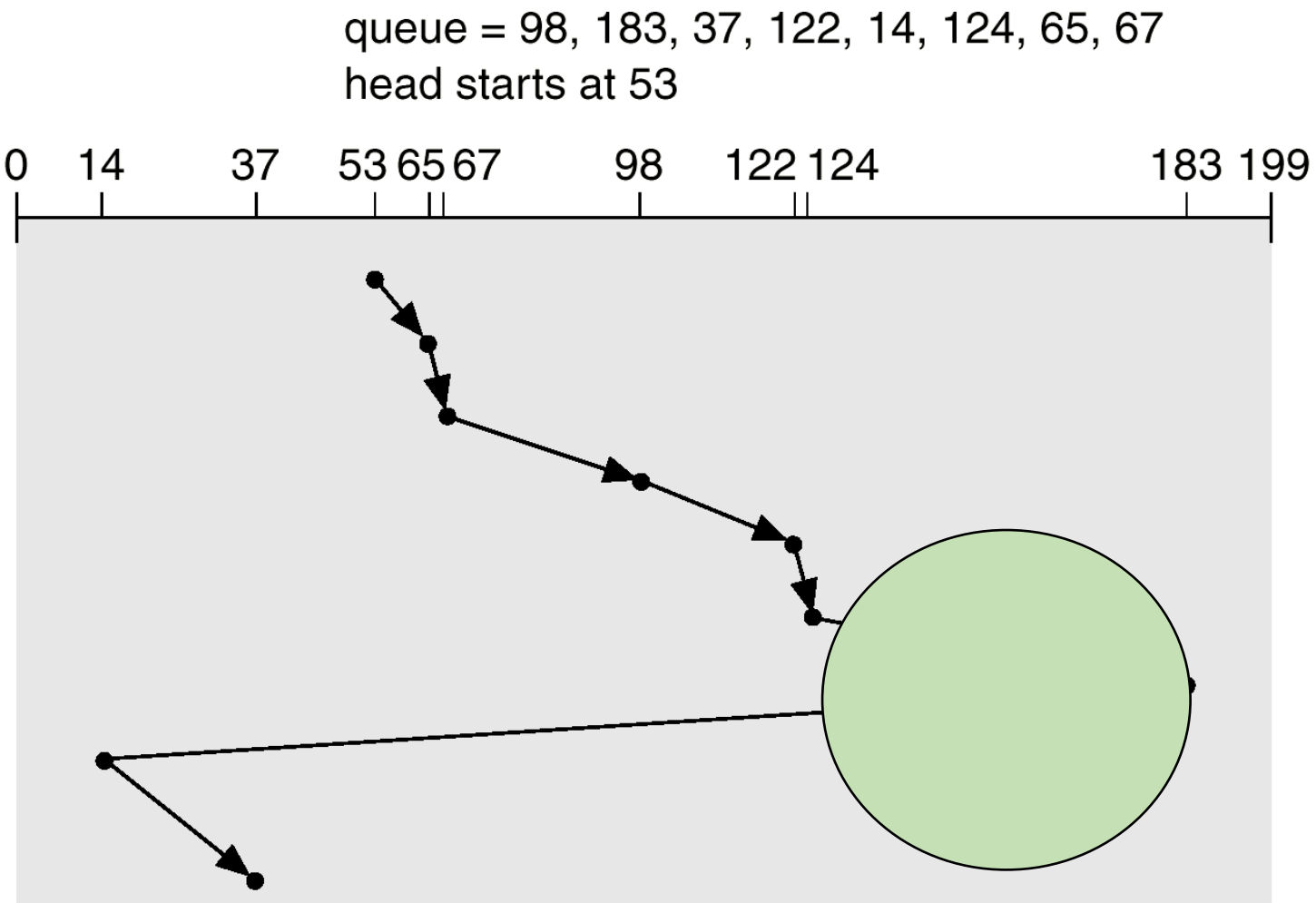

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

C-LOOK

- Version of C-SCAN
- Look-for-Request before continuing
- 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.

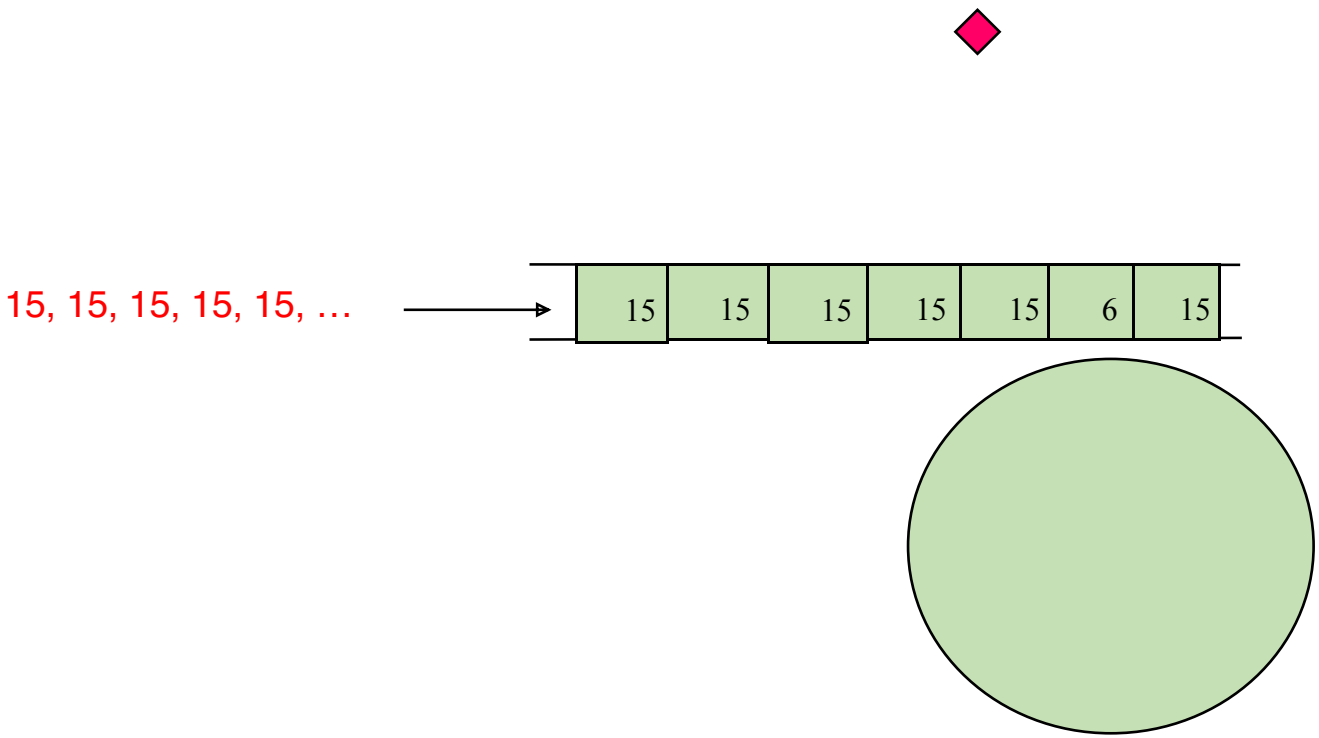


C-LOOK (Cont.)


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

Arm Stickiness Problem

- SCAN, C-SCAN, SSTF may suffer from ‘Arm Stickiness Problem’
  - When one or a few processes have high access rates to one track
  - They can monopolize the entire device by repeating requests to that track



Disk Management

- Physical formatting or Low-level formatting
- Dividing a disk into sectors that the disk controller can R/W
- Each sector = [head + data(512 Bytes, in general) + trailer]
- header/trailer contain sector number, ECC (used by controller)
- spare sectors/cylinder are reserved for bad blocks
- To use a disk to hold files, the operating system still needs to record its own data structures on the disk
  - Partition the disk into one or more groups of cylinders: OS treats each partition as an independent disk
  - Logical formatting or "making a file system": Initialize data structure for file system, used by OS
- Power-up
  - The "small *bootstrap loader*" runs from ROM
  - Which is very small, only duty is to load sector 0 (boot block) of boot disk

RAID

- RAID: Redundant Array of Inexpensive (Independent) Disks
  - Disk organization techniques that manage a large numbers of disks, providing a view of a single disk of
    - high capacity and high speed by using multiple disks in parallel, and
    - high reliability by storing data redundantly, so that data can be recovered even if a disk fails
- The chance that some disk out of a set of *N* disks will fail is much higher than the chance that a specific single disk will fail
  - E.g., a system with 100 disks, each with MTTF of 100,000 hours (approx. 11 years), will have a system MTTF of 1000 hours (approx. 41 days)
  - Techniques for using redundancy to avoid data loss are critical with large numbers of disks

Improvement of Reliability via Redundancy

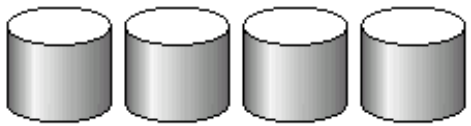
- Redundancy – store extra information that can be used to rebuild information lost in a disk failure
- E.g., Mirroring (or shadowing)
  - Duplicate every disk: Logical disk consists of two physical disks
  - Every write is carried out on both disks
    - Reads can take place from either disk
  - If one disk in a pair fails, data still available in the other
    - Data loss would occur only if a disk fails, and its mirror disk also fails before the system is repaired
      - Probability of combined event is very small, except for dependent failure modes such as fire or building collapse or electrical power surges
- Mean time to data loss depends on mean time to failure, and mean time to repair
  - E.g. MTTF of 100,000 hours, mean time to repair of 10 hours gives mean time to data loss of 500*106 hours (or 57,000 years) for a mirrored pair of disks (ignoring dependent failure modes)

Improvement in Performance via Parallelism

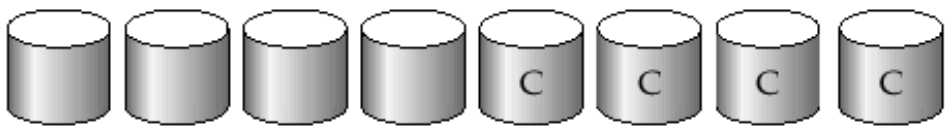
- Two main goals of parallelism in a disk system:
  1. Load balance multiple small accesses to increase throughput
  2. Parallelize large accesses to reduce response time: Improve transfer rate by striping data across multiple disks
- Bit-level striping
  - Split the bits of each byte across multiple disks
  - In an array of eight disks, write bit *i* of each byte to disk *i*
  - Each access can read data at eight times the rate of a single disk
  - But seek/access time worse than for a single disk
    - Bit level striping is not used much any more
- Block-level striping
  - With *n* disks, block *i* of a file goes to disk (*i* mod *n*) + 1
  - Requests for different blocks can run in parallel if the blocks reside on different disks
  - A request for a long sequence of blocks can utilize all disks in parallel

RAID Levels

- Schemes to provide redundancy at lower cost by using disk striping combined with parity bits
  - Different RAID organizations, or RAID levels, have differing cost, performance and reliability characteristics
- RAID Level 0: Block striping, non-redundancy
  - Used in high-performance applications where data loss is not critical
- RAID Level 1: Mirrored disks with block striping
  - Offers best write performance
  - Popular for applications such as storing log files in a database system



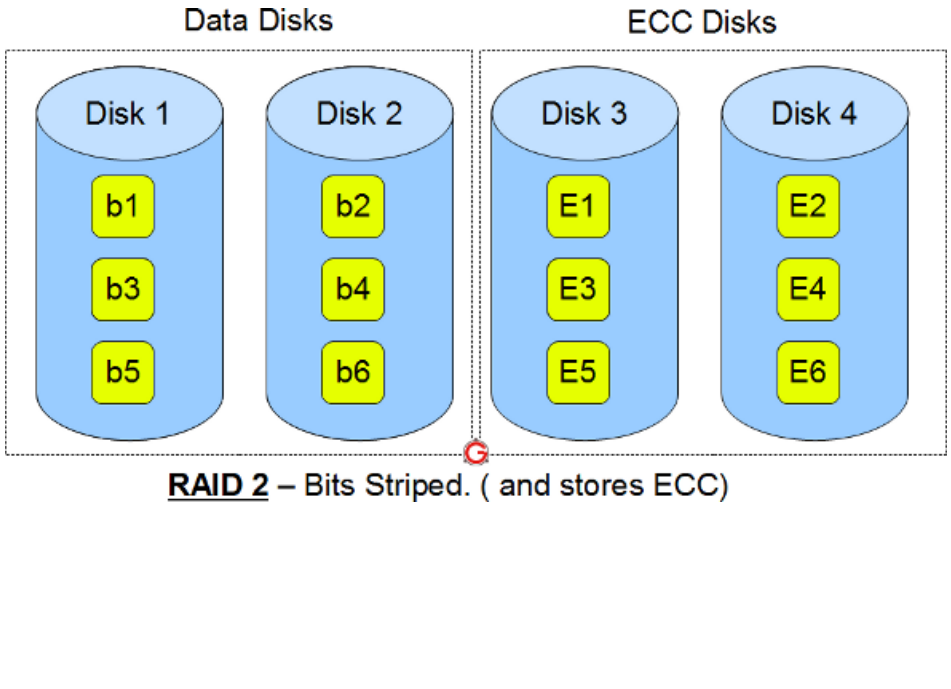
(a) RAID 0: nonredundant striping



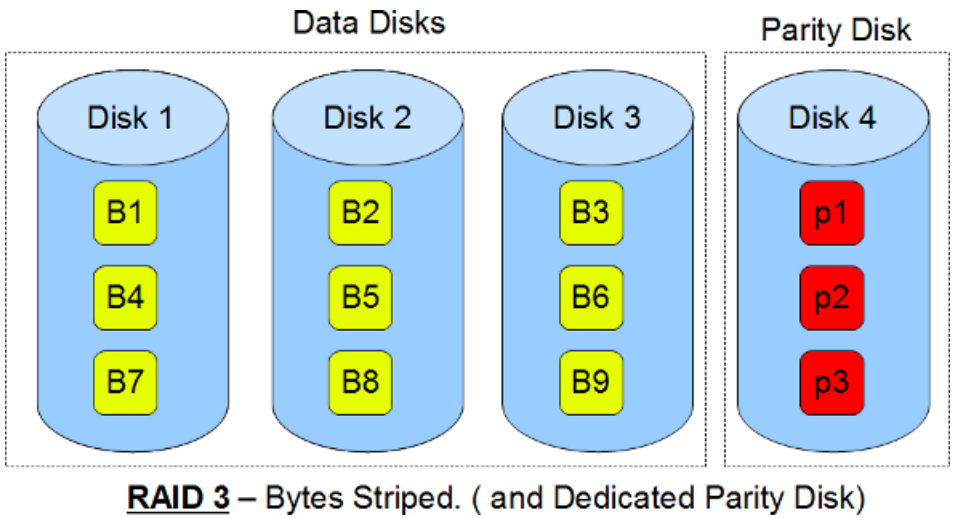
(b) RAID 1: mirrored disks

RAID Levels (Cont.)

- RAID Level 2: Bit-level striping with ECC disks
- RAID Level 3: Byte-level striping with a dedicated parity disk



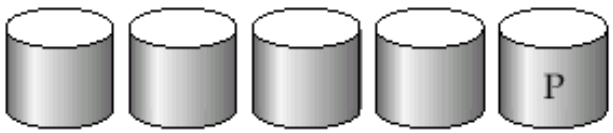
RAID 2 – Bits Striped. (and stores ECC)



RAID 3 – Bytes Striped. (and Dedicated Parity Disk)

RAID Levels (Cont.)

- RAID Level 4: Block-interleaved parity, block-level striping
  - Keeps a parity block on a separate disk for corresponding blocks from *N* other disks
  - When writing data block, corresponding block of parity bits must also be computed and written to parity disk
  - To find value of a damaged block, compute XOR of bits from corresponding blocks (including parity block) from other disks
  - Parity block becomes a bottleneck for independent block writes since every block write also writes to parity disk



(e) RAID 4: block-interleaved parity

RAID Levels (Cont.)

- RAID Level 5: Block-interleaved distributed parity, block-level striping
  - Partitions data and parity among all ~~*N* + 1 disks~~
  - E.g., with 5 disks, parity block for *n*-~~th set of blo~~cks is stored on disk (*n* *mod* 5) + 1, with the data blocks stored on the other 4 disks

(f) RAID 5: block-interleaved distributed parity

P0	0	1	2	3
4	P1	5	6	7
8	9	P2	10	11
12	13	14	P3	15
16	17	18	19	P4

Choice of RAID Level

- Factors in choosing RAID level
  - Monetary cost
  - Performance: Number of I/O operations per second, and bandwidth during normal operation
  - Performance during failure
  - Performance during rebuild of failed disk
- RAID 0 is used only when data safety is not important
  - E.g. data can be recovered quickly from other sources
- Level 2 and 3 are not used anymore since bit/byte-striping forces single block reads to access all disks, wasting disk arm movement, which block striping (level 4, 5) avoids
- Level 4 never used since it is subsumed by level 5
- So competition is between 1 and 5 only

Choice of RAID Level (Cont.)

- Level 1 provides much better write performance than level 5
  - Level 5 requires at least 2 block reads and 2 block writes to write a single block, whereas Level 1 only requires 2 block writes
  - Level 1 preferred for high update environments such as log disks
- Level 1 had higher storage cost than level 5
  - Disk drive capacities increasing rapidly (50%/year) whereas disk access times have decreased much less (x 3 in 10 years)
  - I/O requirements have increased greatly, e.g. for Web servers
  - When enough disks have been bought to satisfy required rate of I/O, they often have spare storage capacity
    - So there is often no extra monetary cost for Level 1!
- Level 5 is preferred for applications with low update rate, and large amounts of data
- Level 1 is preferred for all other applications