



OPERATING SYSTEMS

Storage Management - 4

Nitin V Pujari

Faculty, Computer Science

Dean - IQAC, PES University



OPERATING SYSTEMS

Storage Management - 4: Mass Storage Structure

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 3



Unit 4: Storage Management

Mass-Storage Structure - Mass-Storage overview, Disk Scheduling, Swap-Space Management, RAID structure. File System Interface - file organization/structure and access methods, directories, sharing File System Implementation/Internals: File control Block (inode), partitions & mounting, Allocation methods.

Case Study: Linux/Windows File Systems

OPERATING SYSTEMS

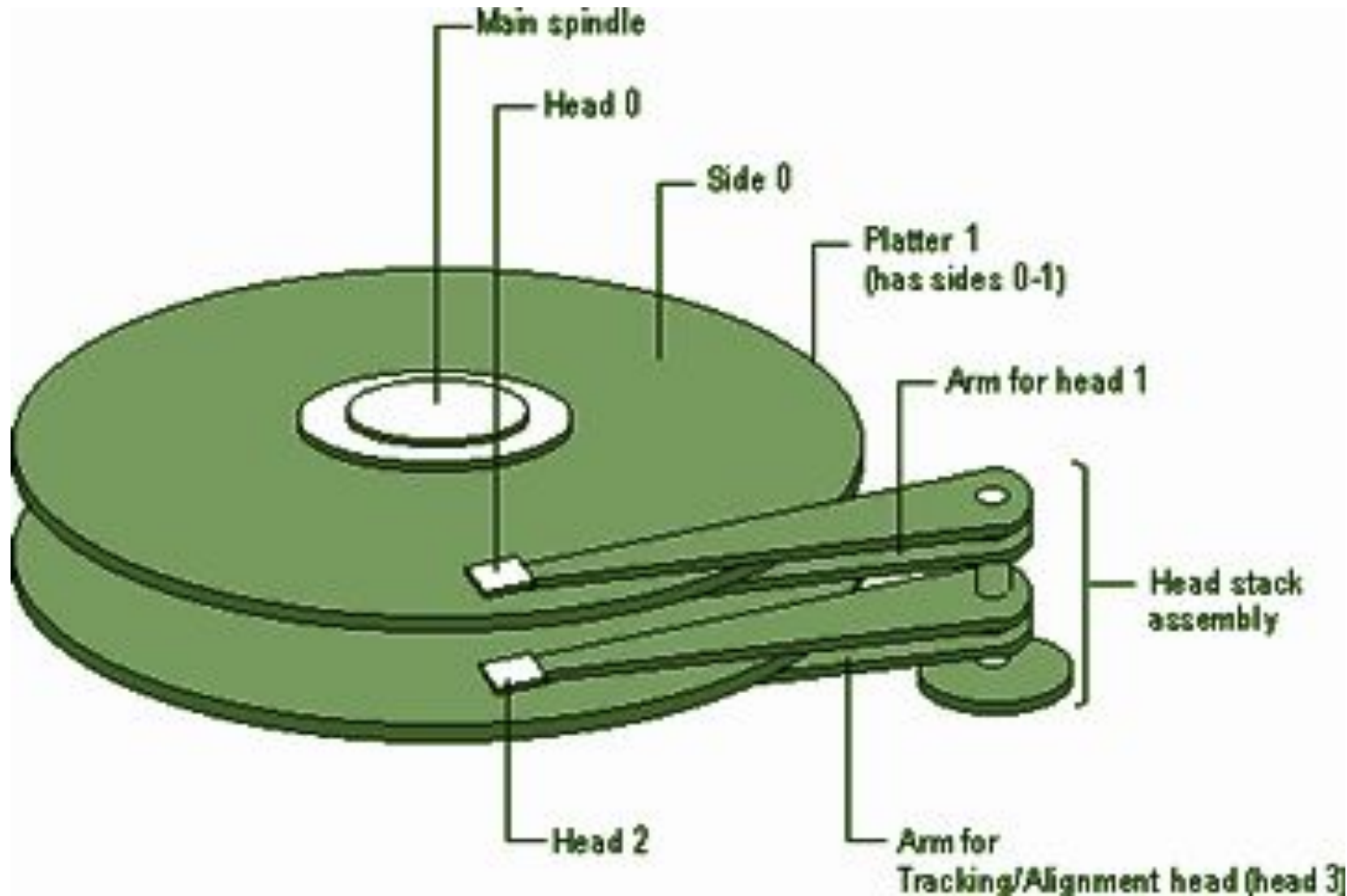
Course Outline

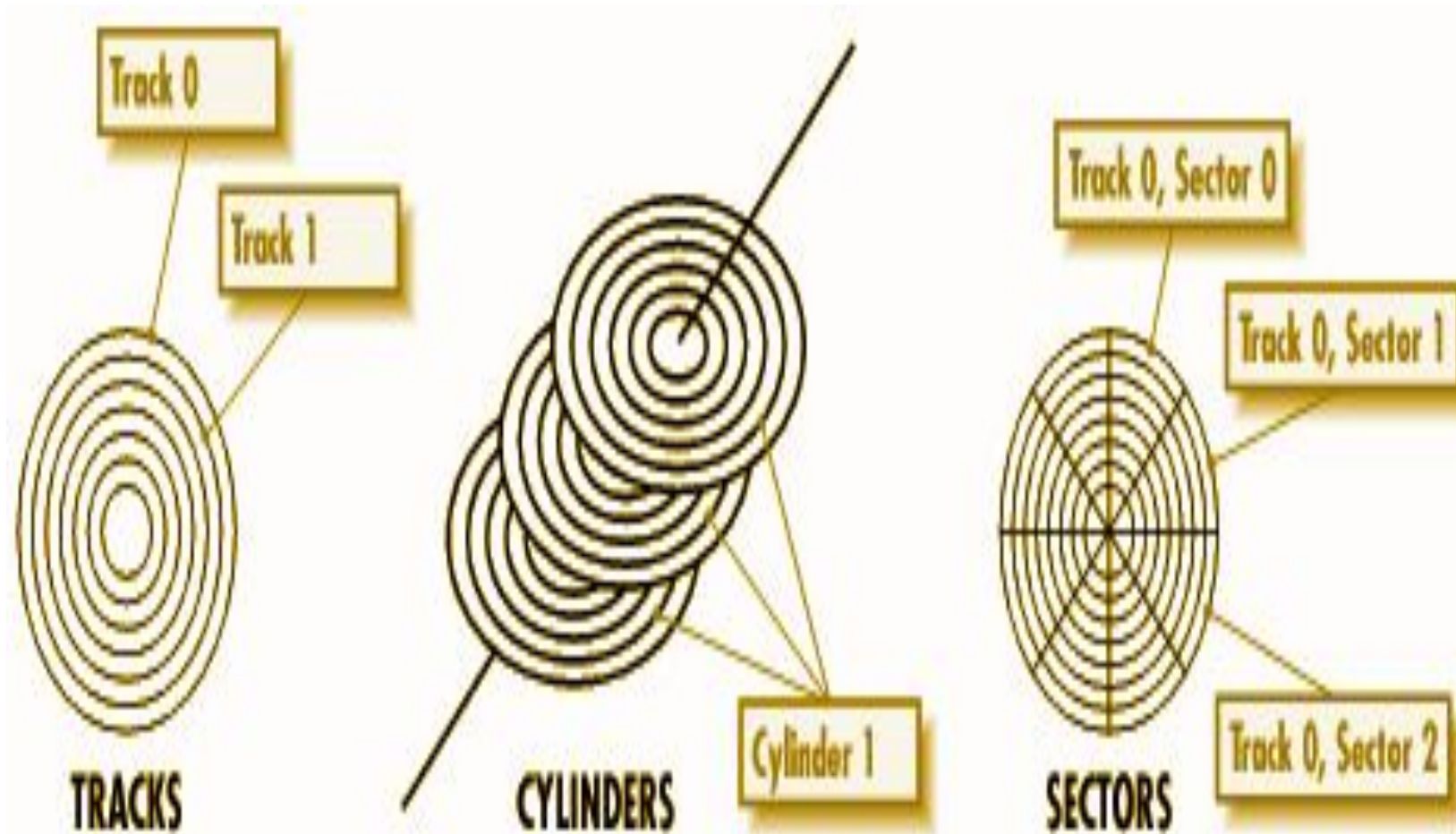


37	Mass-Storage Structure: Mass-Storage overview	12.1	82.1
38	Disk Scheduling - FCFS, SSTF, SCAN, C-SCAN, LOOK	12.4	
39	Swap-Space Management, RAID Structure	12.6,12.7	
40	File Concept, File Structure, Access Methods	10.1-10.2	
41	Directory and Disk Structure	10.3	
42	File-System Mounting, File Sharing, Protecting	10.4-10.6	
43	Implementing File-Systems: File control Block (inode), partitions & mounting	11.1,11.2	
44	Disk Space Allocation methods: Contiguous, Linked, Indexed	11.4	
45	Case Study: Unix/Linux File systems	11.8	
46	NFS	16.7	

- SCAN Disk Scheduling
- C-SCAN Disk Scheduling
- LOOK Disc Scheduling
- C-LOOK Disc Scheduling

Disk Scheduling





Disk Scheduling

- I/O request includes input or output mode, disk address, memory address, number of sectors to transfer
- OS maintains queue of requests, per disk or device
- Idle disk can immediately work on I/O request, busy disk means work must queue
 - Optimization algorithms only make sense when a queue exists
- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying “depth”)
- Several algorithms exist to schedule the servicing of disk I/O requests
- The analysis is true for one or many platters
- Seek Distance is in terms of **Cylinders**

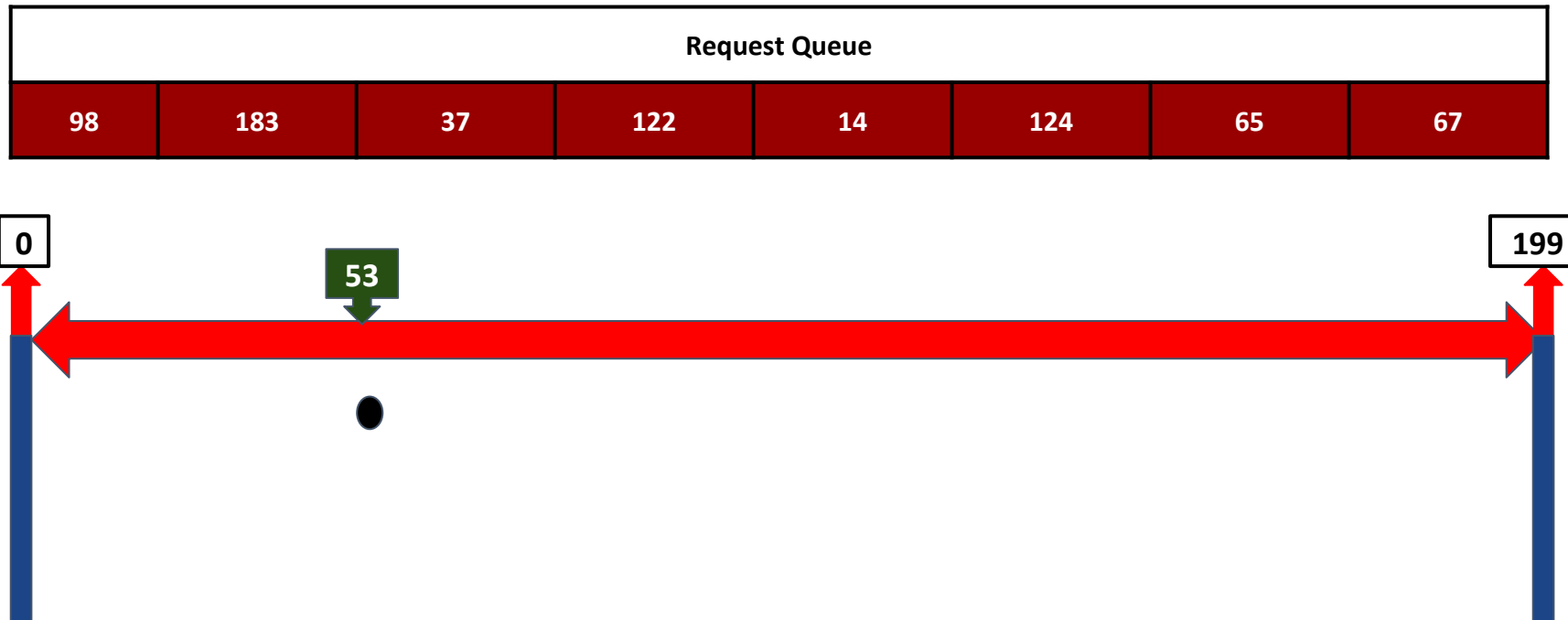
SCAN Disk Scheduling

- 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 is sometimes also called the **elevator algorithm**
- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest

- We illustrate scheduling algorithms with a request queue (0-199)
- **98, 183, 37, 122, 14, 124, 65, 67**
 - Head pointer 53

SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 0

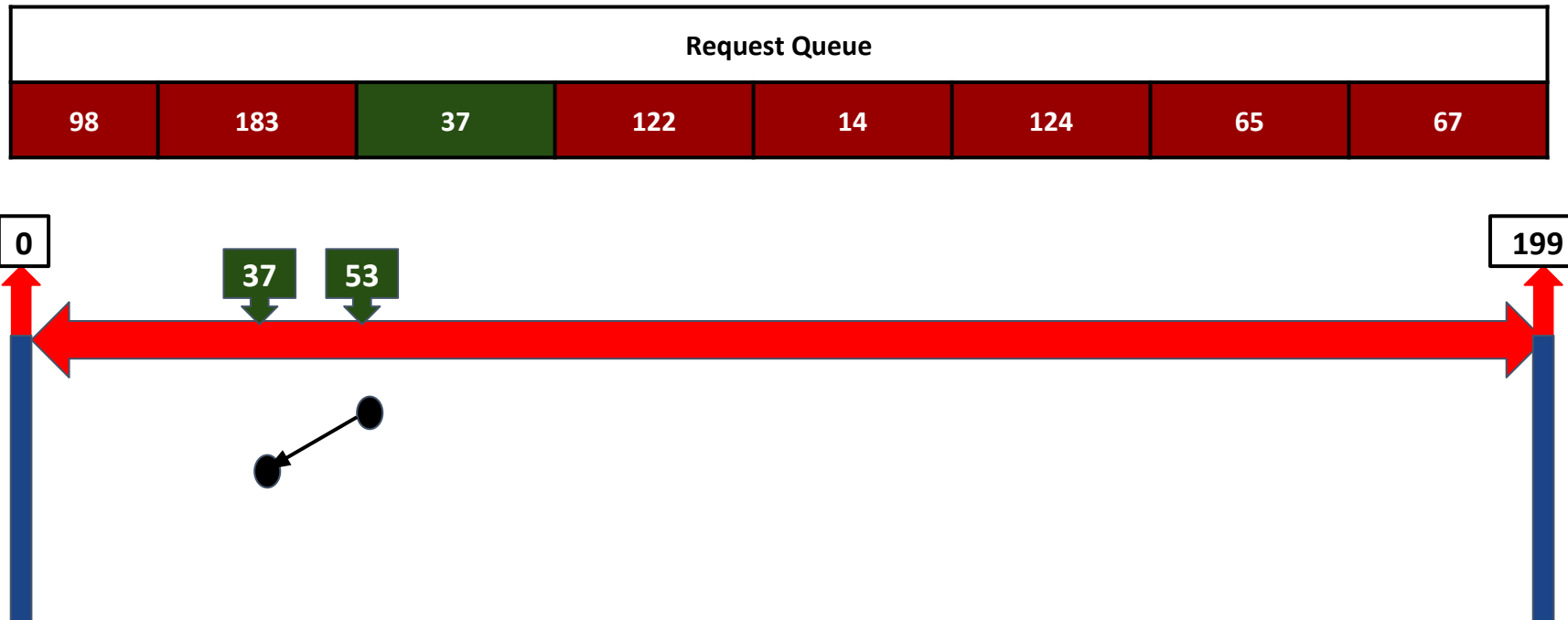
Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 0 + abs()

Seek Distance = 0

SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 0

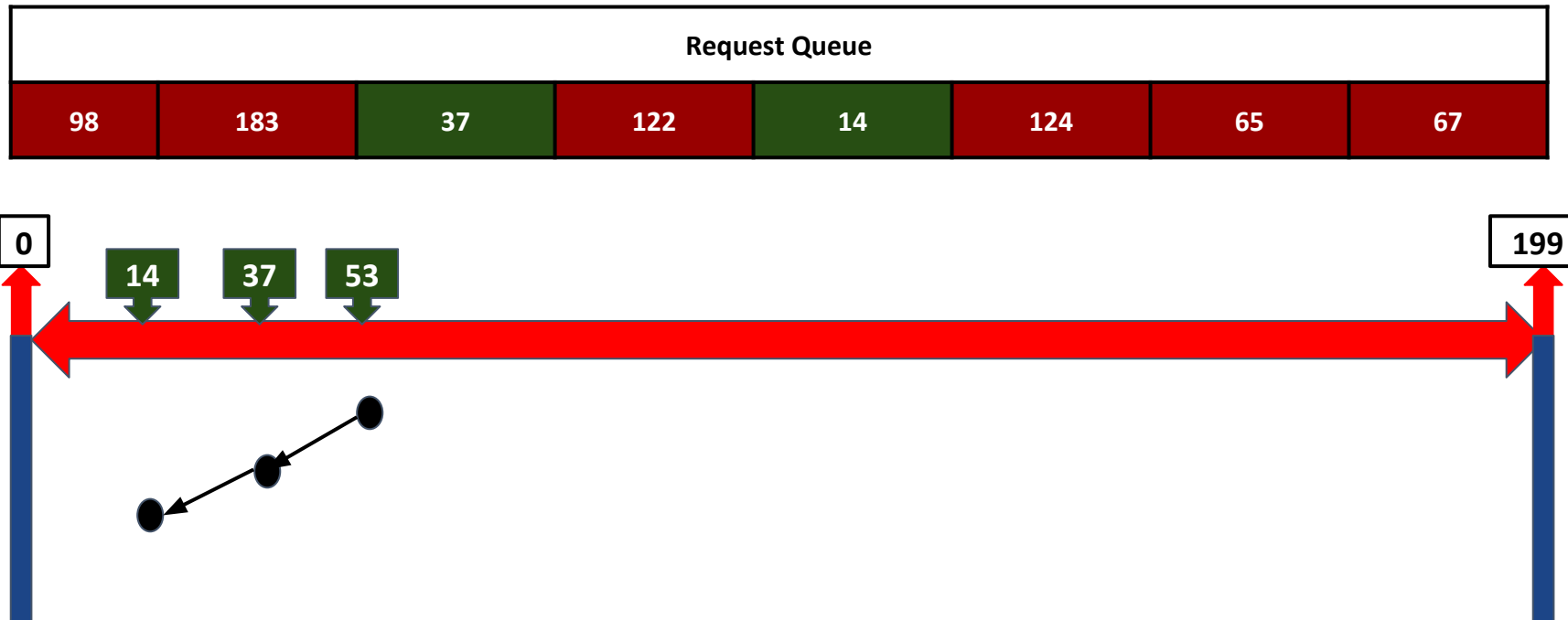
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $0 + \text{abs}(53-37)$

Seek Distance = 16

SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 16

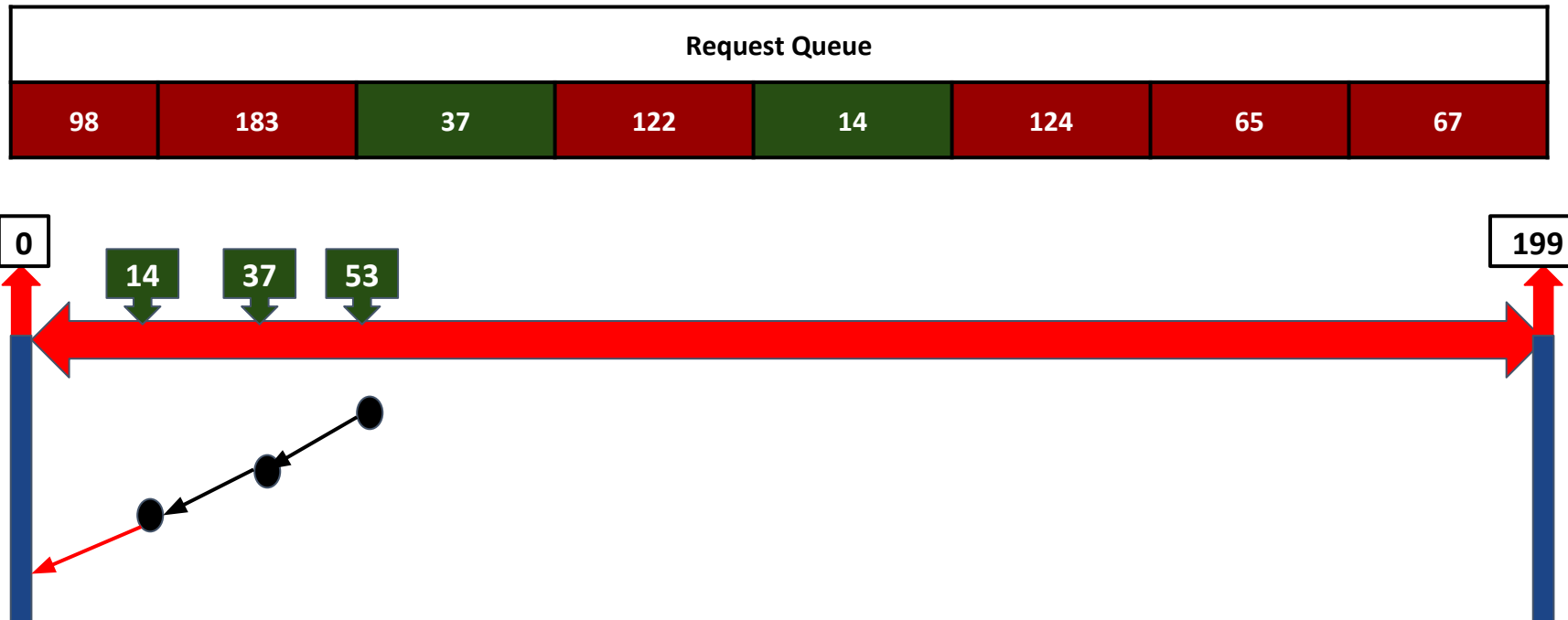
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $16 + \text{abs}(37-14)$

Seek Distance = 39

SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 39

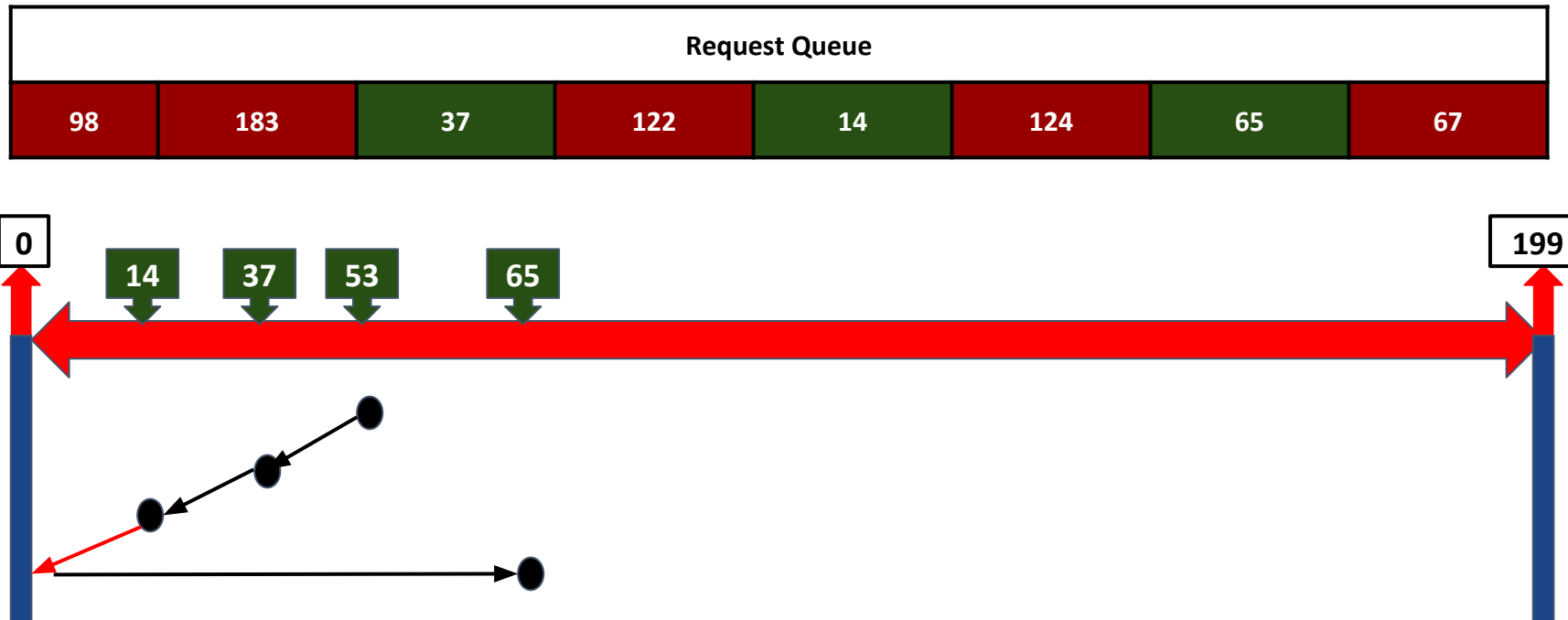
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $39 + \text{abs}(14-0)$

Seek Distance = 53

SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 53

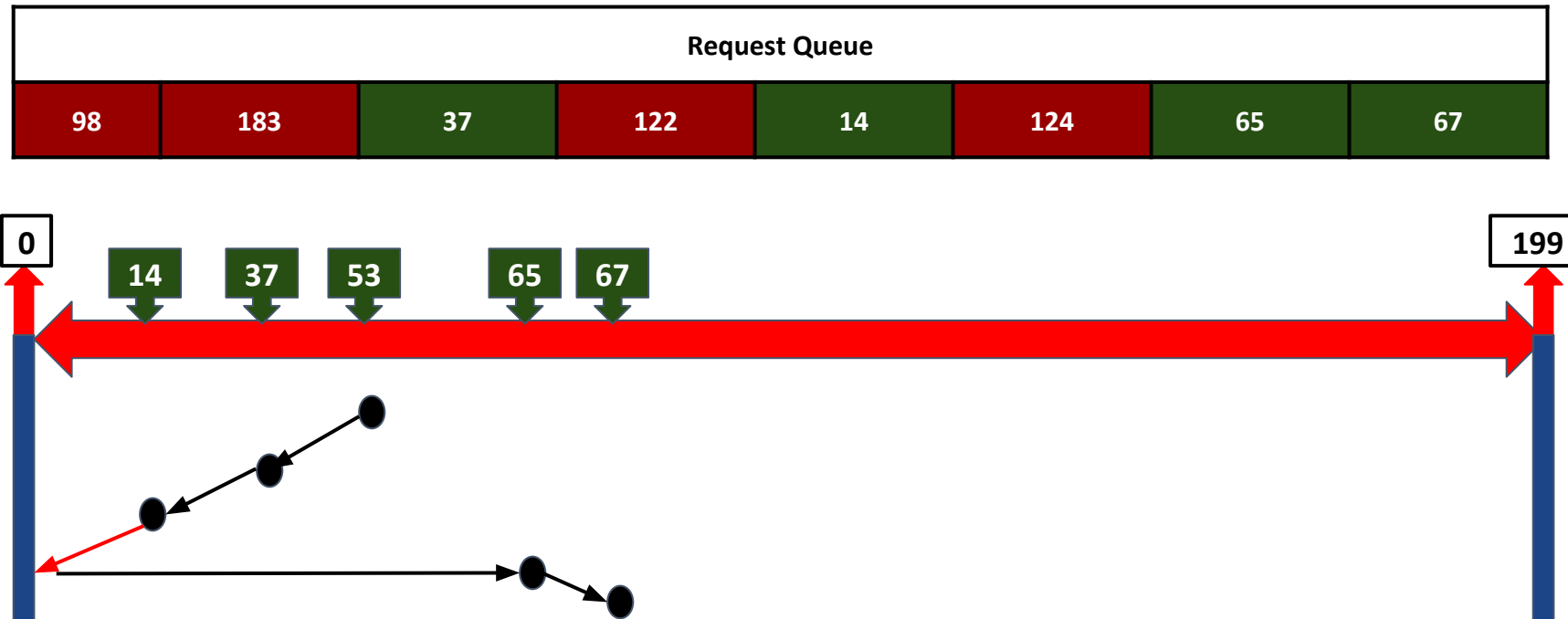
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $53 + \text{abs}(0 - 65)$

Seek Distance = 118

SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 118

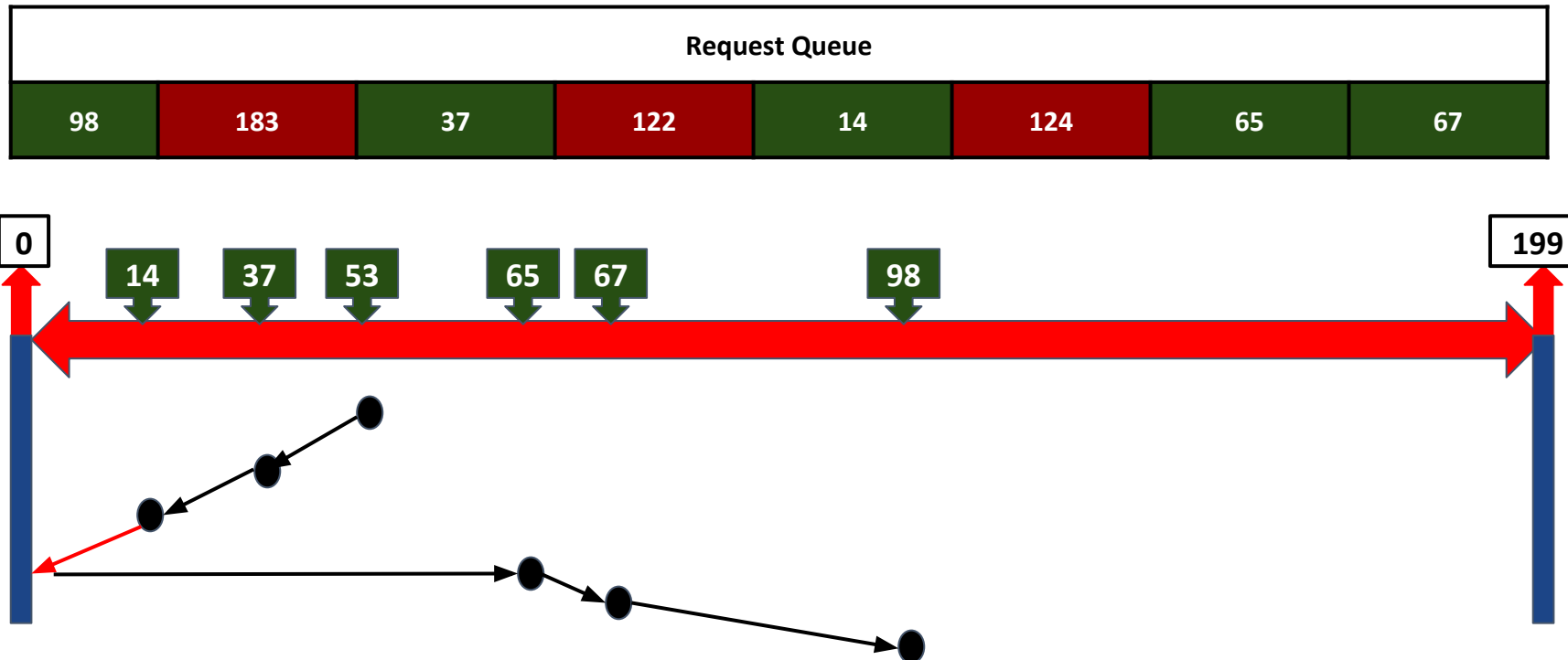
Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 118 + abs(65-53)

Seek Distance = 120

SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 120

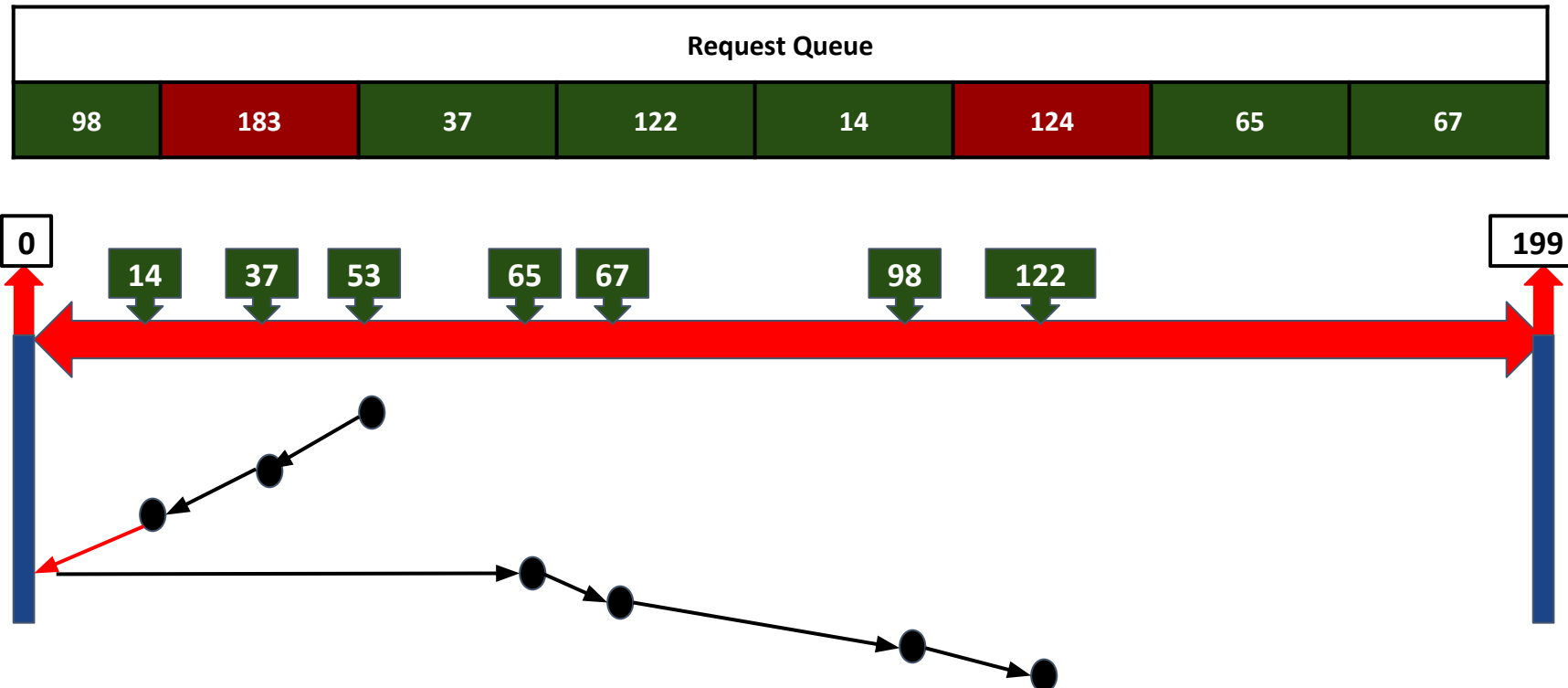
Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 120 + abs(67-98)

Seek Distance = 151

SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 151

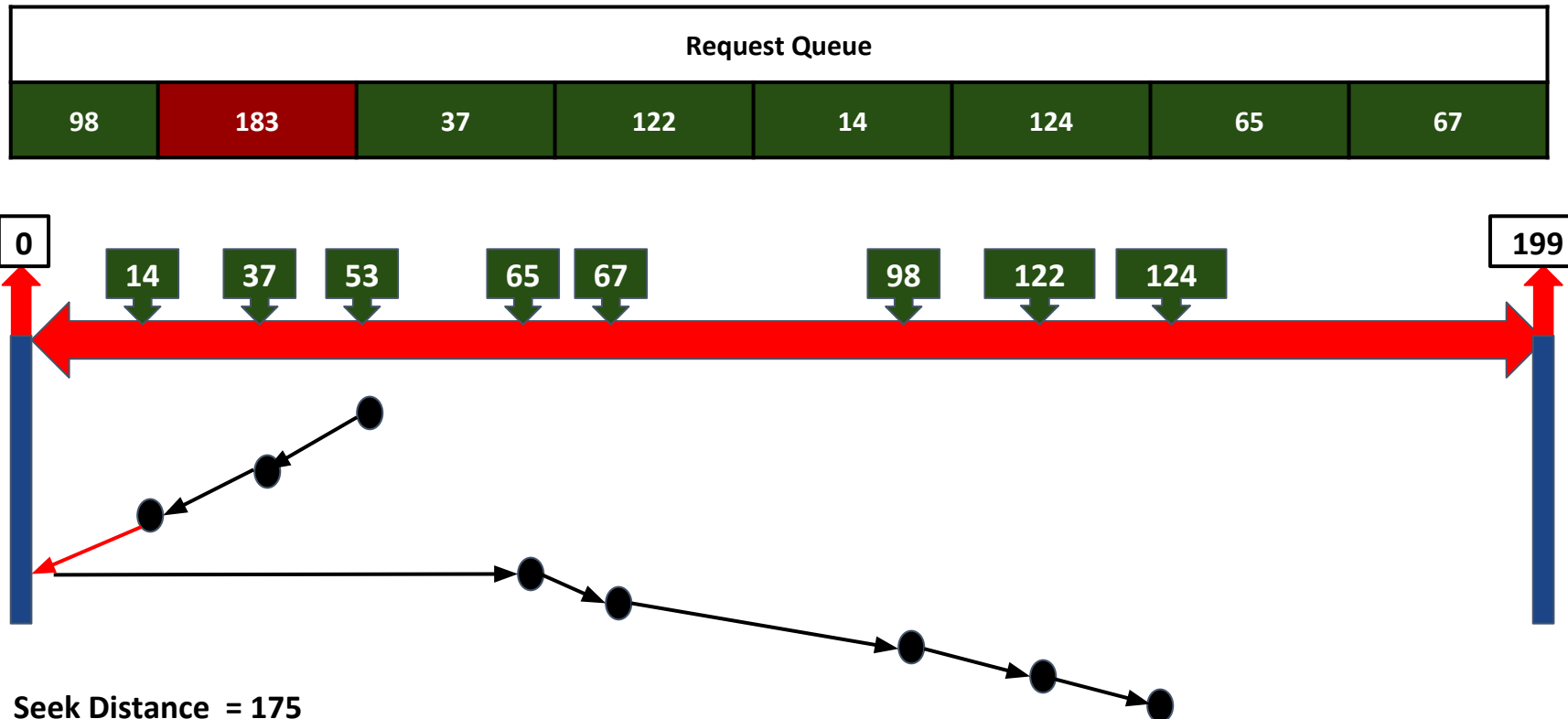
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 151 + $\text{abs}(98 - 122)$

Seek Distance = 175

SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 175

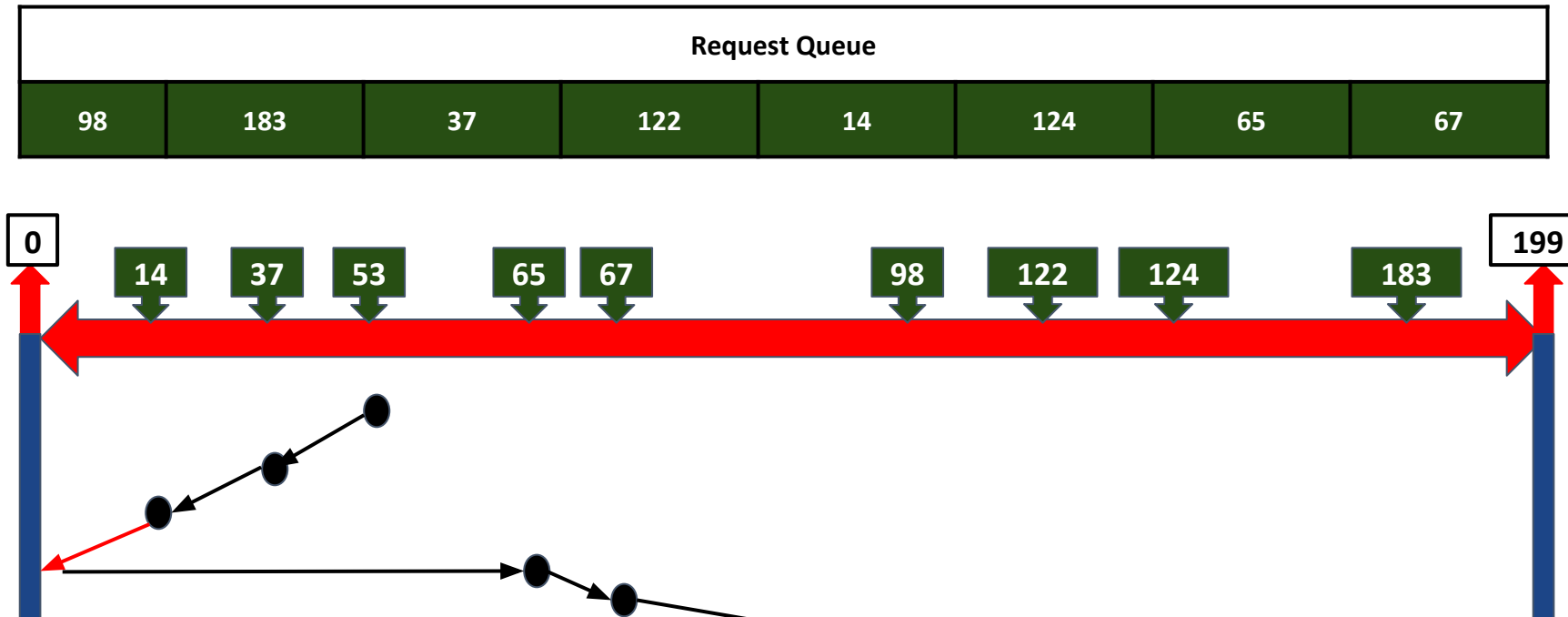
Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 175 + abs(122-124)

Seek Distance = 177

SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Final Seek Count
in terms of
Cylinders

236

Seek Distance = 177

Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $177 + \text{abs}(124 - 183)$

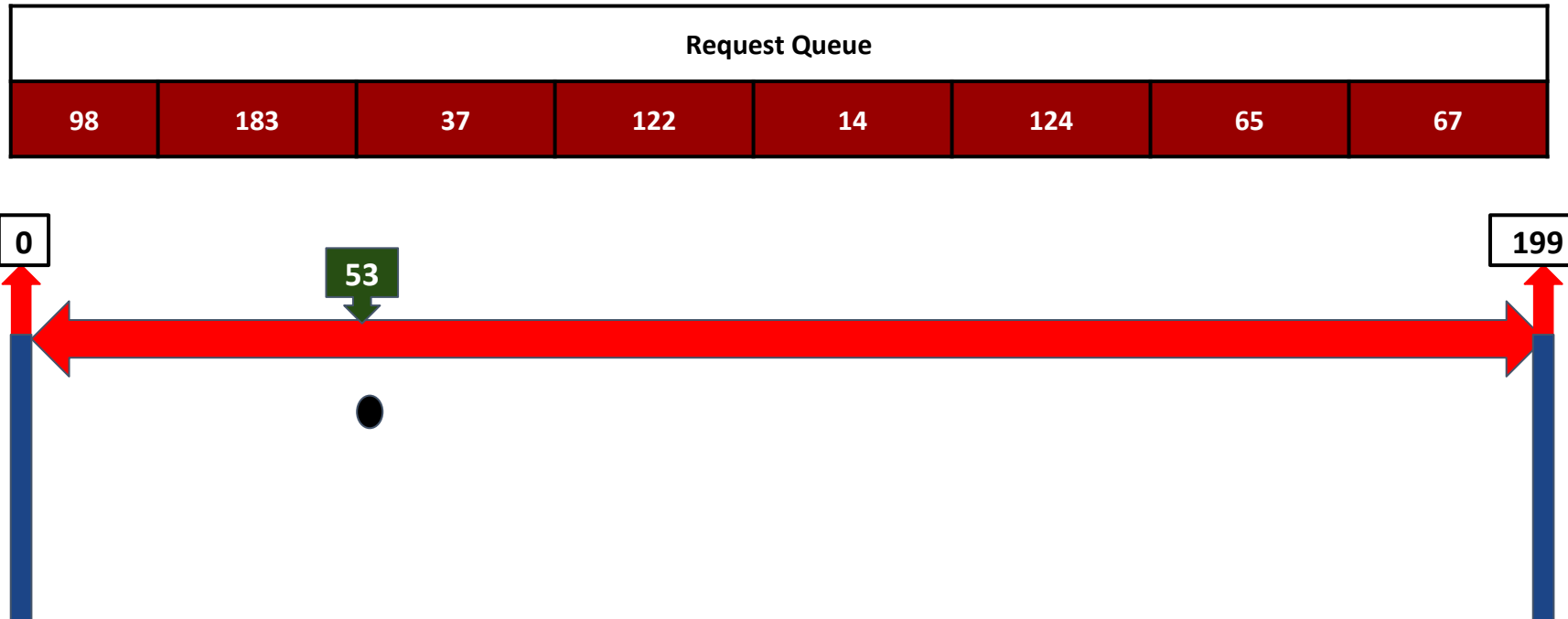
Seek Distance = 236

C - SCAN Disk Scheduling

- 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 Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 0

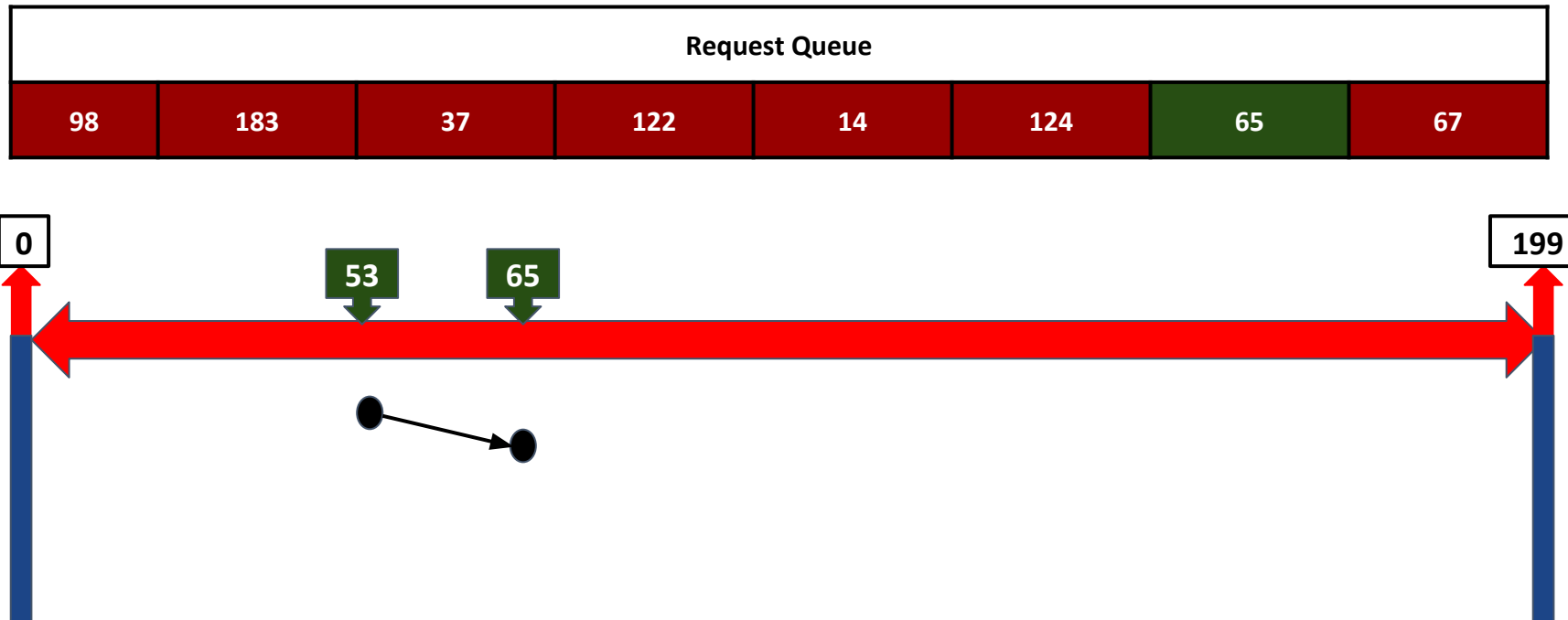
Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 0 + abs(0)

Seek Distance = 0

C - SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 0

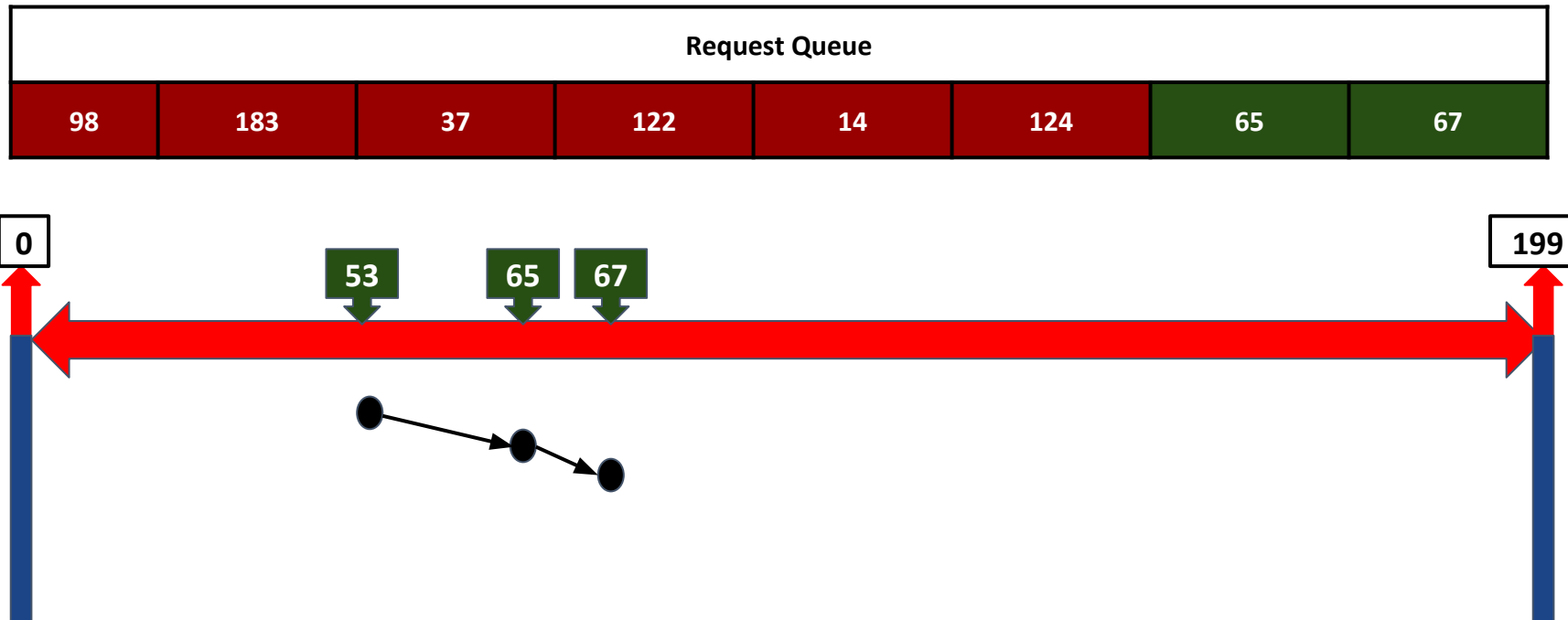
Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 0 + abs(53-65)

Seek Distance = 12

C - SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 12

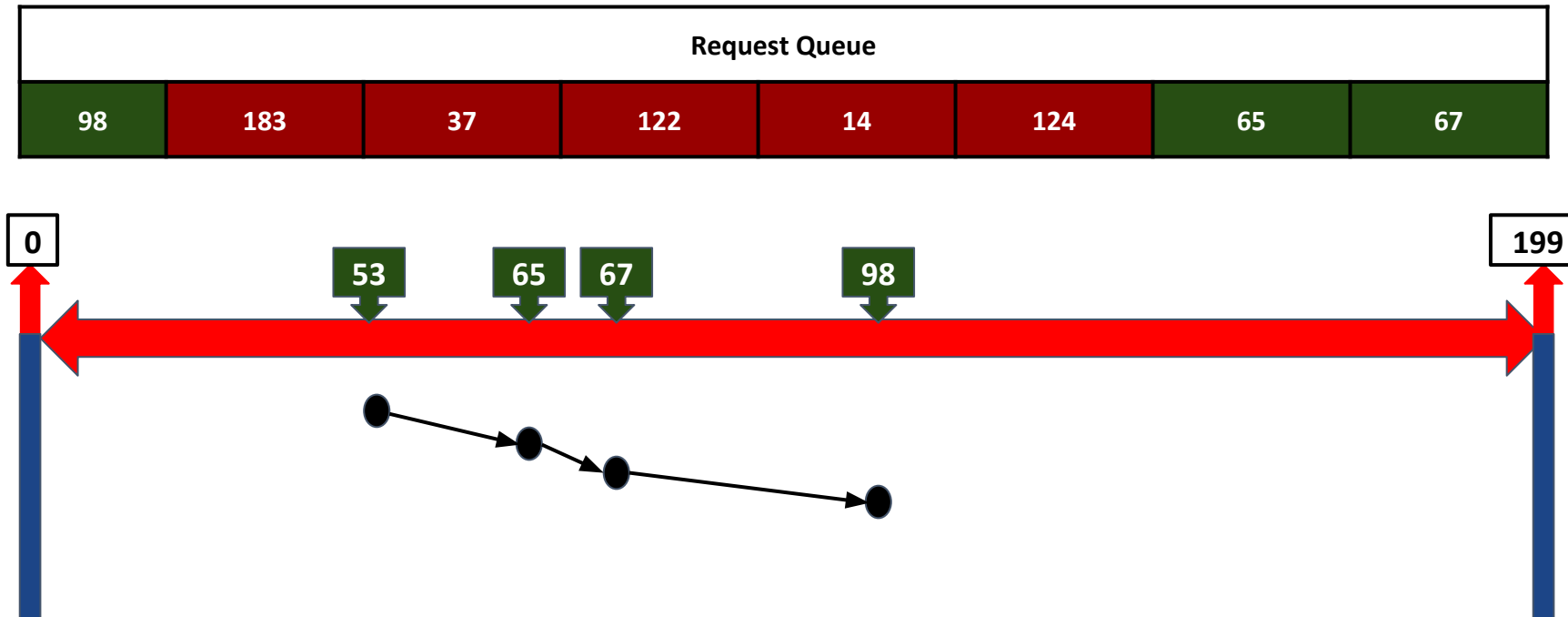
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 12 + $\text{abs}(65 - 67)$

Seek Distance = 14

C - SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 14

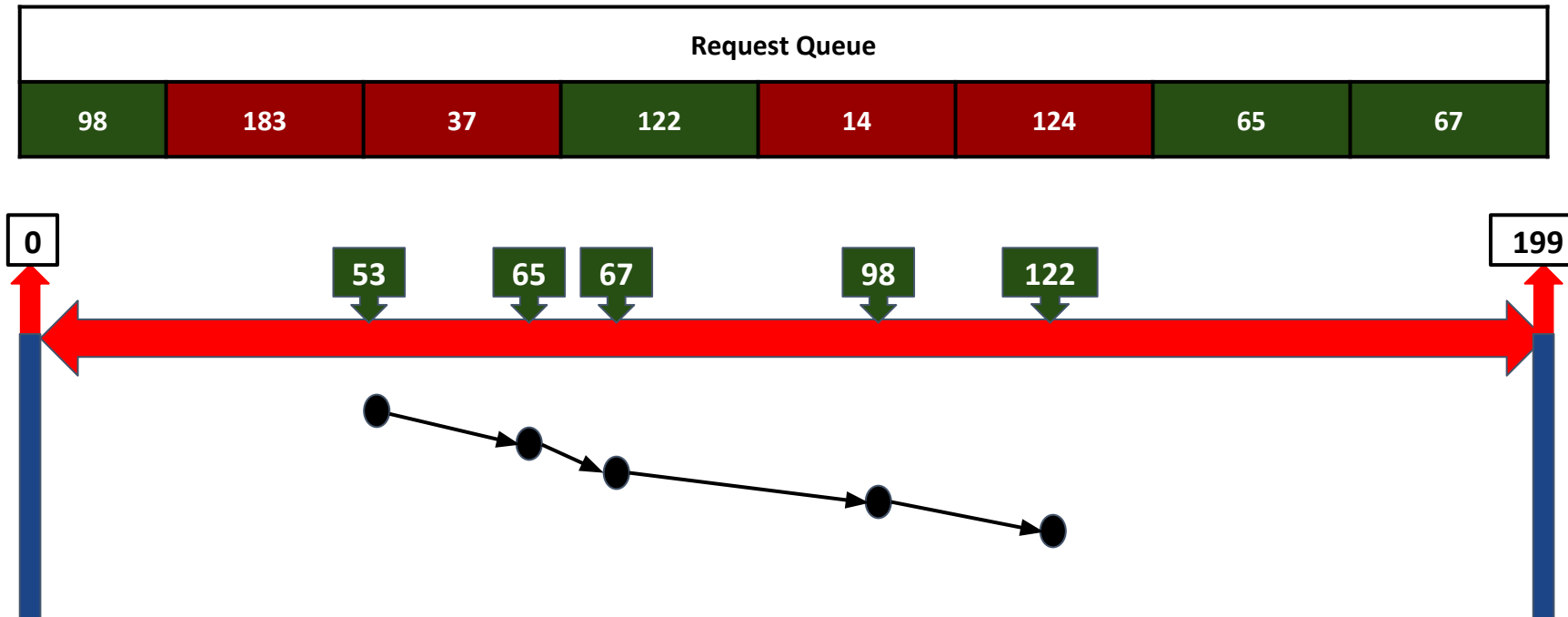
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $14 + \text{abs}(67-98)$

Seek Distance = 45

C - SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 45

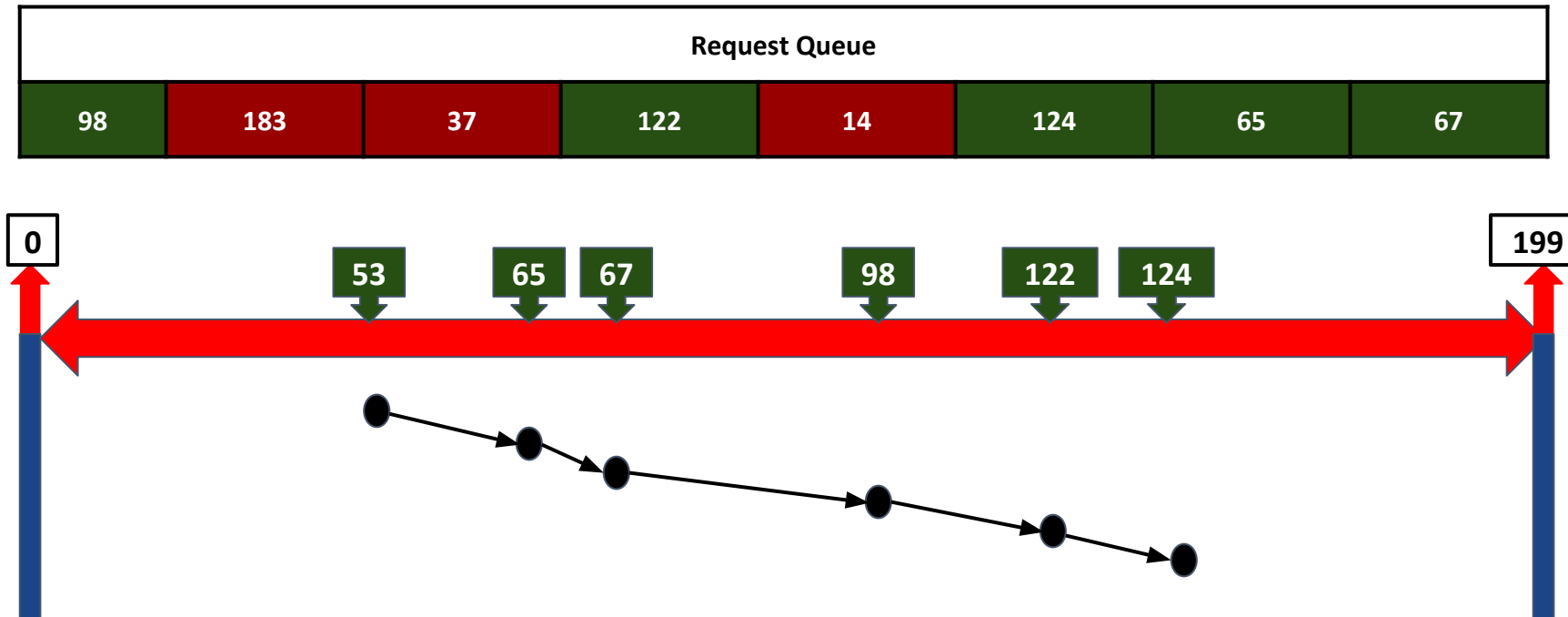
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $45 + \text{abs}(98 - 122)$

Seek Distance = 69

C - SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 69

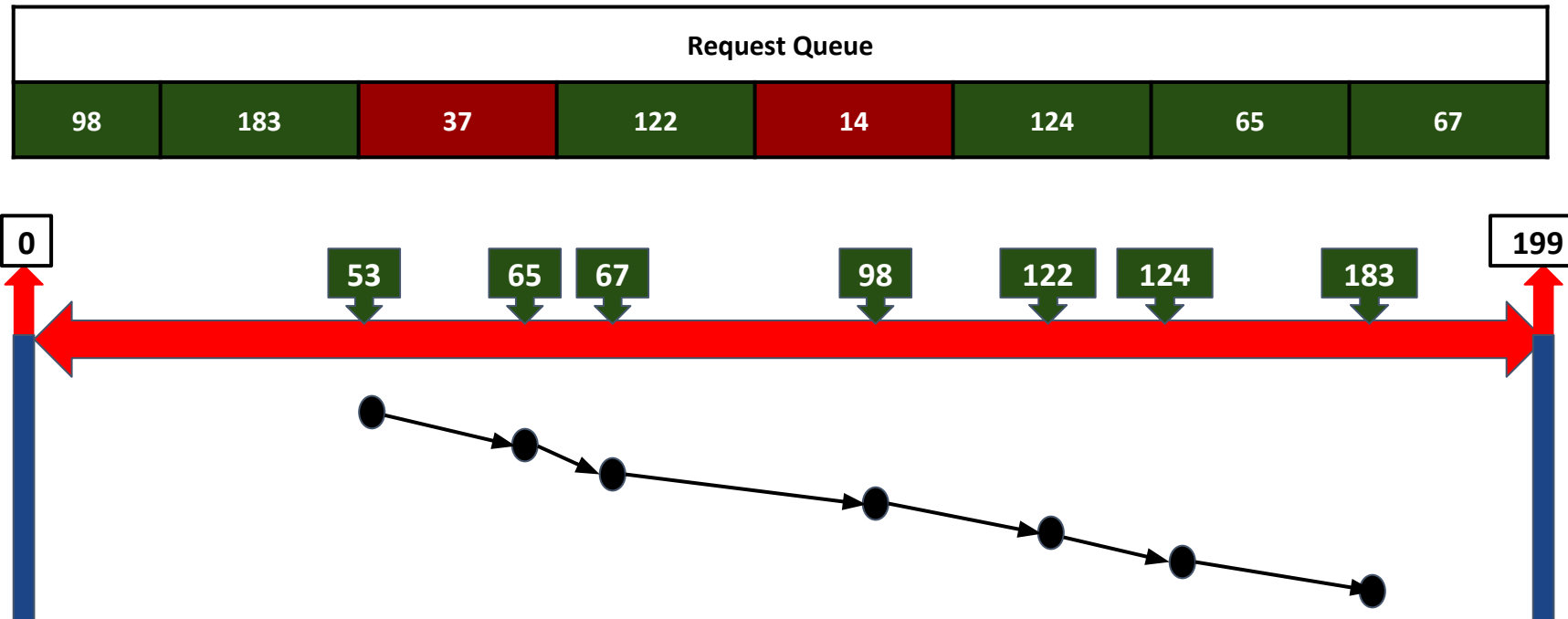
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $69 + \text{abs}(122 - 124)$

Seek Distance = 71

C - SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 71

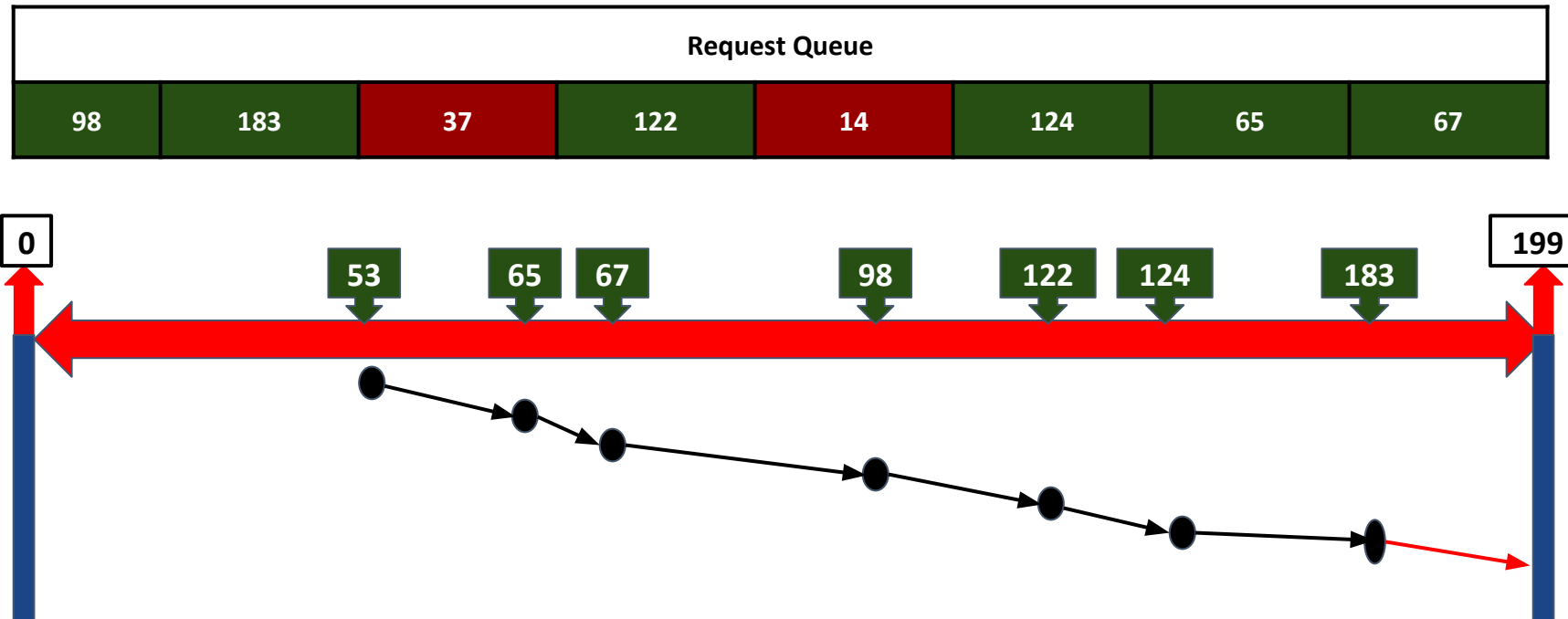
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 69 + $\text{abs}(124 - 183)$

Seek Distance = 130

C - SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 130

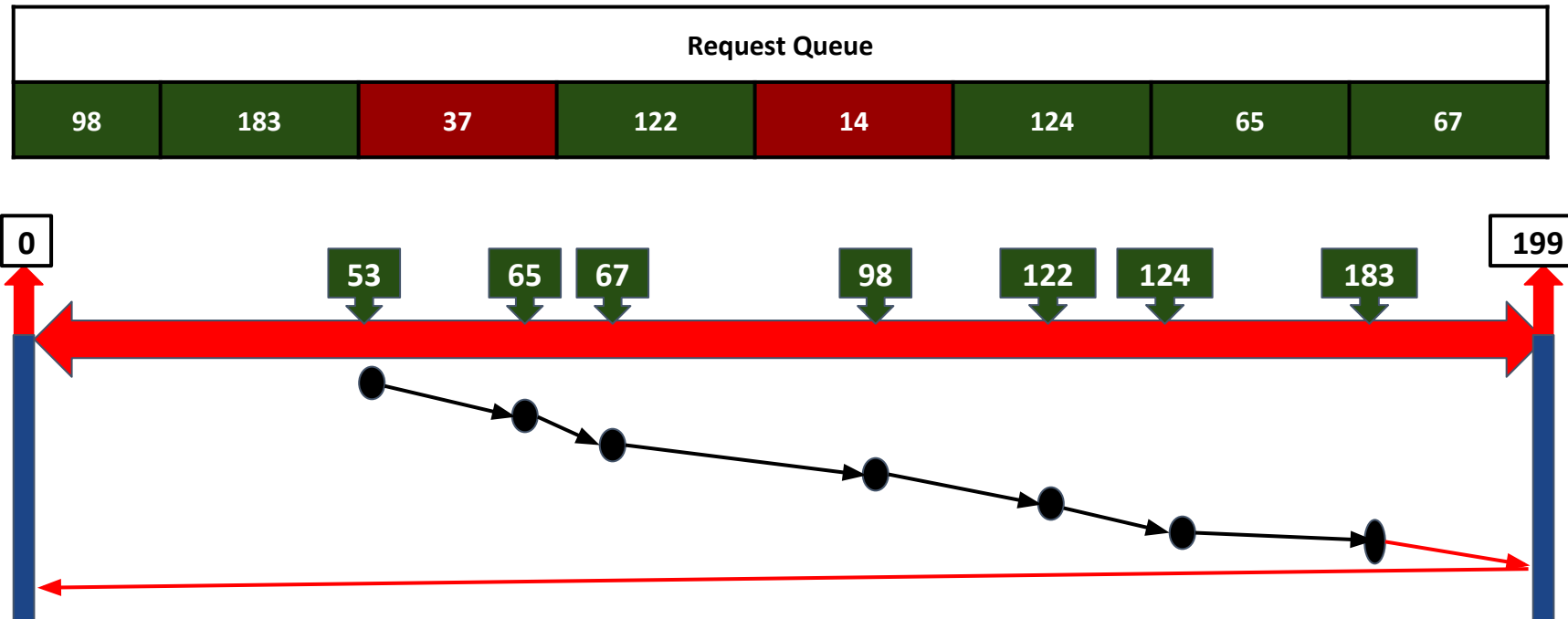
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 130 + $\text{abs}(183 - 199)$

Seek Distance = 146

C - SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 146

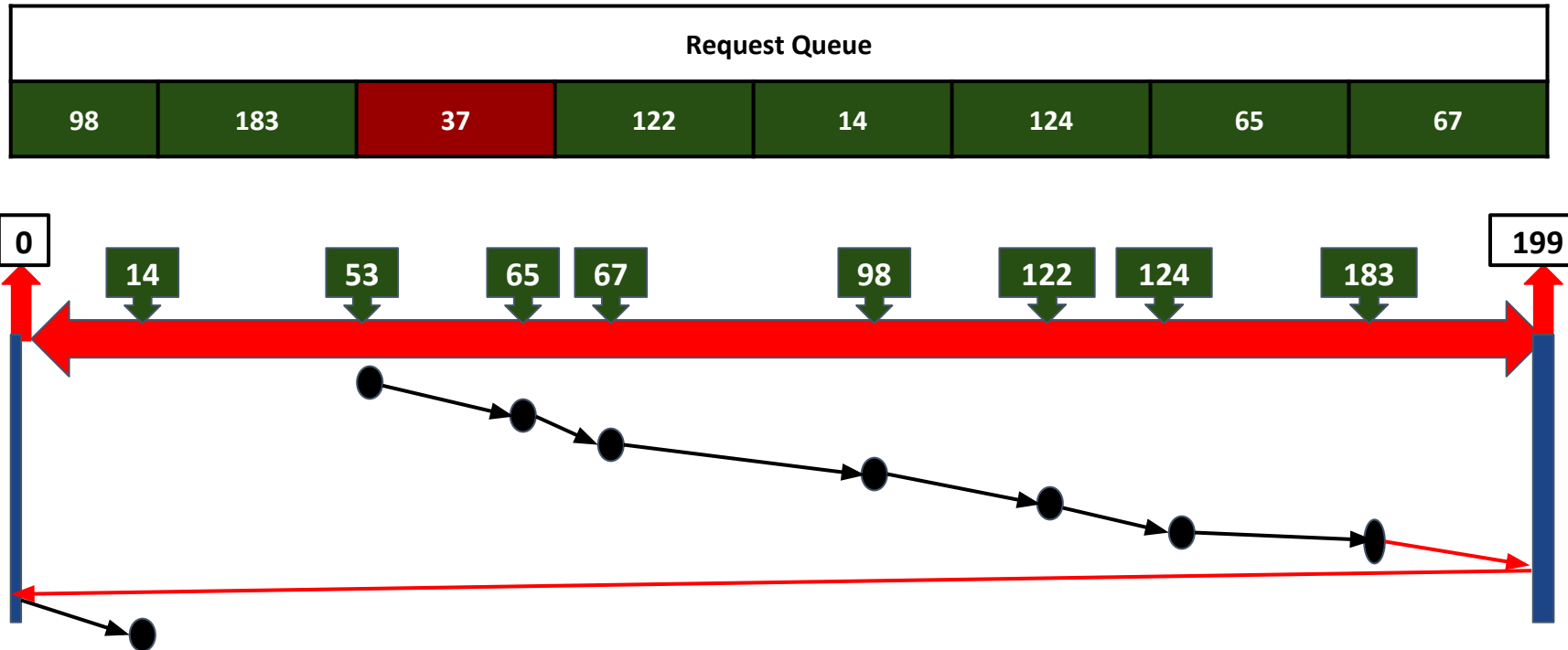
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 146 + $\text{abs}(199-0)$

Seek Distance = 345

C - SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 345

Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 345 + abs(0-14)

Seek Distance = 359

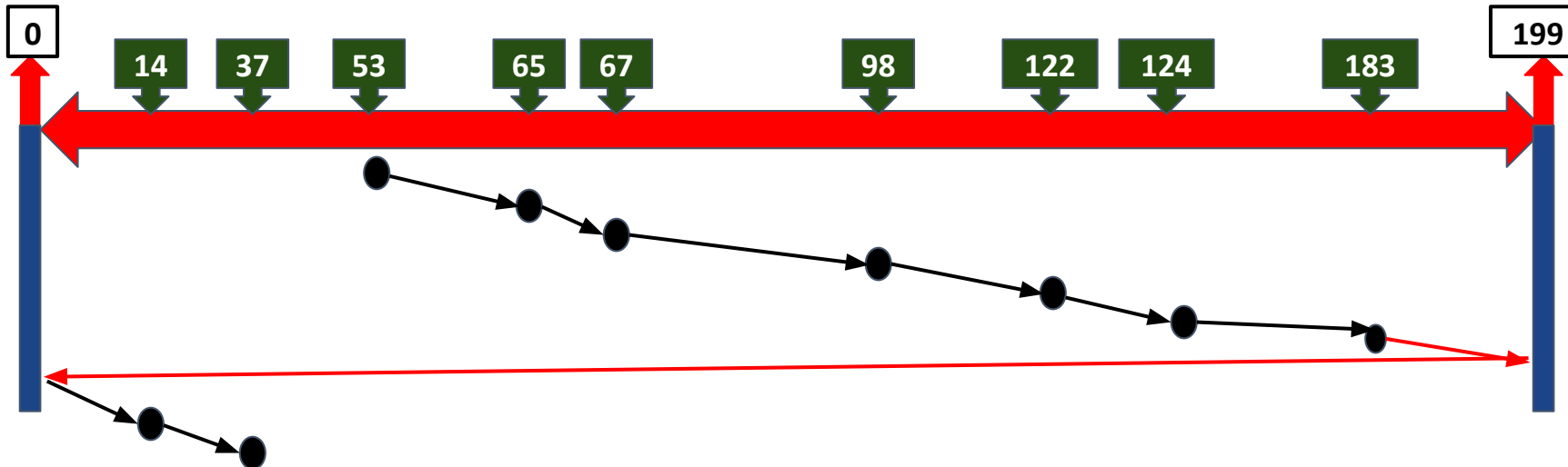
C - SCAN Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53

Request Queue							
98	183	37	122	14	124	65	67

Final Seek Count
in terms of
Cylinders

382



Seek Distance = 359

Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 359 + $\text{abs}(14 - 37)$

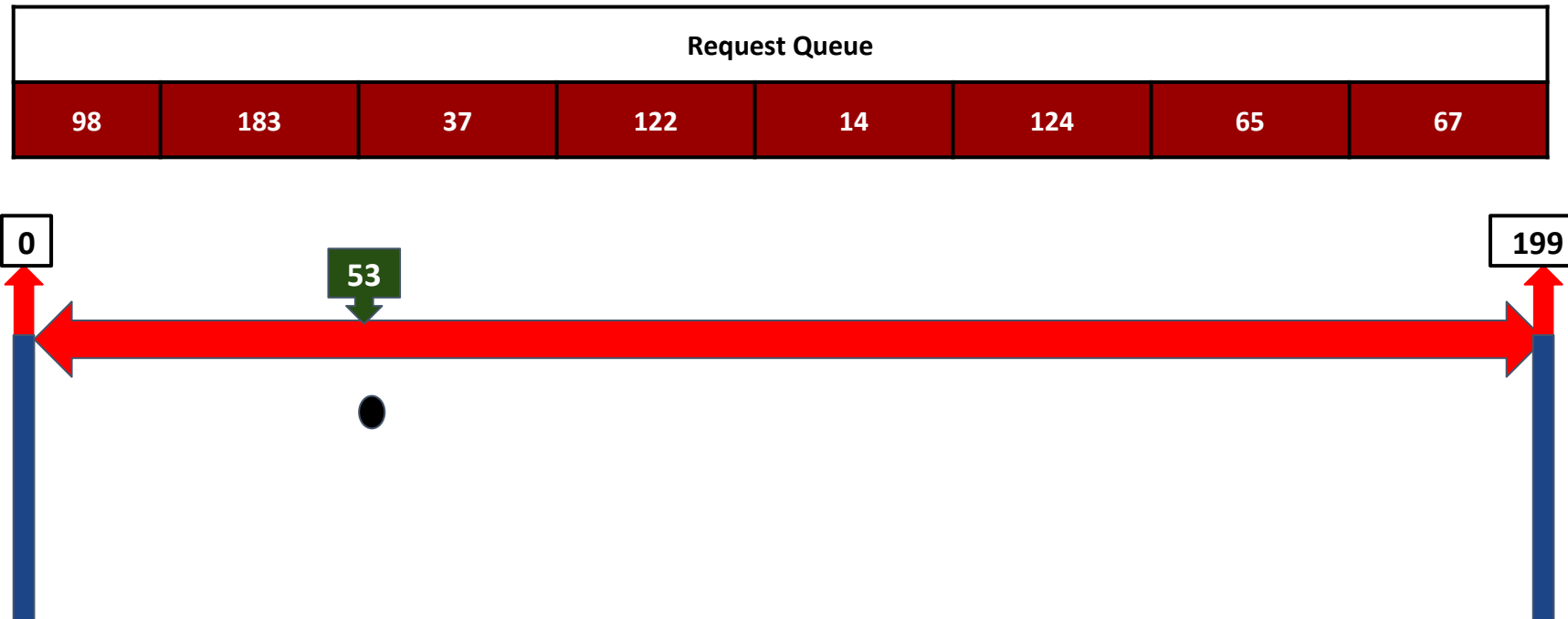
Seek Distance = 382

LOOK Disk Scheduling

- LOOK is the advanced version of SCAN (elevator) disk scheduling algorithm which gives slightly better seek time
- The LOOK algorithm services request similarly as SCAN algorithm meanwhile it also “looks” ahead as if there are more tracks that are needed to be serviced in the same direction.
- If there are no pending requests in the moving direction the head reverses the direction and start servicing requests in the opposite direction.
- The main reason behind the better performance of LOOK algorithm in comparison to SCAN is because in this algorithm the head is not allowed to move till the end of the disk.

LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 0

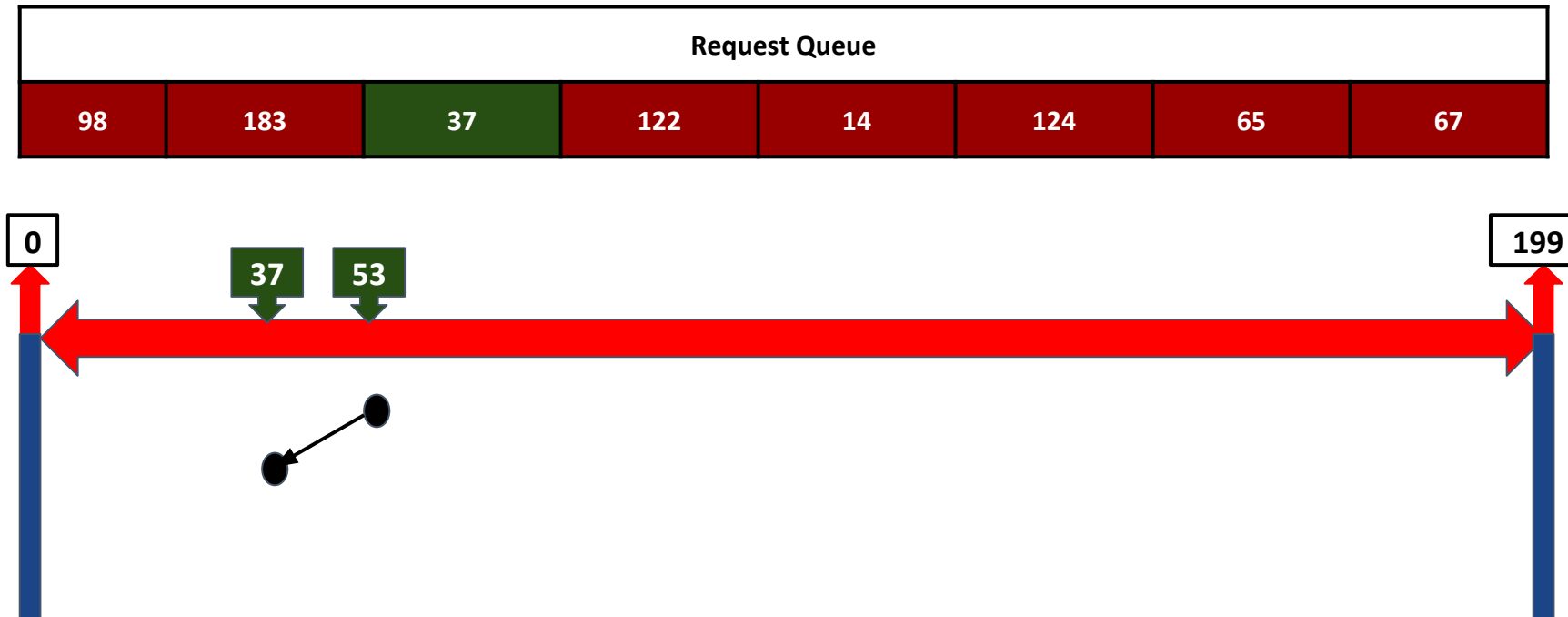
Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 0 + abs()

Seek Distance = 0

LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 0

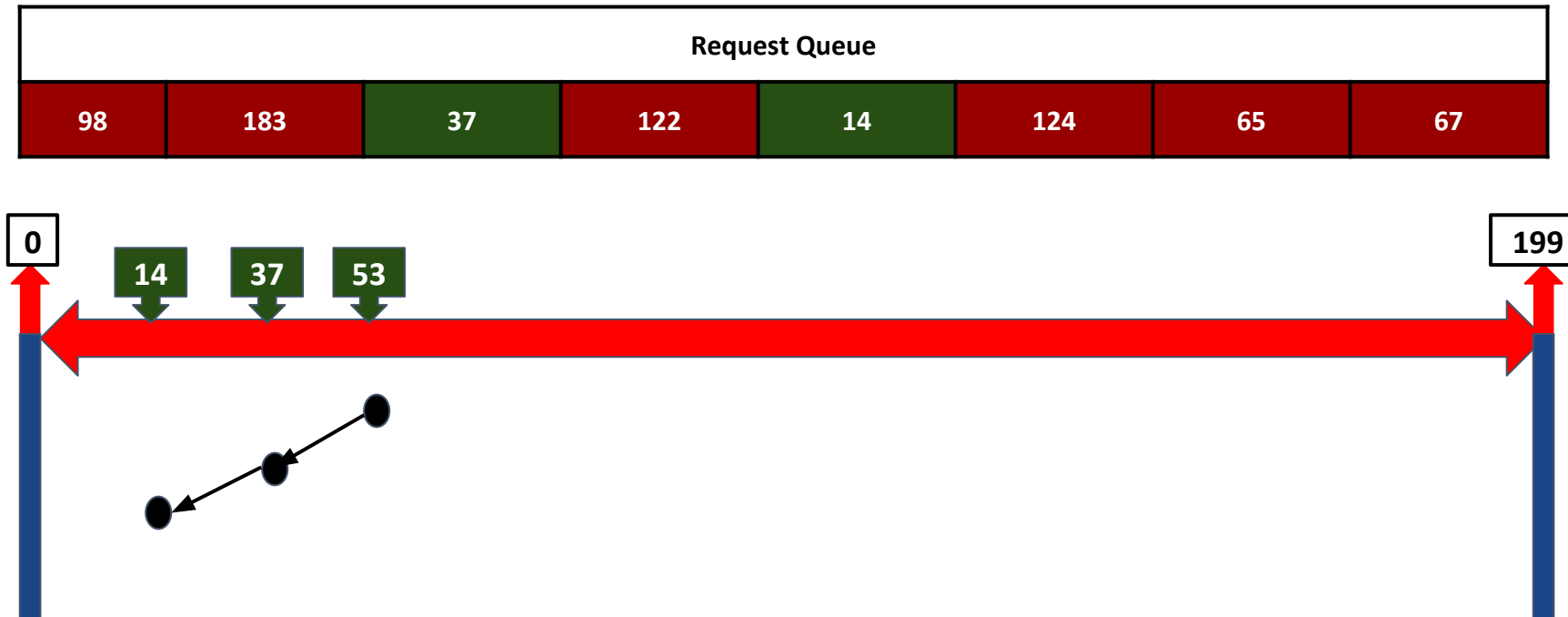
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $0 + \text{abs}(53-37)$

Seek Distance = 16

LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 16

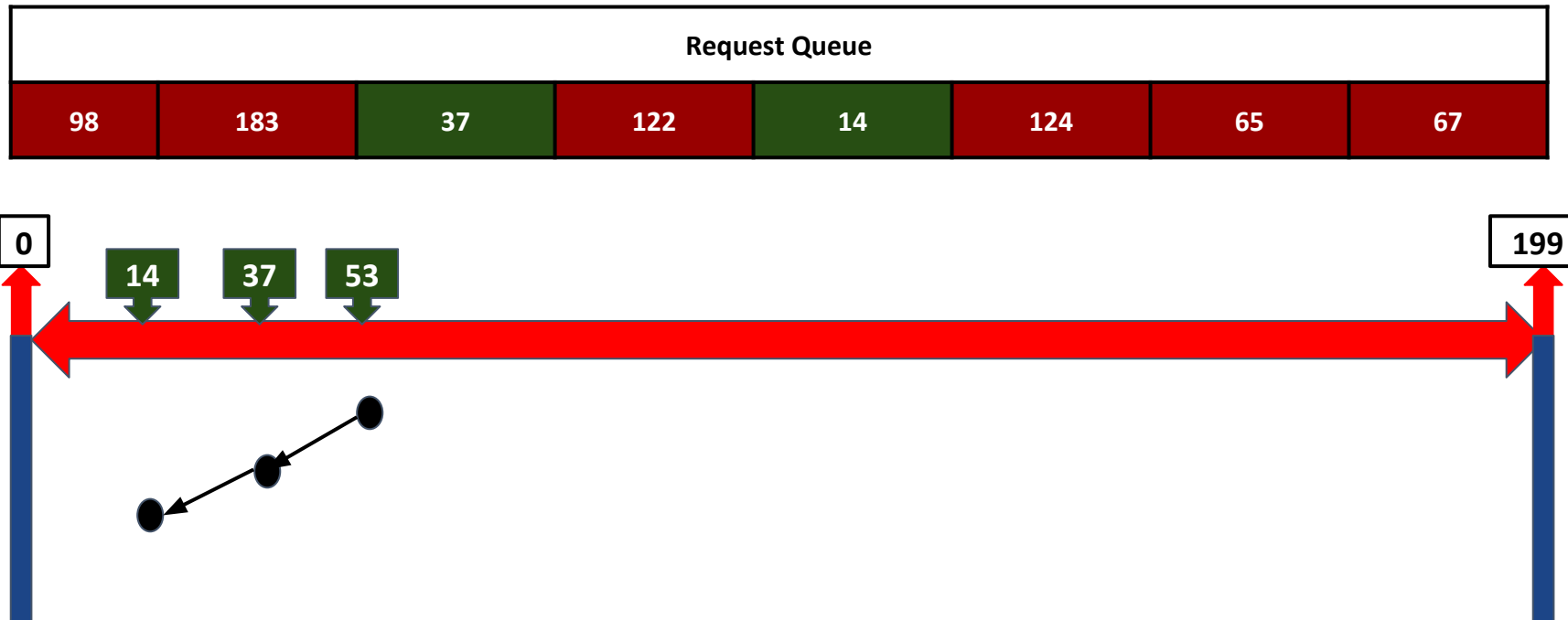
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $16 + \text{abs}(37-14)$

Seek Distance = 39

LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 39

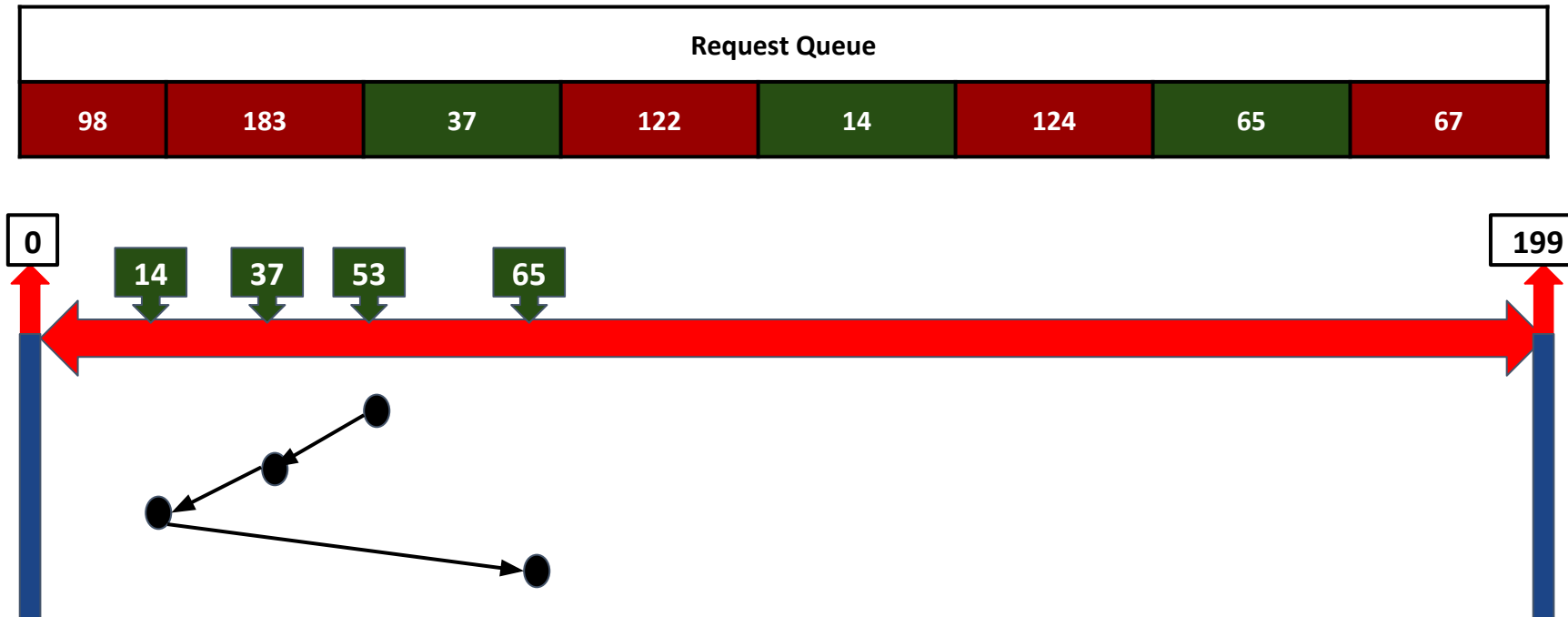
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $39 + \text{abs}(0-0)$

Seek Distance = 39

LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 39

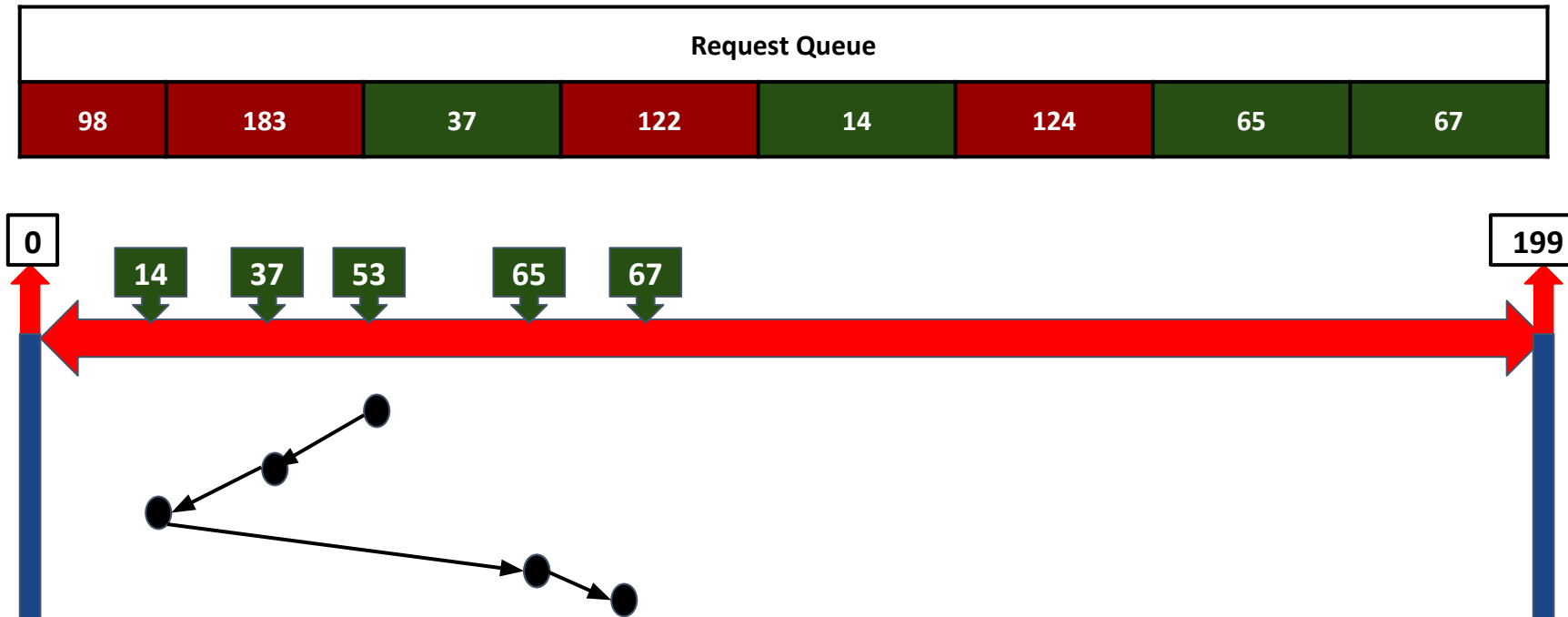
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 39 + $\text{abs}(14 - 65)$

Seek Distance = 90

LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 90

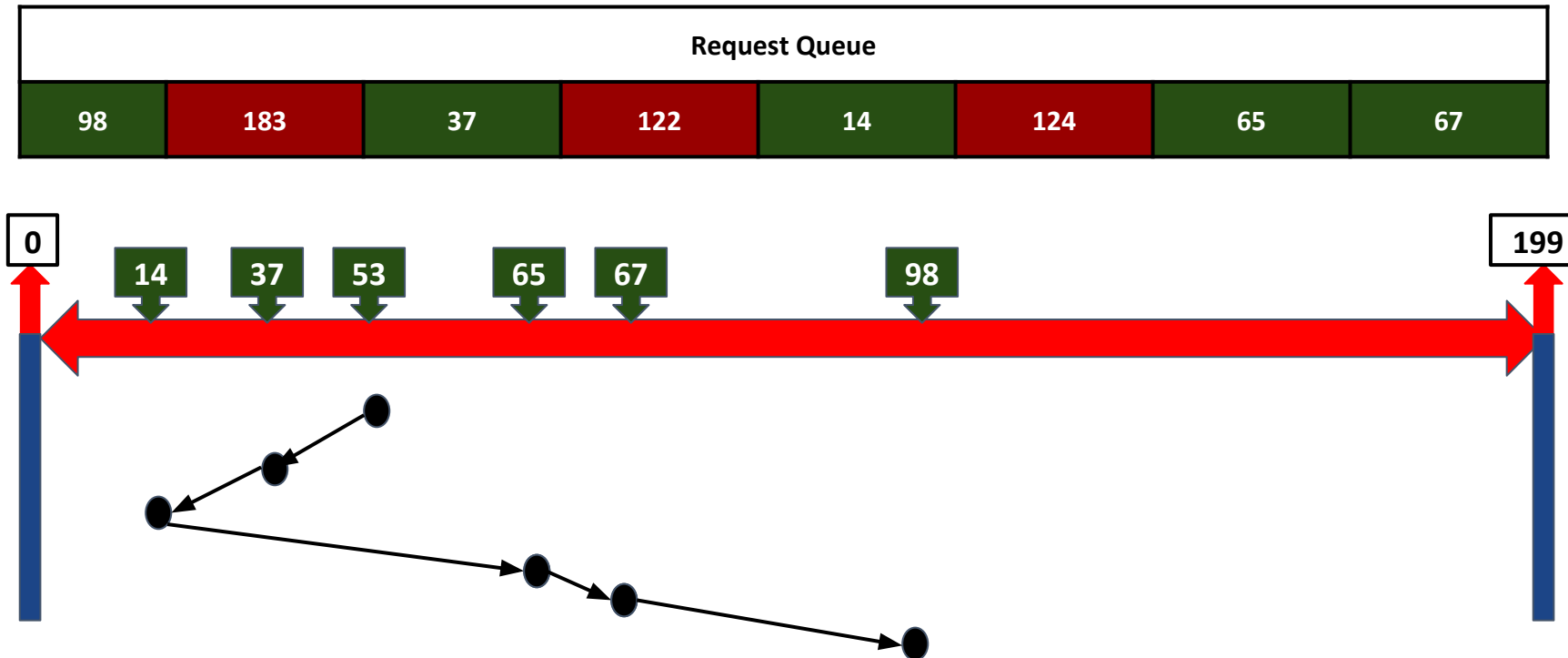
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $90 + \text{abs}(65 - 67)$

Seek Distance = 92

LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 92

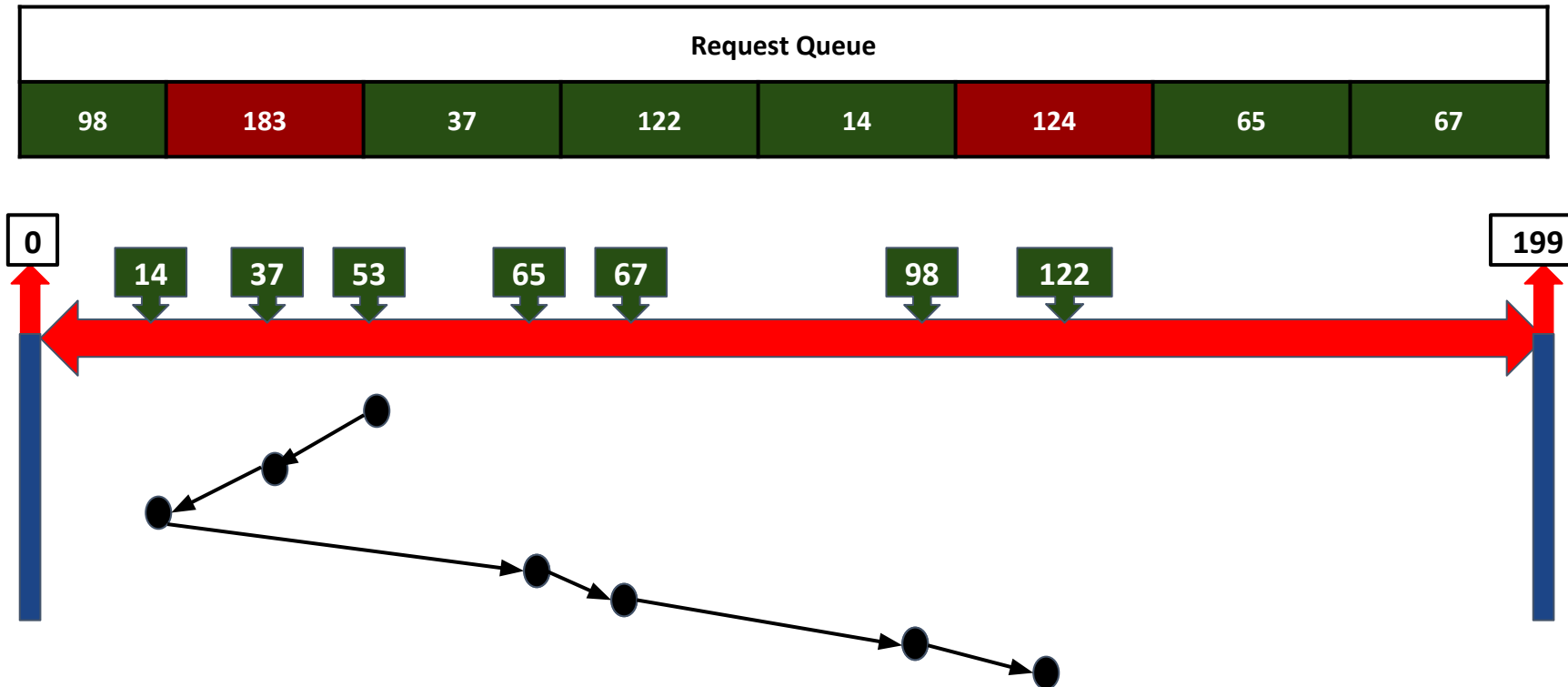
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $92 + \text{abs}(67-98)$

Seek Distance = 123

LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 123

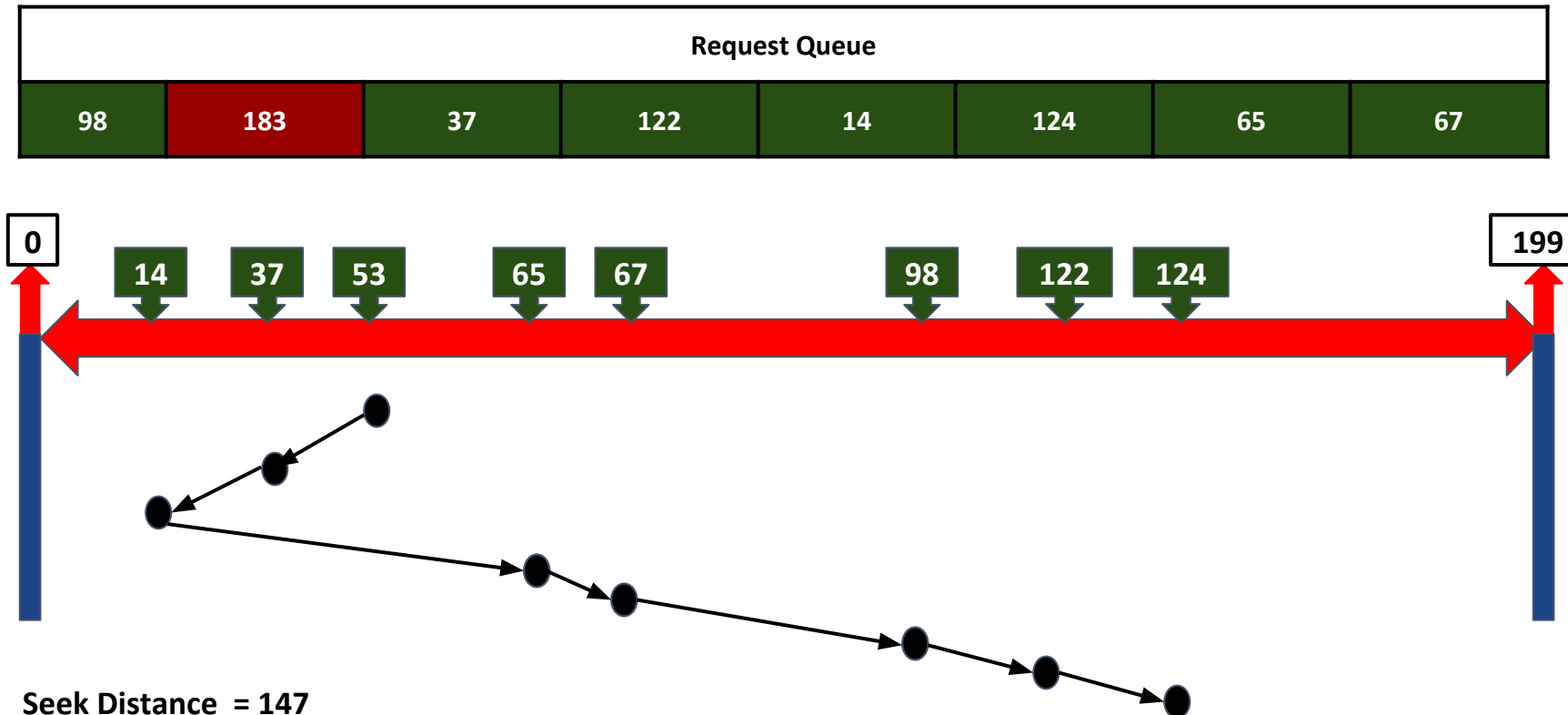
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 123 + $\text{abs}(98-122)$

Seek Distance = 147

LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 147

Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 147 + abs(122-124)

Seek Distance = 149

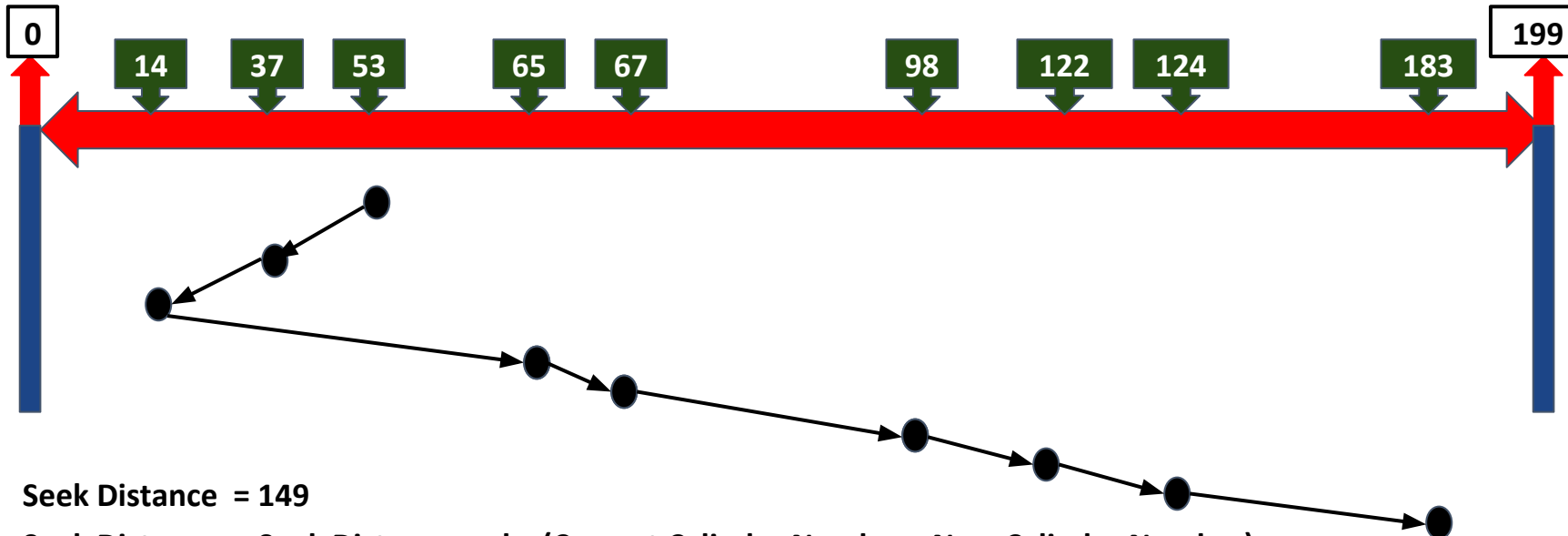
LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53

Request Queue							
98	183	37	122	14	124	65	67

Final Seek Count
in terms of
Cylinders

208



Seek Distance = 149

Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $149 + \text{abs}(124 - 183)$

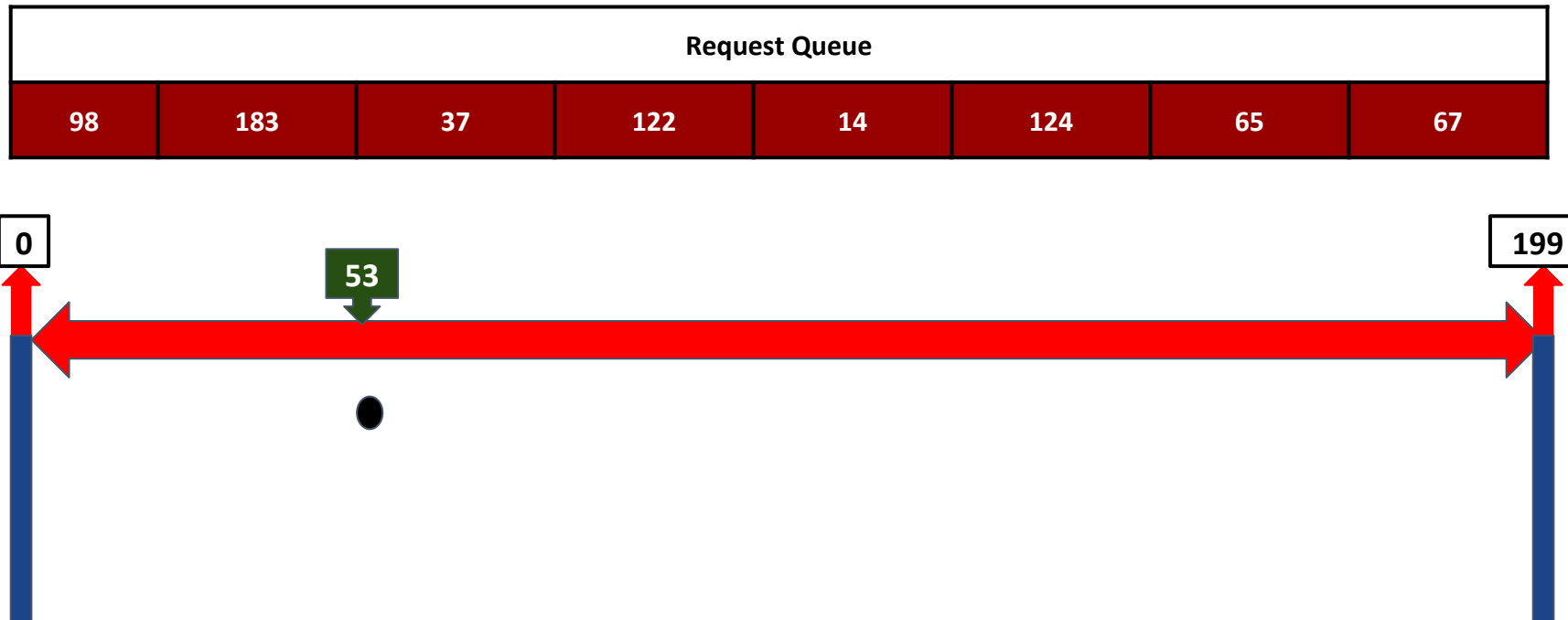
Seek Distance = 208

C - LOOK Disk Scheduling

- LOOK a version of SCAN, C-LOOK a 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
- In this algorithm, the head services requests only in one direction(either left or right) until all the requests in this direction are not serviced and then jumps back to the farthest request on the other direction and service the remaining requests which gives a better uniform servicing as well as avoids wasting seek time for going till the end of the disk.
- Total number of cylinders ?

C - LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 0

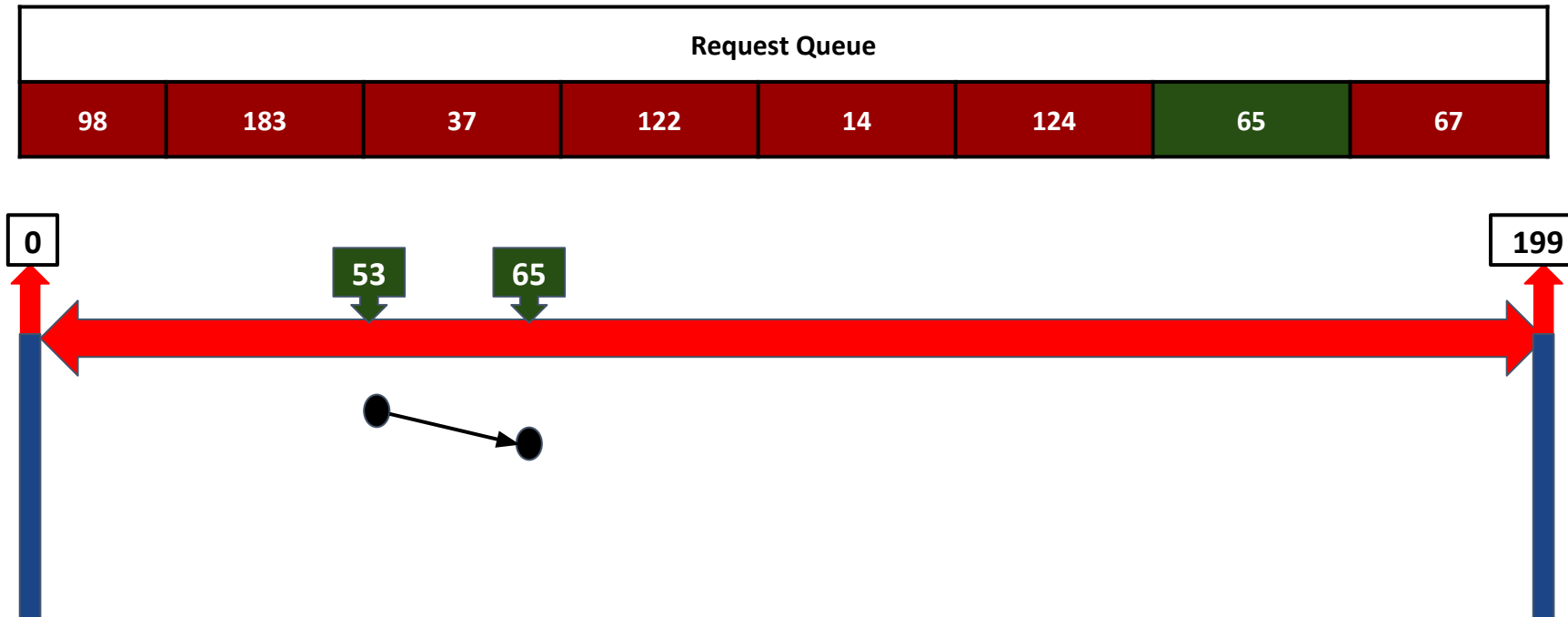
Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 0 + abs(0)

Seek Distance = 0

C - LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 0

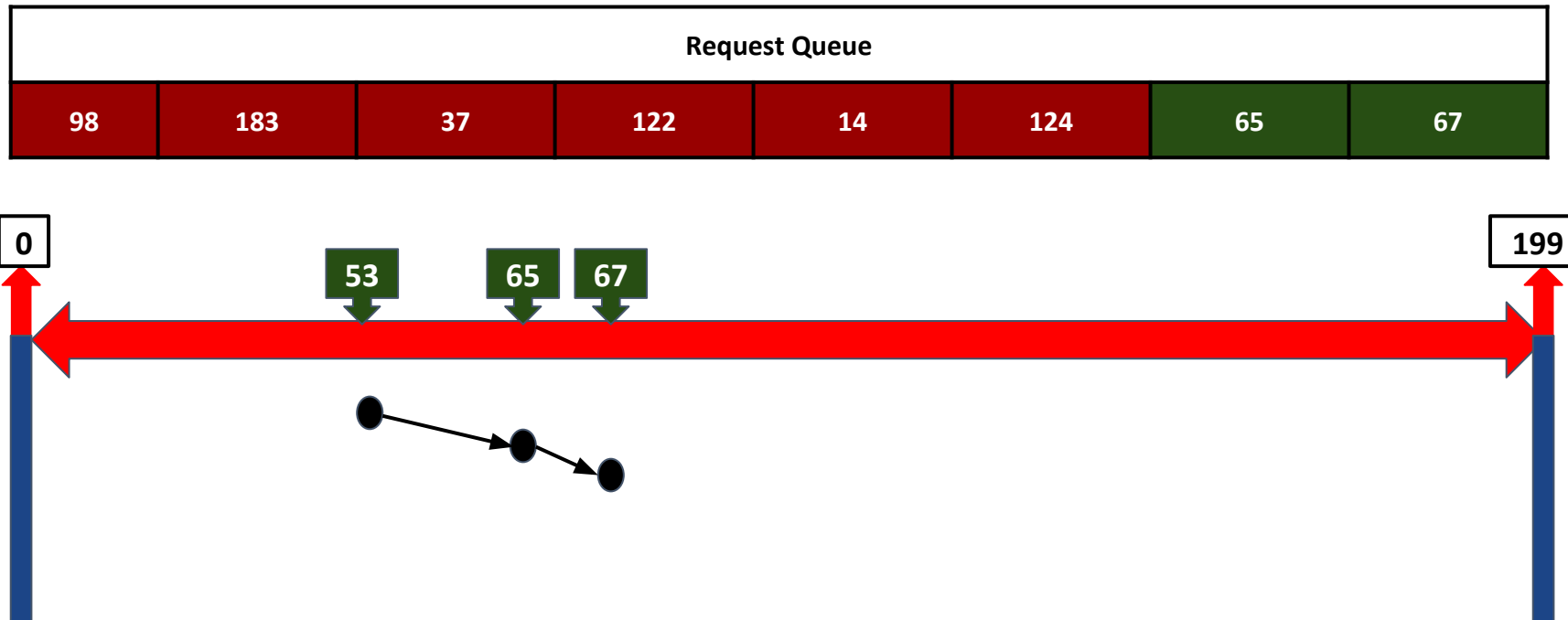
Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 0 + abs(53-65)

Seek Distance = 12

C - LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 12

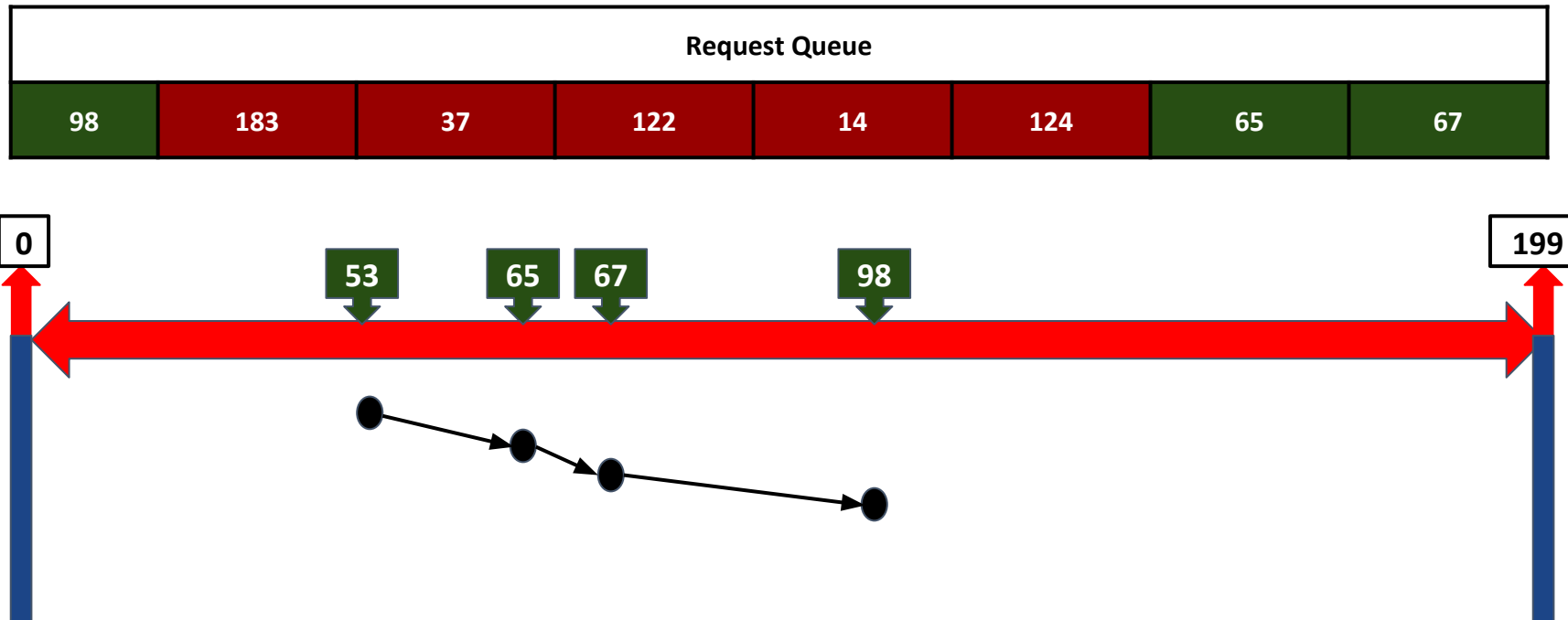
Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 12 + abs(65-67)

Seek Distance = 14

C - LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 14

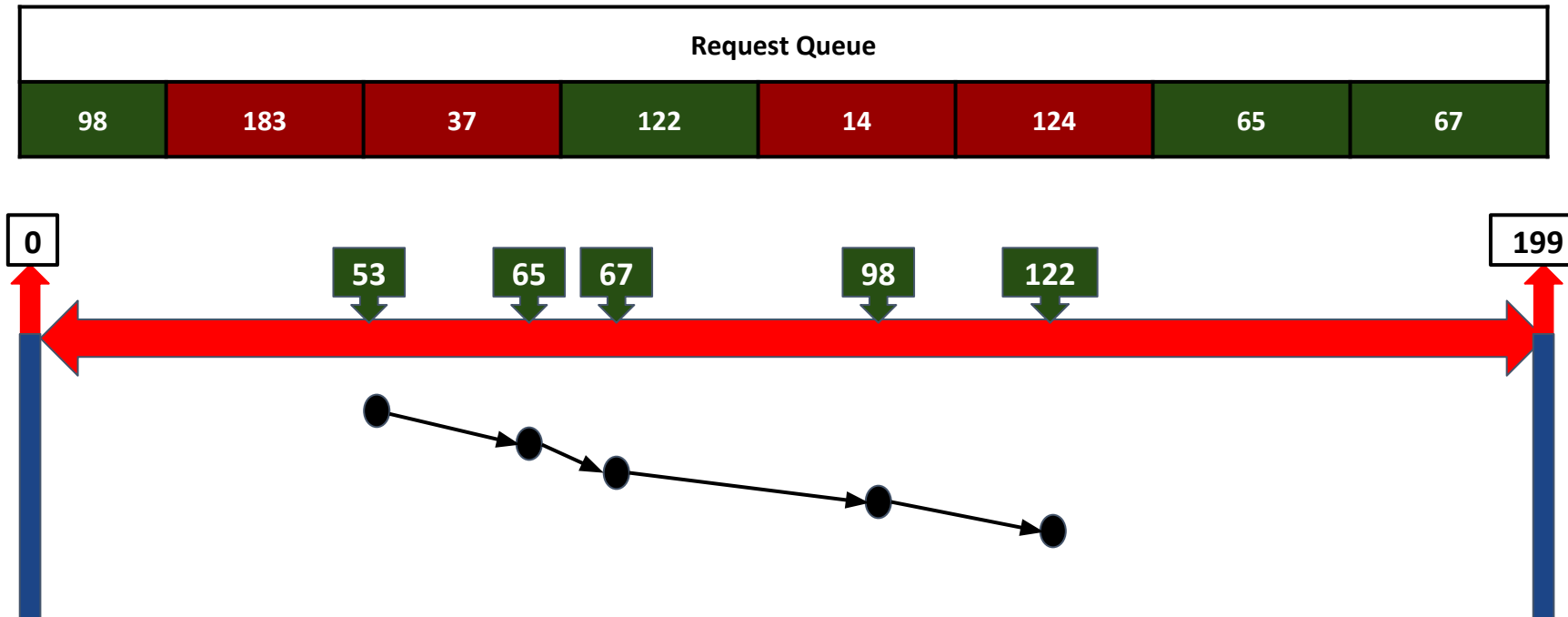
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 14 + $\text{abs}(67-98)$

Seek Distance = 45

C - LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 45

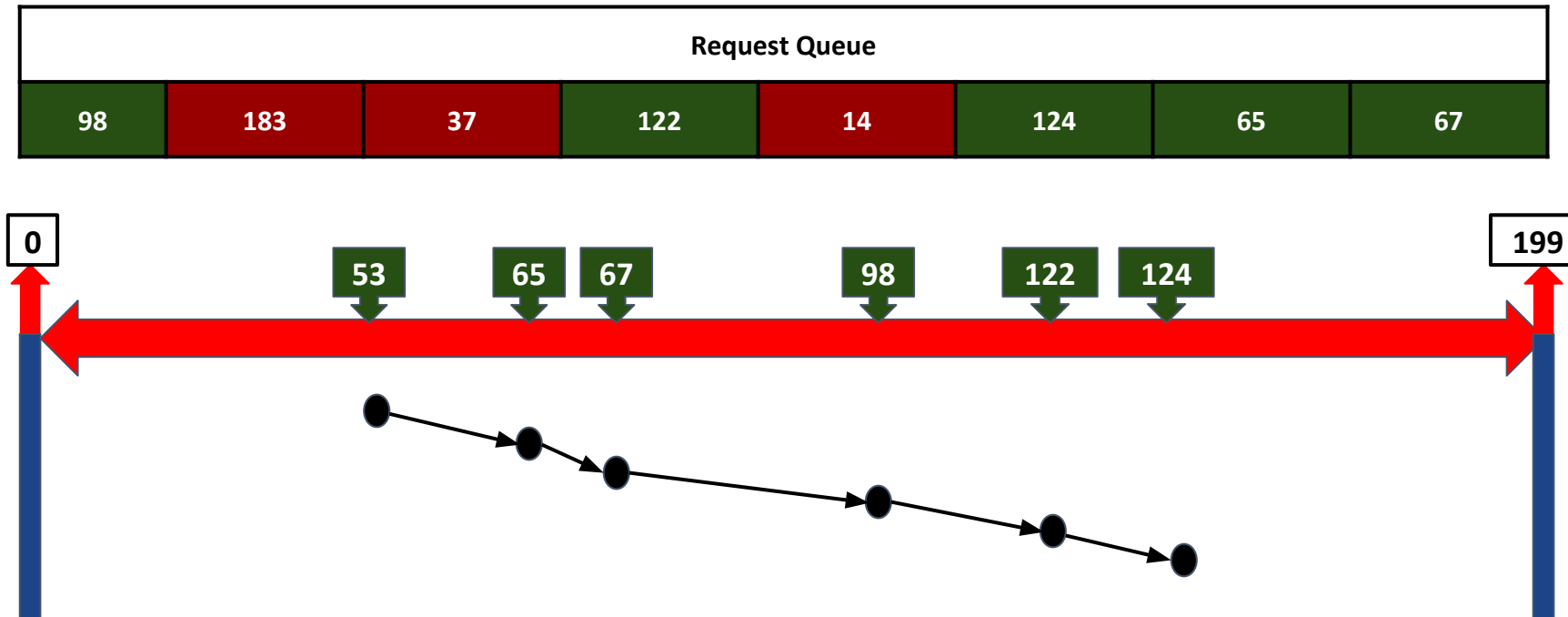
Seek Distance = Seek Distance + abs (Current Cylinder Number - New Cylinder Number)

Seek Distance = 45 + abs(98-122)

Seek Distance = 69

C - LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 69

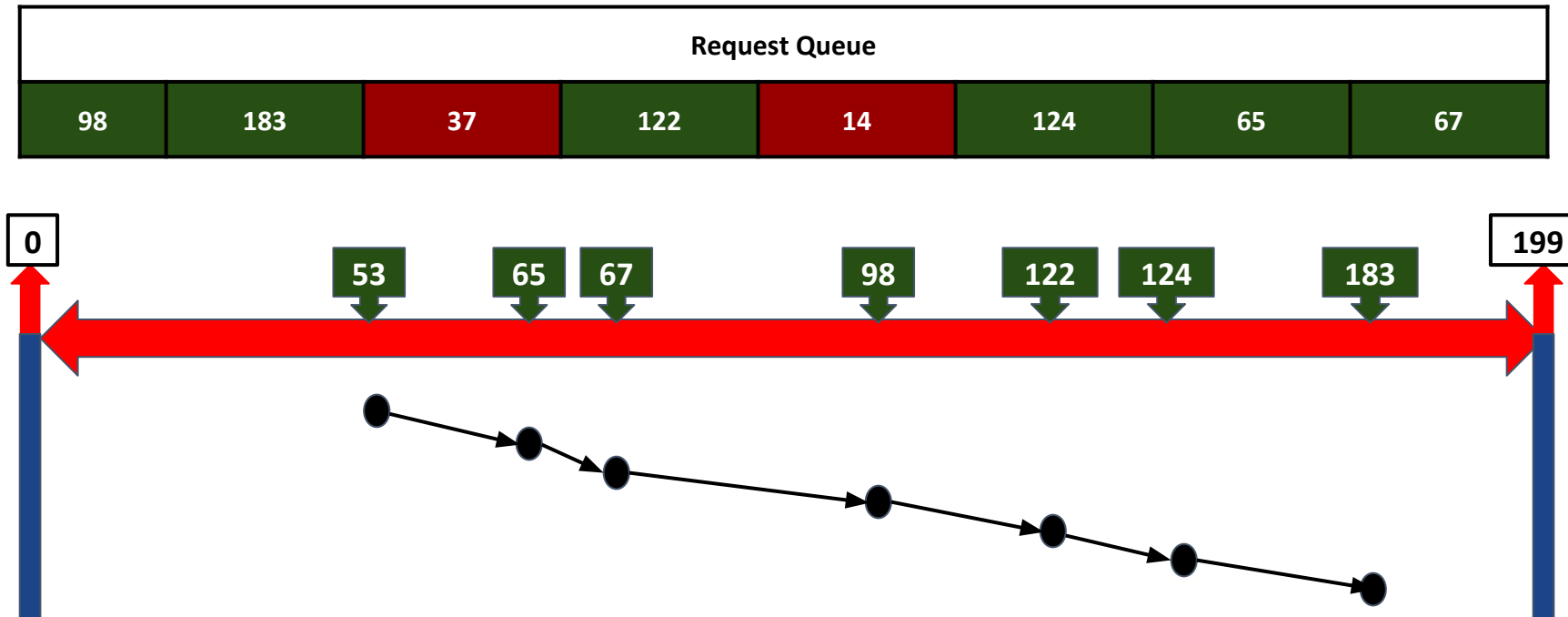
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = $69 + \text{abs}(122 - 124)$

Seek Distance = 71

C - LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 71

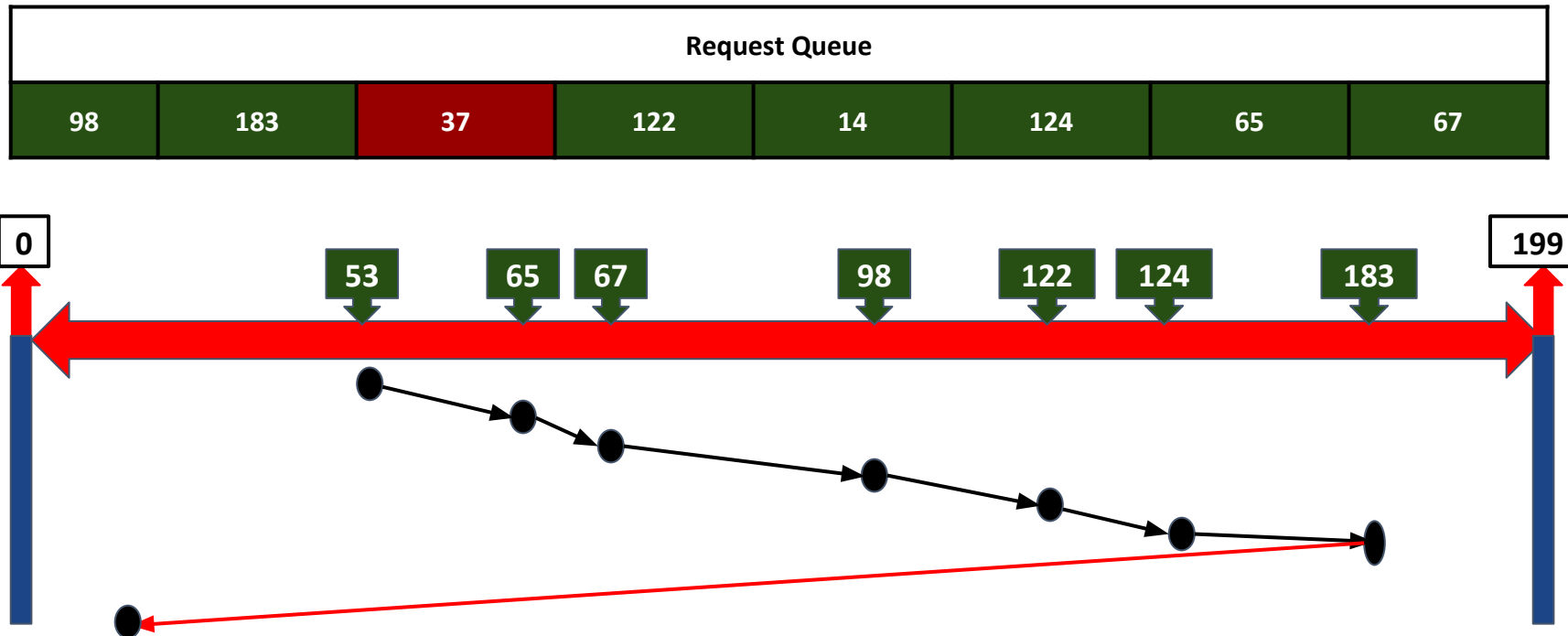
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 69 + $\text{abs}(124 - 183)$

Seek Distance = 130

C - LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 130

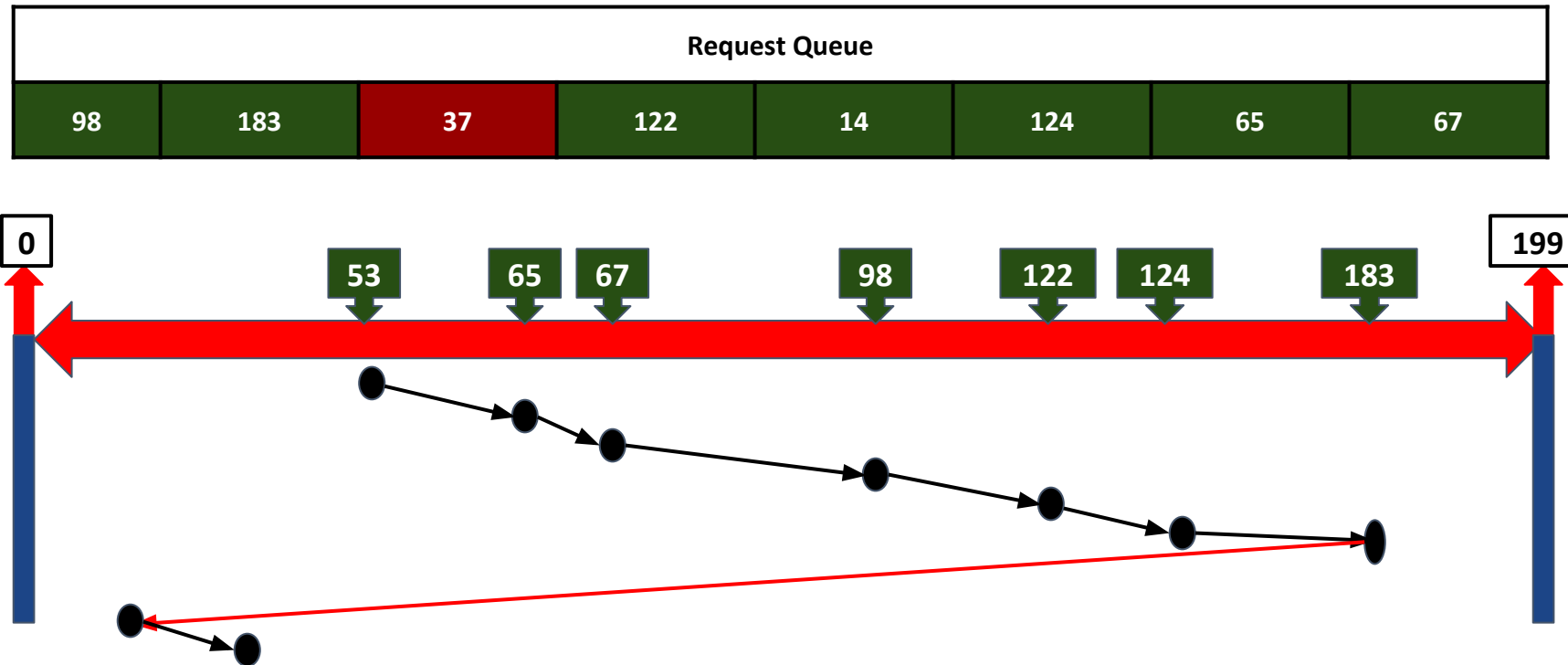
Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 130 + $\text{abs}(183 - 14)$

Seek Distance = 299

C - LOOK Disk Scheduling

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53



Seek Distance = 299

Seek Distance = Seek Distance + $\text{abs}(\text{Current Cylinder Number} - \text{New Cylinder Number})$

Seek Distance = 299 + $\text{abs}(14 - 37)$

Seek Distance = 322

Disk Scheduling Performance - Example

- Request Queue => 0 .. 199
- 98, 183, 37, 122, 14, 124, 65, 67
- Head pointer initially @ 53

FCFS	SSTF	SCAN	C-SCAN	LOOK	C-LOOK
640	236	236	382	208	322

Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
 - Less starvation
 - Performance depends on the number and types of requests
 - Requests for disk service can be influenced by the file-allocation method and metadata layout
- 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
- What about rotational latency?
- Difficult for OS to calculate
- How does disk-based queueing effect OS queue ordering efforts?

- **SCAN Disk Scheduling**
- **C-SCAN Disk Scheduling**
- **LOOK Disc Scheduling**
- **C-LOOK Disc Scheduling**



THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com