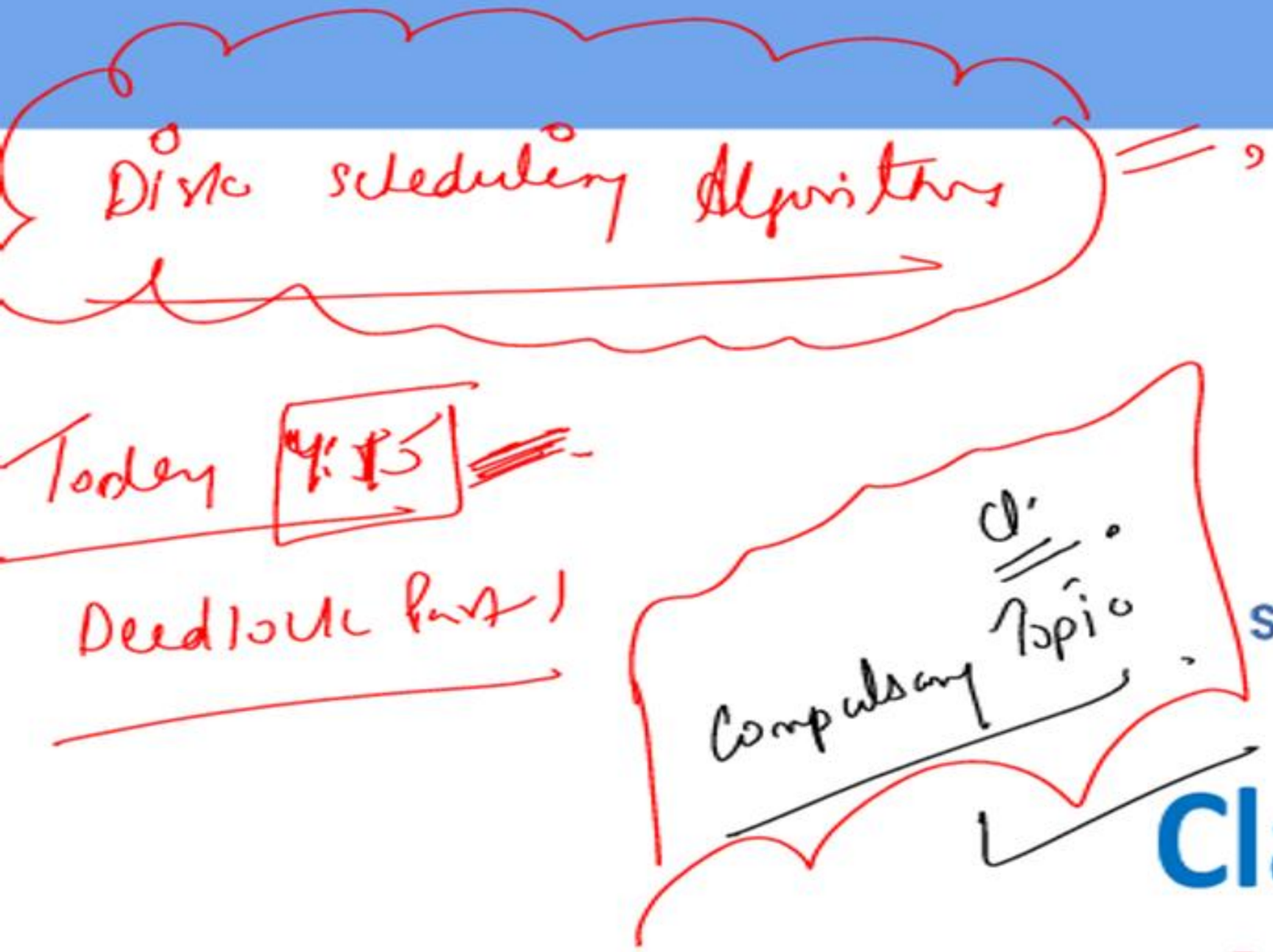


gradeup

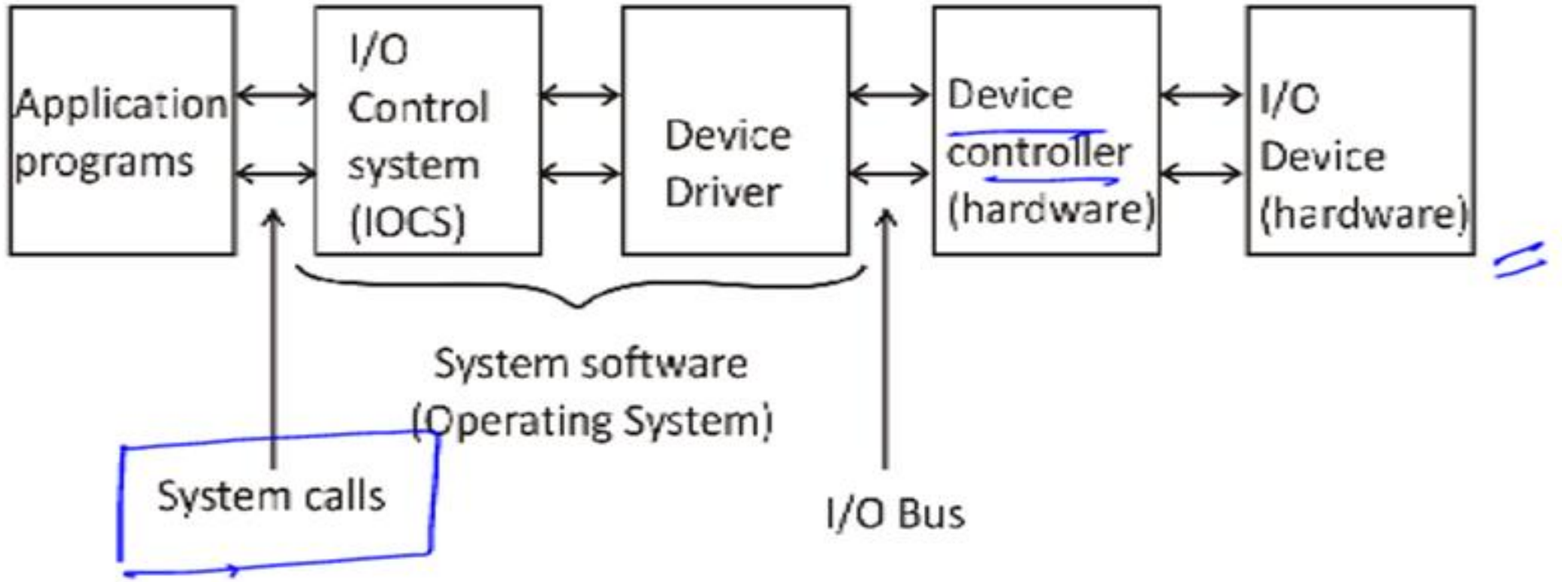
Sahi Prep Hai Toh Life Set Hai

Class 8: Storage Management

Content :-

1. Structure of I/O System
2. Device Controller
3. Buffering
4. Disk Scheduling Algorithm

1. Structure of I/O System :



I/P .

Mouse, Keyboard, Scanner, Microphone

O/P

Monitor, Printer, Speaker.

I/O Devices :-

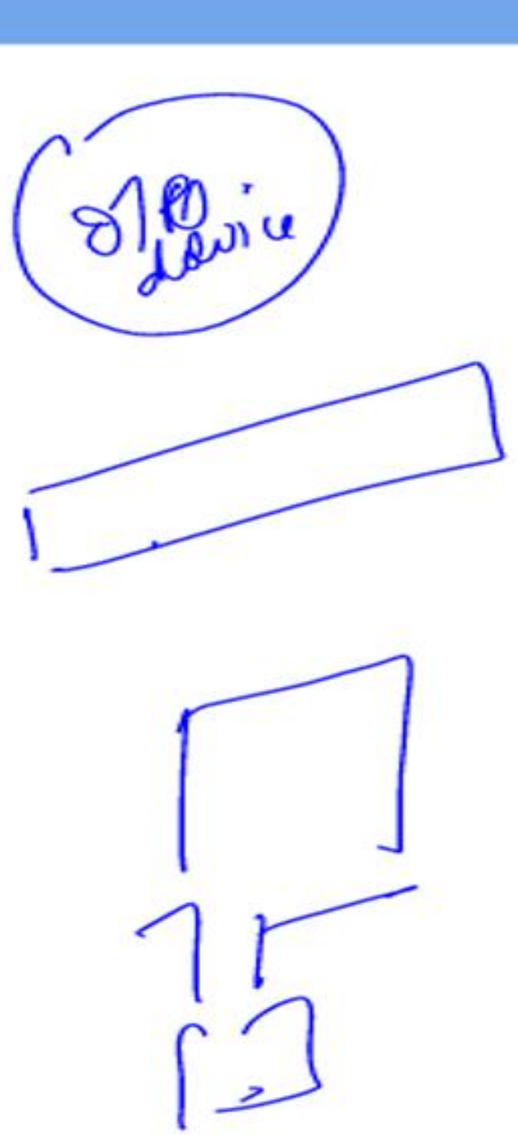
Input Device:- It is an electromechanical device which generates the data for system to read

Output Device :- It is electromechanical device that accepts data from a computer .

Input-only Devices :- Mouse, Keyboard, CD-ROM

Output-only Devices :- Printer , graphical display screen

Input/output Devices :- tape, writable CD , network



Device Controller :- A device controller is the hardware unit which is attached with the input/output bus of the computer and provides the hardware interface between the computer and the input/output devices. On one side it knows how to communicate with the input/output devices and on the other hand it knows how to communicate with the computer system through input/output bus .

DMA: DMA stands for Direct Memory Access , it used when the large volume of data is require to move than it is the efficient technique. The DMA function can be performed by a separate module on the system bus or it can be incorporated into the I/O module.

Buffering :-

It is convenient to perform the input transfers in advance of request being made and to perform output transfer sometimes later after the request has been made . This technique is known as buffering .

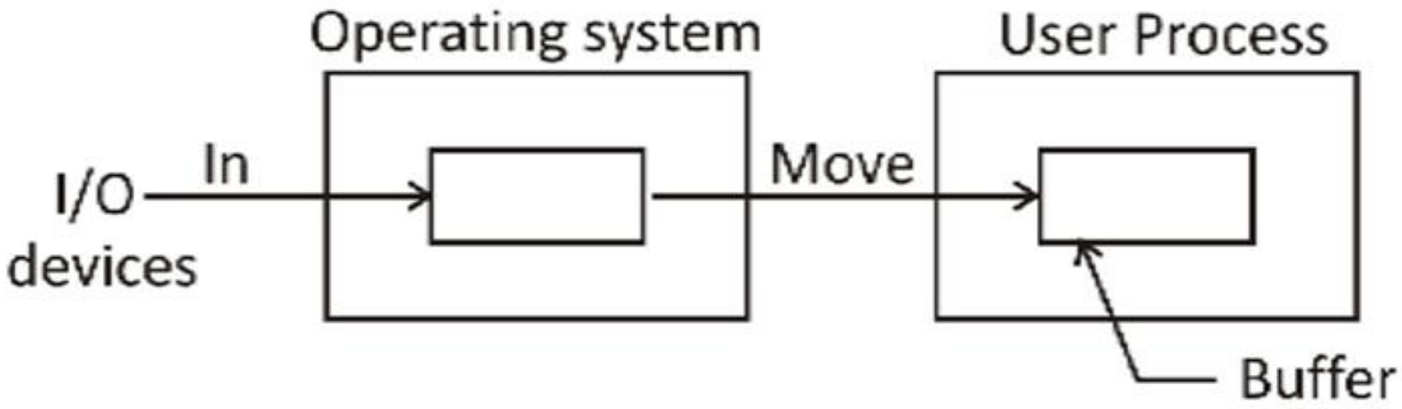
Two type of Input/Output devices :-

I. **Block-oriented devices :-** The I/O devices which store information in block, that are usually of fixed size and transfers are made one block at a time .Generally, the data is referenced by their block number. Disk and tapes are examples of block-oriented devices.

II. **Stream oriented devices** :- These devices transfer data in and out as a stream of bytes. They have NO block structure .Printers , terminals ,mouse, communication ports and other pointing devices that are not secondary storage are stream-oriented.

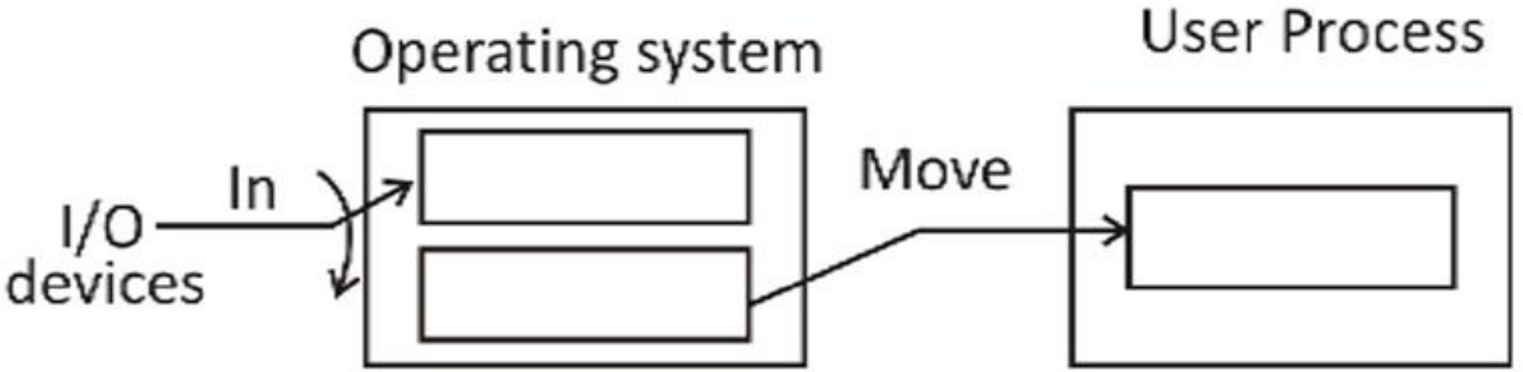
Single Buffering :-

It is the simple type of buffering which is supported by the operating system. When a user process issues an input/output request, the OS assigns a buffer in the system that is a portion of main memory to the operation.



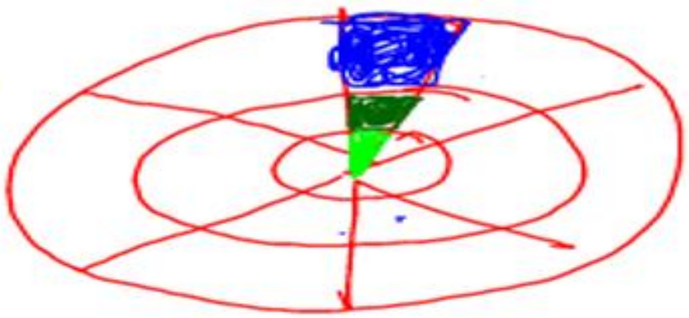
For the stream oriented input/output device, the single buffering scheme can be used as a line-at-a-time function or a byte-at-a-time fashion .

Double Buffering :- An improvement over a single buffering can be made by assigning two system buffer to the input/output operations. A process new transfer data to (or from) one buffer while the operating system empties (or fill) the other. This is known as double buffering or buffer swapping .

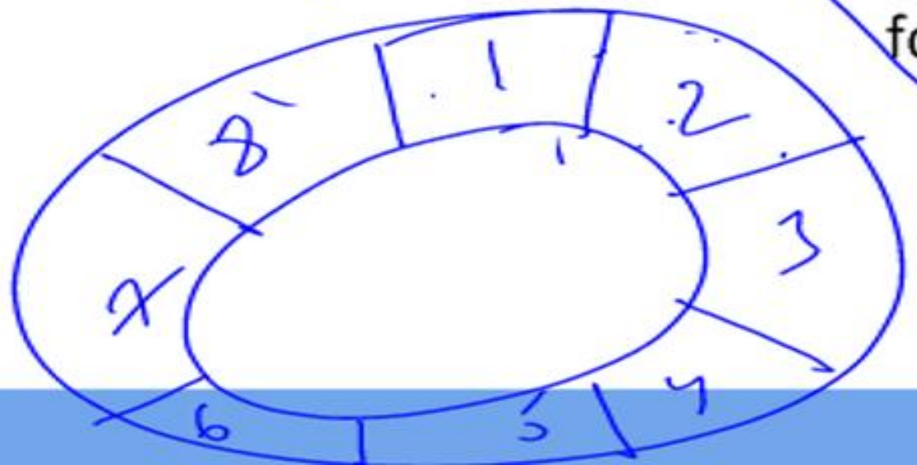


Disk Scheduling Algorithms

V. V. Imp



Tracks



Disk Performance Parameters :-

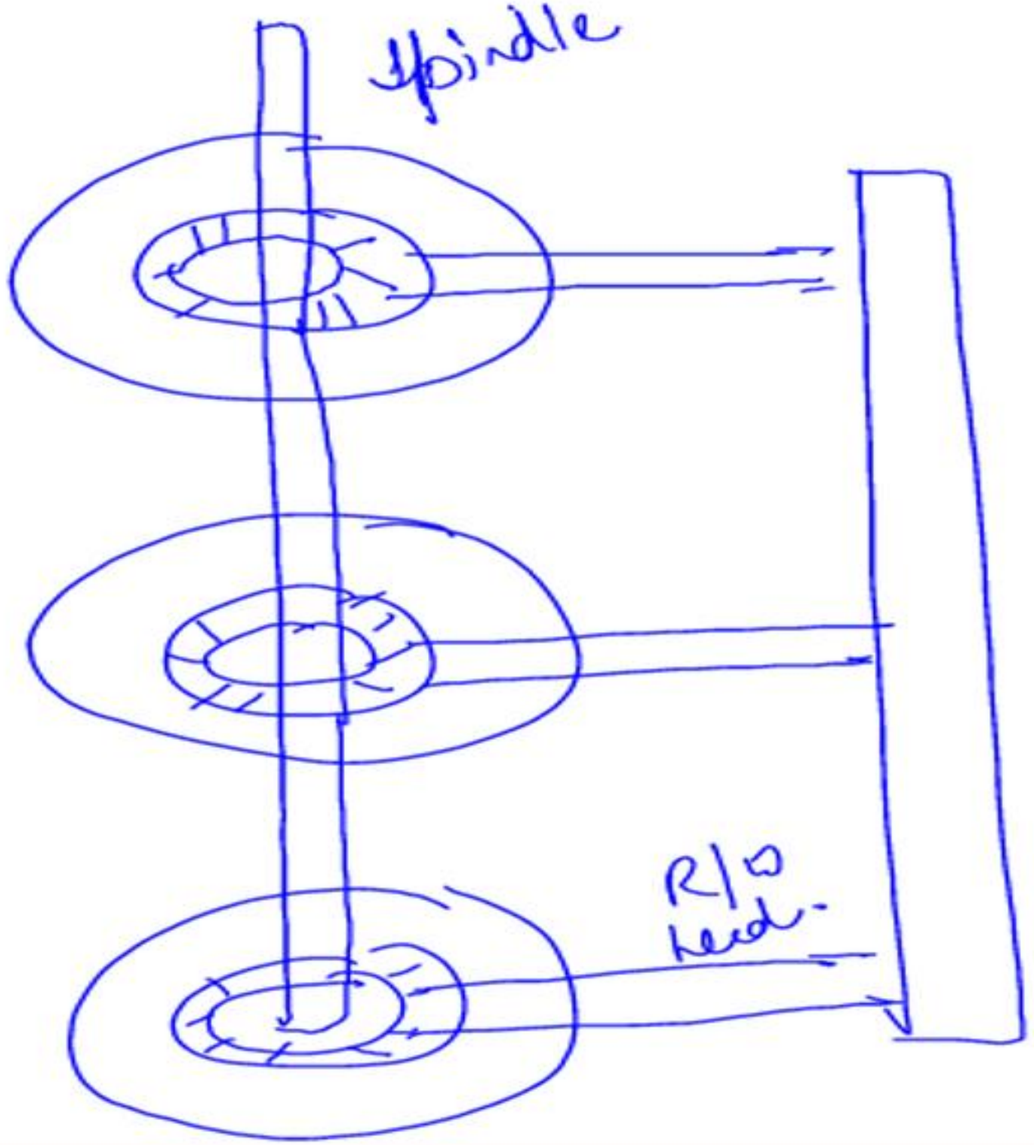
Seek Time : Time required to move the read/write head to an addressed track.

Latency Time :- Time required to bring the starting position of the addressed sector under read/write head.

Access Time :- sum of seek time and latency time ; the time it takes to get into position to read or write

Data Transfer Time :- The data transfer time to or from disk , depends on the rotation speed of the disk in the following fashion :

Seek time + latency + Transfer time


Spindle
R/O
head.

(1) FCFS. order of request is 82, 170, 43, 140, 24, 16, 190

Current position of Read/Write head is at 50.

→ Simplest Disk scheduling Algorithm.

{ Calculate the Total seek time.

→ Requests are addressed in the order in which they arrive.

(1) 643 (2) 642 (3) 644 (4) 640

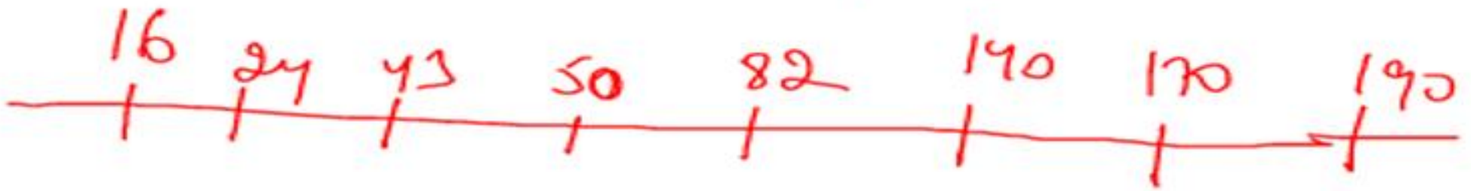
Add all the Differences =.

50 → 82 → 170 → 43 → 140 → 24 → 16 → 190

② SSTF (Shortest Seek time first)

Request with shortest seek time will be executed 1st.
So seek time for every req is calculated in advance.

82, 170, 43, 140, 24, 16, 190 Currently head is at 50



1 ~~208~~
2 209
3 211
4 212

50 → 43 → 24 → 16 → 82 → 140 → 170 → 190

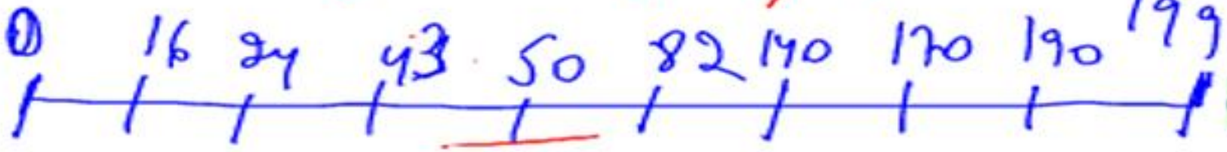
(3) Very imp

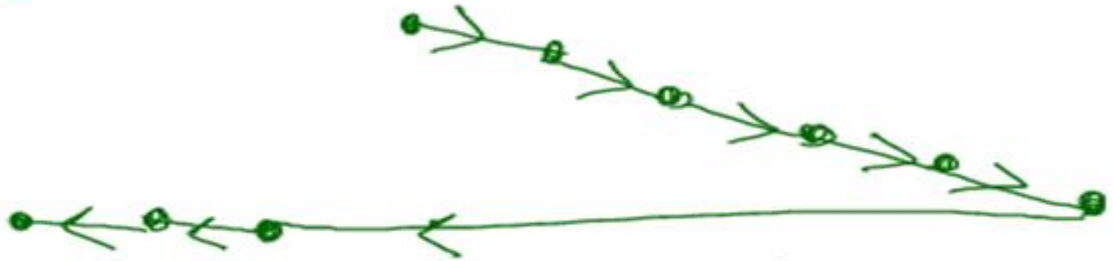
SCAN scheduling :- He move in dir (given in Q) service all request coming in its path & after reaching the end, reverse the dir. & again service request coming in its path.

82, 170, 43, 140, 24, 16, 190

head is at 50.

"Move towards large value"





Short Trick :-

(199 - 50) + (199 - 16)

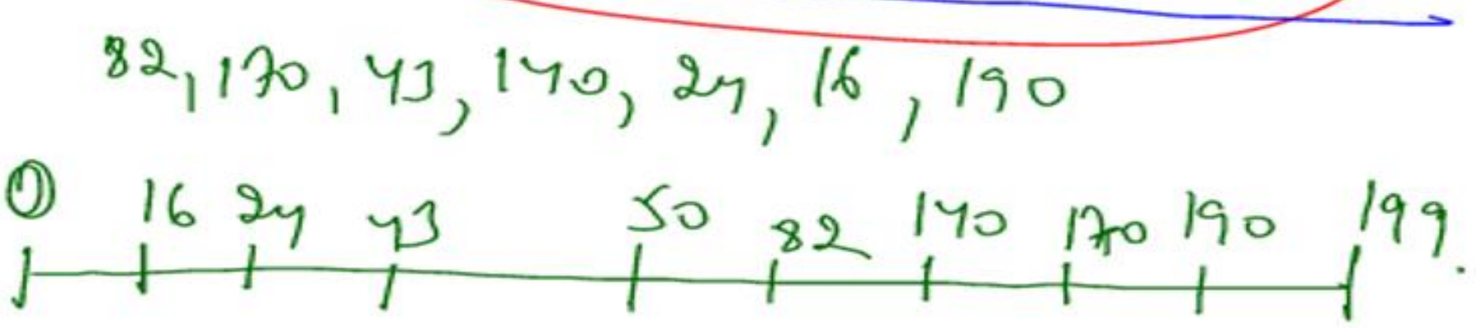
=

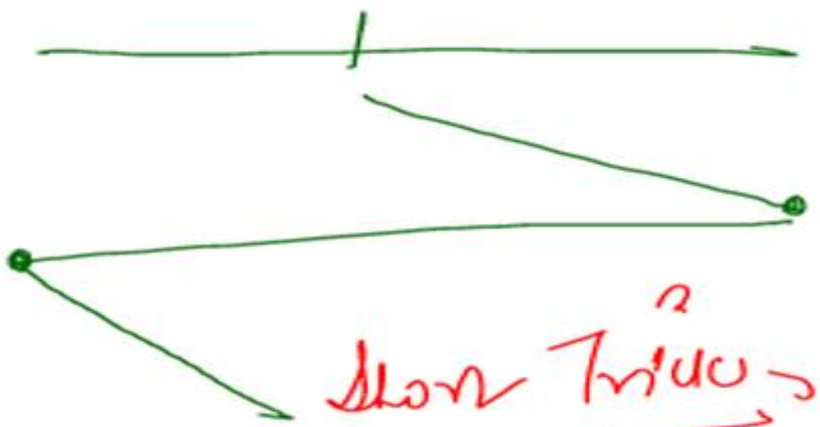
① 331 ② 333 ③ 332 ④ 334

(4) C-SCAN :- Disk arm instead of reversing its direction, goes to the other end of disk and servicing from there.

Move towards larger value

82, 170, 43, 140, 24, 16, 190

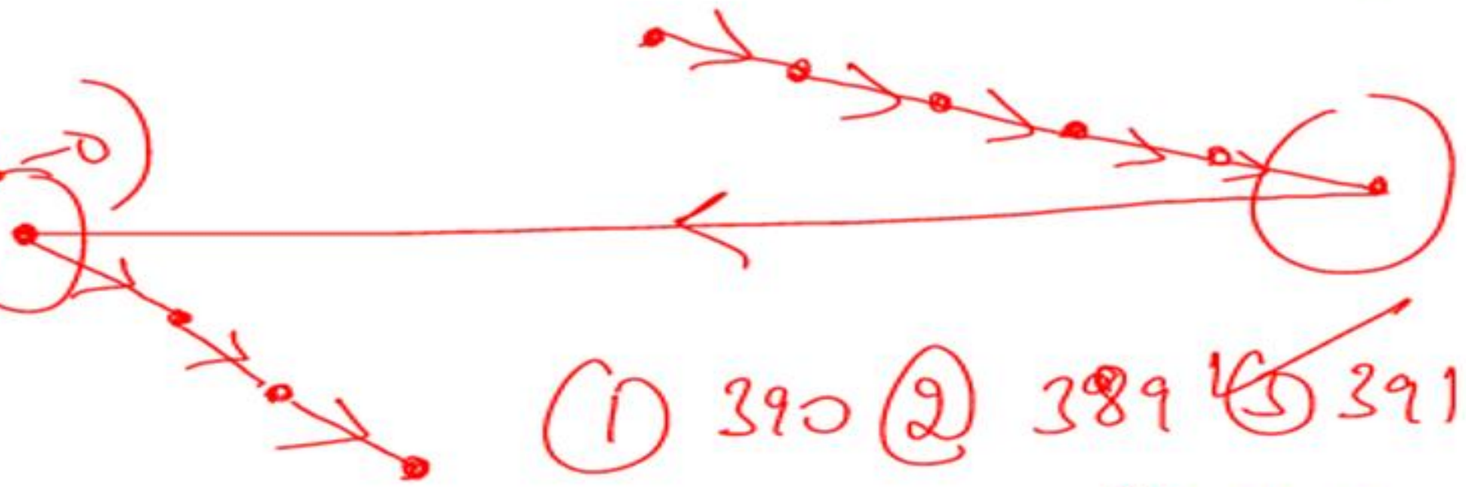




Move Towards

(199 - 50) + (199 - 0) + (43 - 0)

= (391) =



(1) 390 (2) 389 (3) 391

(4) 392

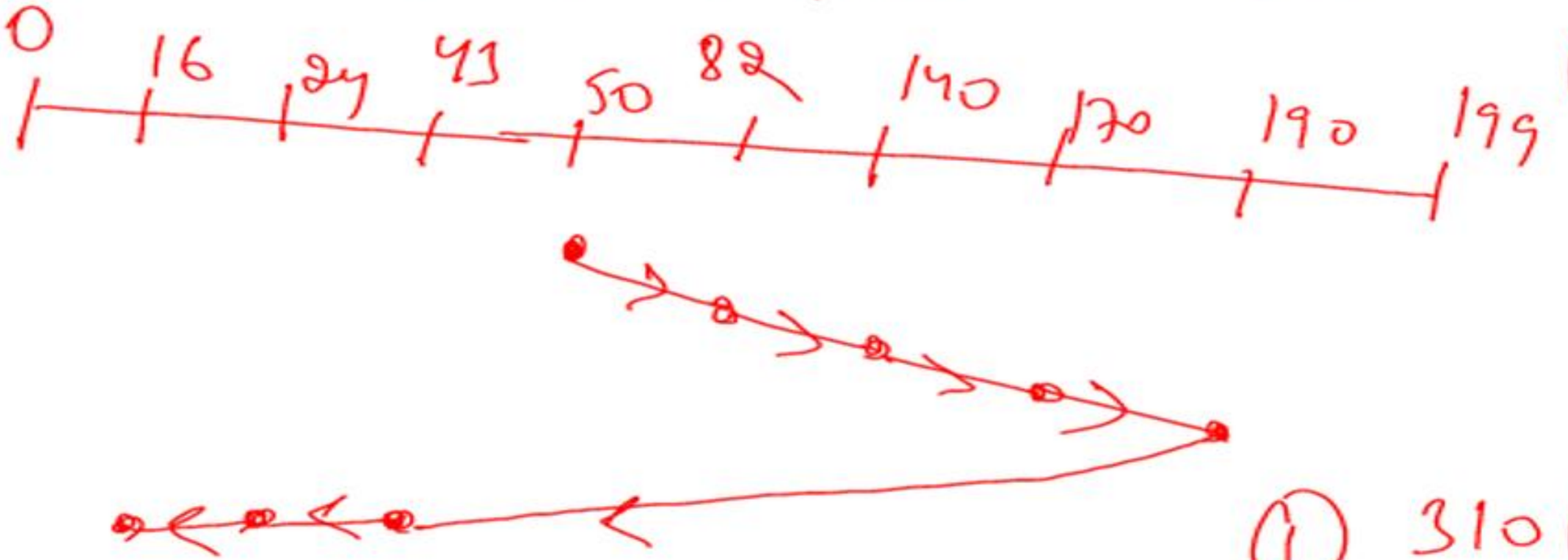
(5) LOOK :- Similar to SCAN.

Disk arm instead of going to the end, will only go to the last request.

82, 170, 43, 140, 24, 16, 190

Move towards Temperature

Short Trick



$(190-50)+(190-16)$

$= (314) = .$

(1) 310 (2) 311 (3) 313

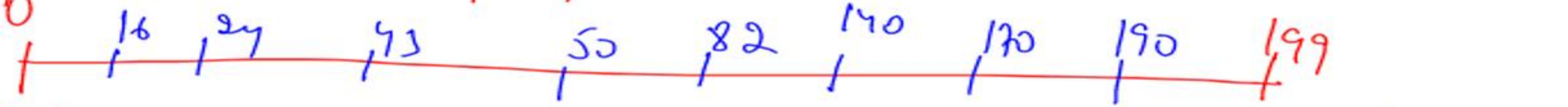
(4) 314

(6) C-Look :- Similar to C-SCAN

Instead of going to the ends it only goes to the last request.

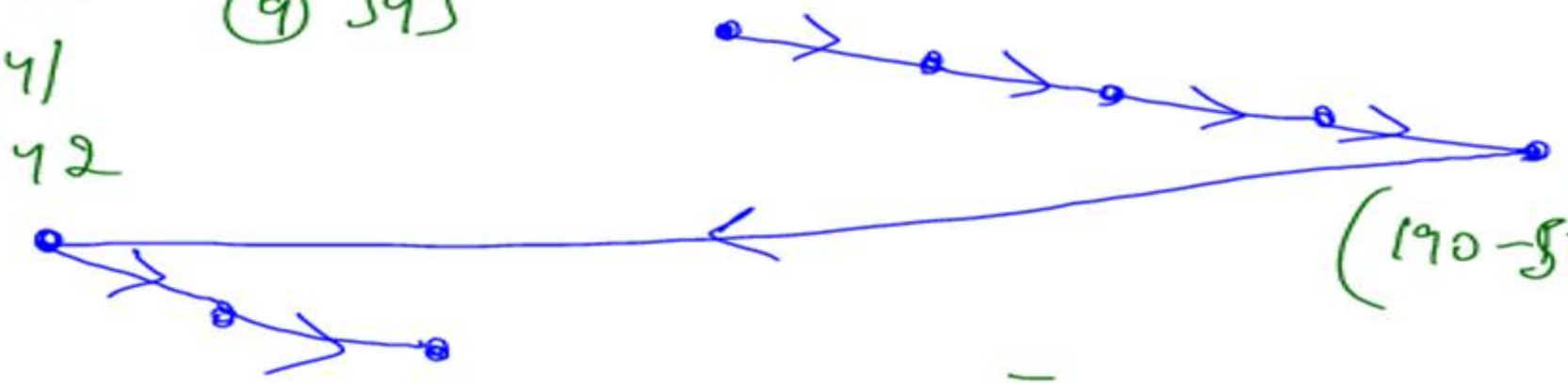
"Move towards larger value"

82, 170, 43, 140, 24, 16, 190



(1) 340
(2) 341
(3) 342
(4) 343

Short Title



(190-50) + (190-16) + (43-16)

=

July 16

Q: If the Disc head is located at track 32, find the no. of disc moves required with FCFS scheduling if queue are :-

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

(1) 320 (2) 322 (3) 321 (4) 319

HW. Jan 17

Q:

6, 10, 12, 54, 97, 73, 128, 15, 44, 110, 34, 45

Head = 23

SCAN.

Total no of cylinders = 150

Disk Scheduling Algorithm :-

1. **First Come First Serve : -** It is the simplest algorithm defined for disk scheduling .
2. **Shortest seek time first :-** It is responsible to serve all the requests close head position , before moving the head for away in service other requests
3. **SCAN Scheduling :-** It is some time called elevator algorithm as disk arm behaves like an elevator in the building . i.e. first serve the requests going up and then reversing to service requests the other way .
4. **Circular Scan(C-Scan) :-** It is the variant of SCAN moves the head from one end of the disk to the other, serving the requests along the way. When it reaches one end it immediately return back towards other end without serving any request on reverse trip.