Date ..../..../.....

Disk Scheduling Algorithms

- FCFS (first come first serve)
- SSTF (shortest seek time)
- SCAN
- LOOK
- C SCAN
- C LOOK

* Goal: To minize the seek time
* Seek Time: Time taken to reach up to desired track

FCFS

- Simple and Straightforwad
- Request served in the order which they arrive
- Process the I/O requests one by one, from head of the queue to the end without any consideration for the location of the data on the disk

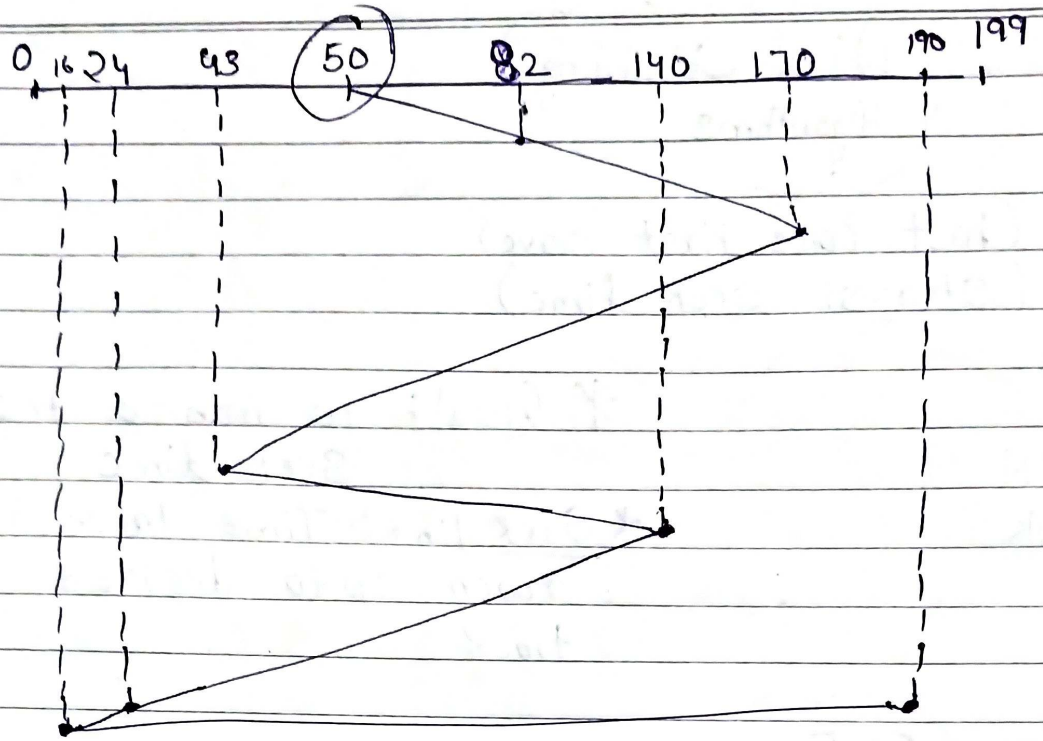
Advantages
- Simple implementation
- Fairness

Limitations
- Possibility of starvation
- High avg. seek time
- No optimization

Q) A disk contains 200 tracks (0-199) request queue contains track no. 82, 170, 43, 140, 24, 16, 190 respectively. Current position of R/W head = 50 Calculate total no. of tracks. movement by R/W head

Spiral

Date ..../..../.....



Seek time = (82 - 50) + (170 - 82) + (170 - 43)
+ (140 - 43) + (140 - 24) + (24 - 16)
+ (190 - 16) + (170 - 50) + (170 - 43)
+ (140 - 43) + (140 - 16)

= 642

Date ..../..../.....

SSTF

- priorities servicing the I/O request that is closet to the current head position of the disk arm.
- minimize the seek time

Advantages
1) Reduced seek time
2) Improved throughput

Limitations
1) No Guarantees for Best Performance
2) Possibity of head Thrashing
3) Potential for Starvation

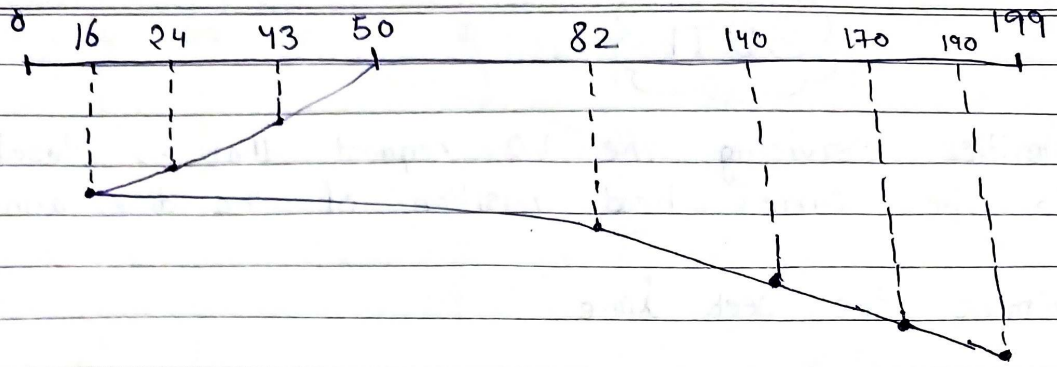
Q A disk contains 200 tracks (0-199) request queue contains track no. 82, 170, 43, 140, 24, 16, 190 respectively. Current position of R/W head = 50

→ Calculate total no of trends movement ~~by~~ by R/W head using Shortest seek time first?

→ If R/W head takes 1ms to move from one track to another then total time taken?

Spiral

Date ..../..../.....



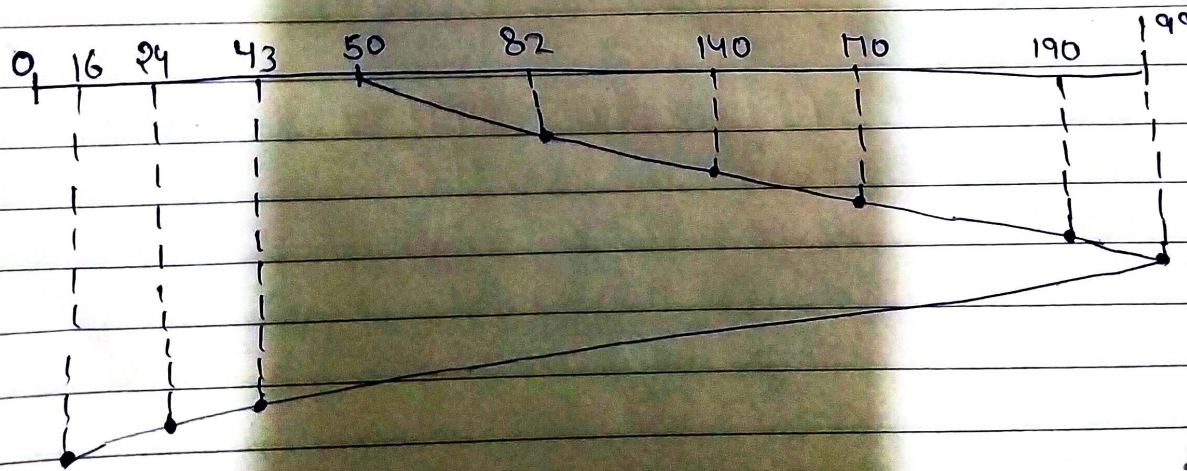
Seek time = (50-43) + (43-24) + (24-16) + (82-16)
(140-82) + (170-140) + (190-170)
= 208

SCAN

- Elevator algo
- moves the disk arm in sweeping motion
- elevator travelling from one end of the floor to other and then reversing its direction

Advantages
- Prevents starvation
- Simple implementation
- Efficient disk arm utilization
- reduce avg. seek time

Disadvantages
1) Possible increased wait time for edge requests
2) No consideration for request priority



Spiral

Date ..../..../.....

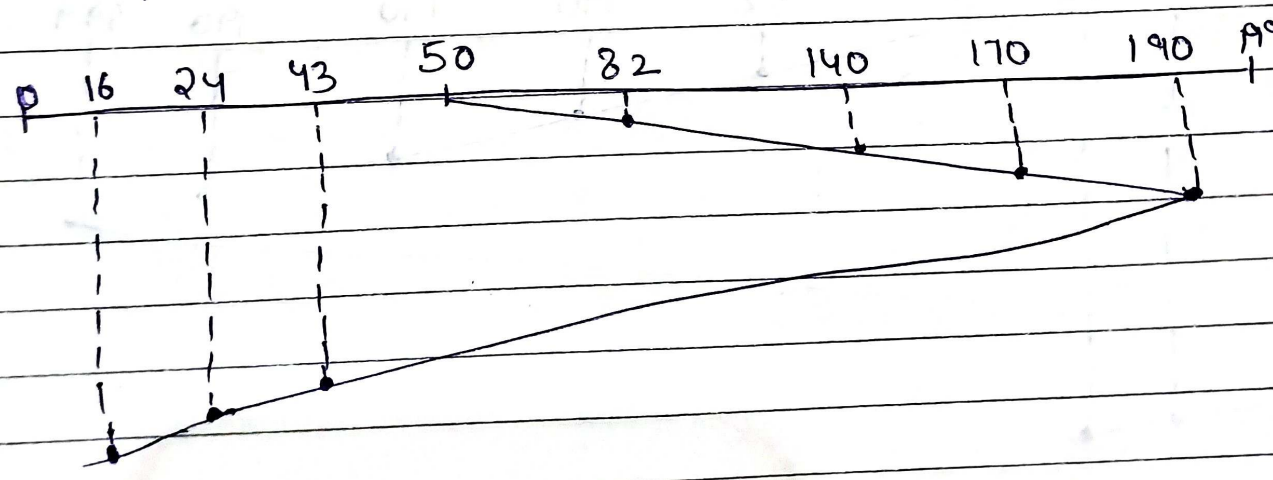
Seek time = (199 - 50) + (199 - 16)
= 332

LOOK

- Similar to SCAN
- Doesn't touch the end of the disk
- only goes till last request
- extension of SCAN

Advantages
- Reduce seek time
- Efficient use of DISK arm
- Improve disk performance

Limitation
1) No priority consideration
2) Potential Starvation
3) Complexity in dynamic work loads



Seek time => (190 - 150) + (190 - 16)
= 314

Date ..../.... /.....

C SCAN

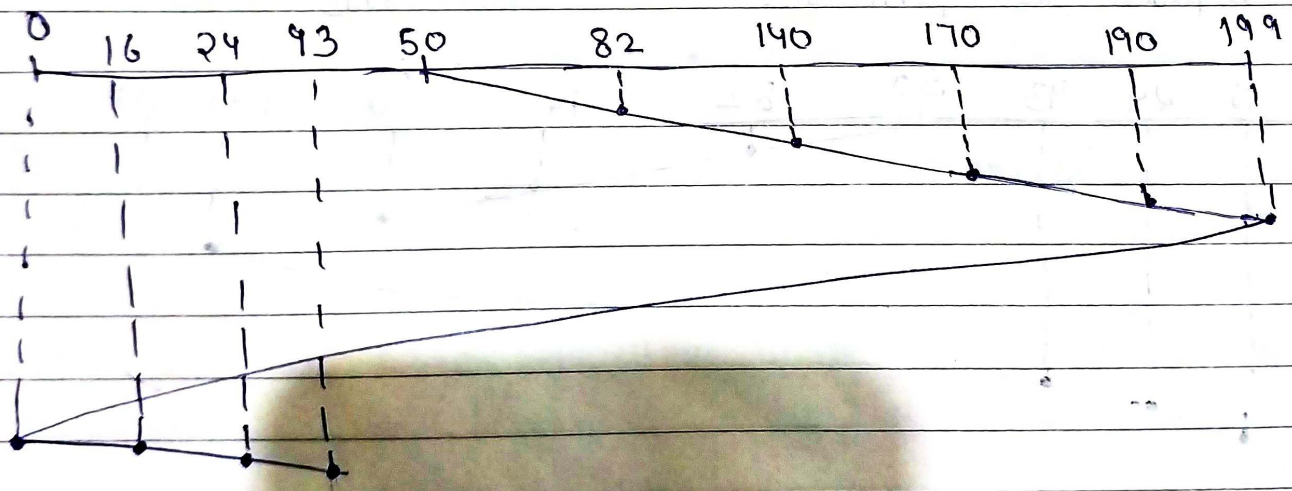
- Serves in one direction till the last head and jump back to start of the disk without serving any request in a path

Advantages

- Reduce waiting time for edge requests
- Efficient disk arm movement
- Prevent Stavation at disk ends

Disadvantages

- Lack of priority handling
- Potential for increased wait time
- Not for dynamic workloads



Seek time = (199 - 50) + (199 - 0) + (43 - 0)
= 391

Spiral

Date ..../..../.....

C LOOK

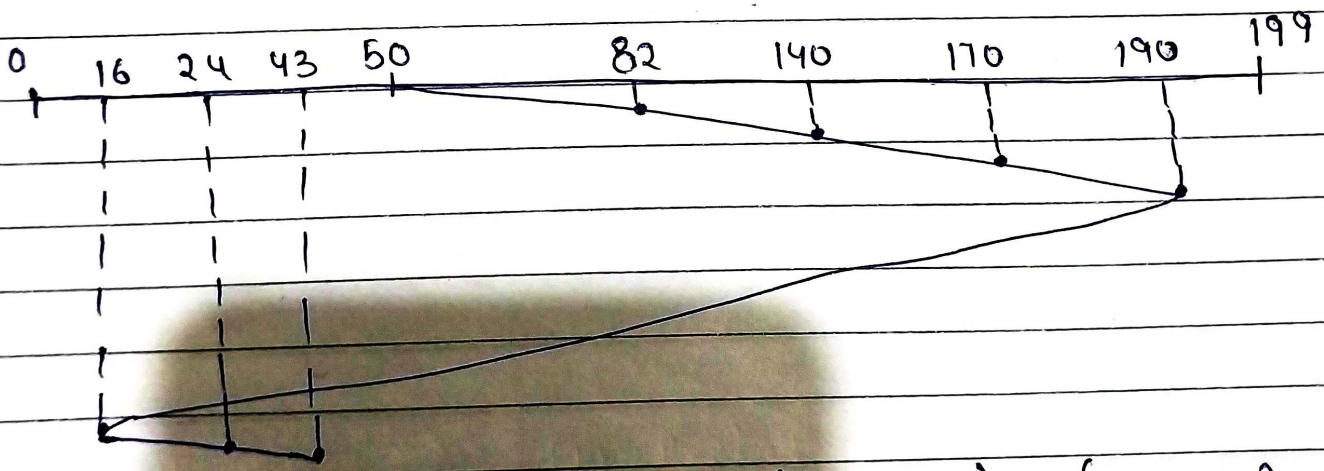
- Serves till the last request and the jump back to the lowest request head (without serving any request in a path) and the serves remaining request as go.

Advantages

1) Reduced seek time within defined range
2) Efficient disk arm movement
3) Prevent starvation in specified range

Disadvantages

1) Limited coverage area
2) No priority handling
3) Lack of adaptability to dynamic work loads



Seek time = (190 - 50) + (190 - 16) + (43 - 16)
= 341