



 [Home](#) →

## User-Login

[Forgot password?](#) or [Sign Up](#) →  
[Admin](#) →



localhost/project/admin login.php



 [Home](#) →

## Admin-Login

**Login** →

User Login →



localhost/project/signup.php



 [Home](#) →

## SignUp

[SignUp](#) →

[Forgot password? or Login](#) →



localhost/project/home.php



## Disk Scheduling Algorithm's

[Login](#)[Logout](#)

Operating System  
Disk Scheduling Algorithm's

[Know more](#)

localhost/project/admin\_dash.php?success=logged

Logout

ID	Username	Email
**	*****	*****
**	*****	*****
**	*****	*****
**	*****	*****



localhost/project/info.php



 [Home](#) →

## About U



**Pooja RD**  
**PES1PG21CA057**  
**Balsamiq**



**Tejas P Srivatsa**  
**PES1PG21CA09**  
**HTML and CSS**



**Keval Vaghasiya**  
**PES1PG21CA09**  
**CSS,JS,and PHP**



**Ramkumar HR**  
**PES1PG21CA06**  
**Balsamiq,Desig**

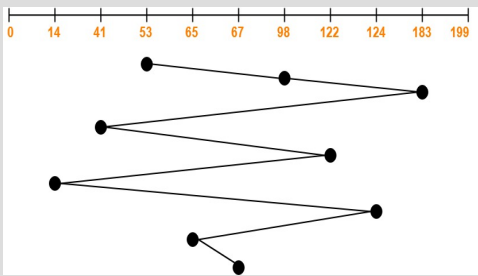


Home

About

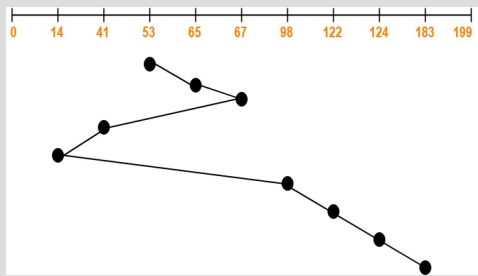
Disk scheduling is done by operating systems to schedule I/O requests arriving for the disk. Disk scheduling is also known as I/O scheduling. Disk scheduling is important because: Multiple I/O requests may arrive by different processes and only one I/O request can be served at a time by the disk controller.

First Come First Serv



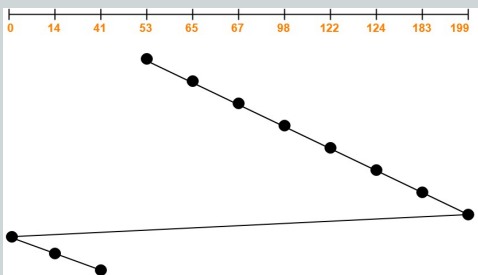
Demo

Shortest Seek Time First



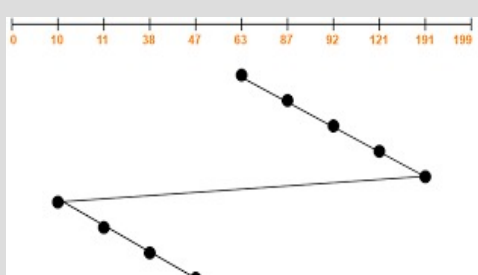
Demo

Circular - Scan



Demo

Circular - Look



Demo



localhost/project/sstf.php

[Home](#)

## SHORTEST SEEK TIME FIRST

In SSTF (Shortest Seek Time First), requests having shortest seek time are executed first. So, the seek time of every request is calculated in advance in the queue and then they are scheduled according to the calculated seek time. As a result, the request near the disk arm will get executed first. SSTF is certainly an improvement over FCFS as it decreases the average response time and increases the throughput of system

### Advantages:

- Average Response Time decreases
- Throughput increases

### Disadvantages:

- Overhead to calculate seek time in advance
- Can cause Starvation for a request if it has higher seek time as compared to incoming requests
- High variance of response time as SSTF favours only some requests

Enter order of request sequence:

Add

Remove

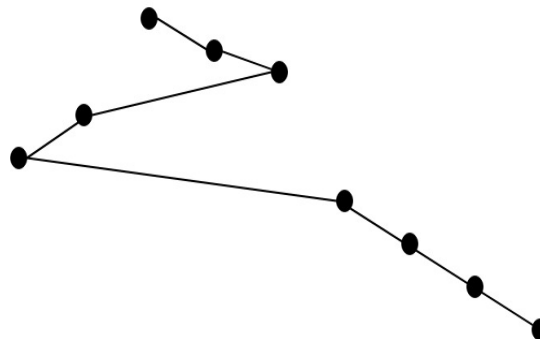
Enter read/write head position:

SUBMIT IT

### Adding & Updating datapoints

Secondary X Axis

0 14 41 53 65 67 98 122 124 183 199







localhost/project/fcfs.php

[Home](#)

## FIRST COME FIRST SERVE

FCFS is the simplest disk scheduling algorithm. As the name suggests, this algorithm entertains requests in the order they arrive in the disk queue. The algorithm looks very fair and there is no starvation (all requests are serviced sequentially) but generally, it does not provide the fastest service.

### Advantages:

- Every request gets a fair chance
- No indefinite postponement

### Disadvantages:

- Does not try to optimize seek time
- May not provide the best possible service

Enter order of request sequence:

Add

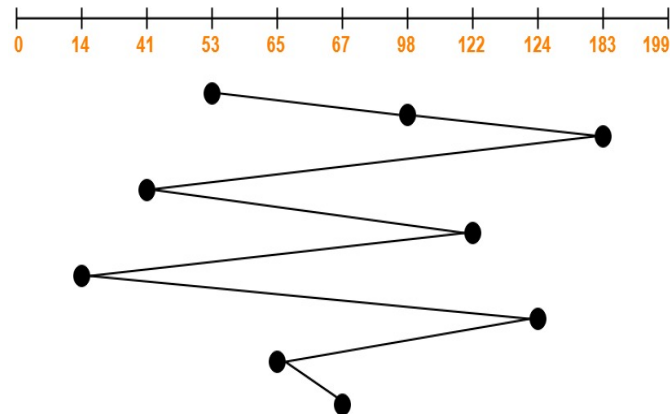
Remove

Enter read/write head position:

SUBMIT IT

### Adding & Updating datapoints

Secondary X Axis





localhost/project/c-scan.php

[Home](#)

## CIRCULAR SCAN

In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area. These situations are avoided in CSCAN algorithm in which the disk arm instead of reversing its direction goes to the other end of the disk and starts servicing the requests from there. So, the disk arm moves in a circular fashion and this algorithm is also similar to SCAN algorithm.

### Advantages:

- C-SCAN Algorithm is the successor and the improved version of the SCAN scheduling Algorithm
- Uniform waiting time is provided
- Better response time is provided

### Disadvantages:

- More seek movements are caused in C-SCAN compared to Algorithm.
- In C-SCAN even if there are no requests left to be serviced the Head will still travel to the end of the disk unlike SCAN algorithm.

Enter order of request sequence:

Add

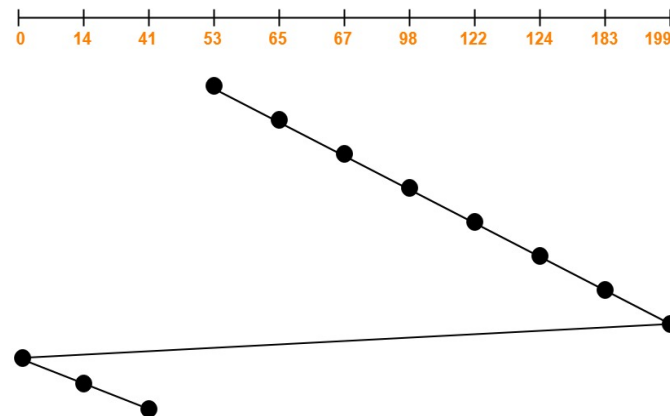
Remove

Enter read/write head position:

SUBMIT IT

### Adding & Updating datapoints

Secondary X Axis





localhost/project/c-look.php

[Home](#)

## CIRCULAR LOOK

As LOOK is similar to SCAN algorithm, in similar way, CLOOK is similar to CSCAN disk scheduling algorithm. In CLOOK, the disk arm in spite of going to the end goes only to the last request to be serviced in front of the head and then from there goes to the other end's last request. Thus, it also prevents the extra delay which occurred due to unnecessary traversal to the end of the disk..

### Advantages:

- In C-LOOK the head does not have to move till the end of the disk if there are no requests to be serviced
- Better performance is provided compared to SCAN Algorithm
- Starvation is avoided in C-LOOK

### Disadvantages:

- In C-LOOK an overhead of finding the end requests is present.

Enter order of request sequence:

Add

Remove

Enter read/write head position:

SUBMIT IT

### Adding & Updating datapoints

Secondary X Axis

