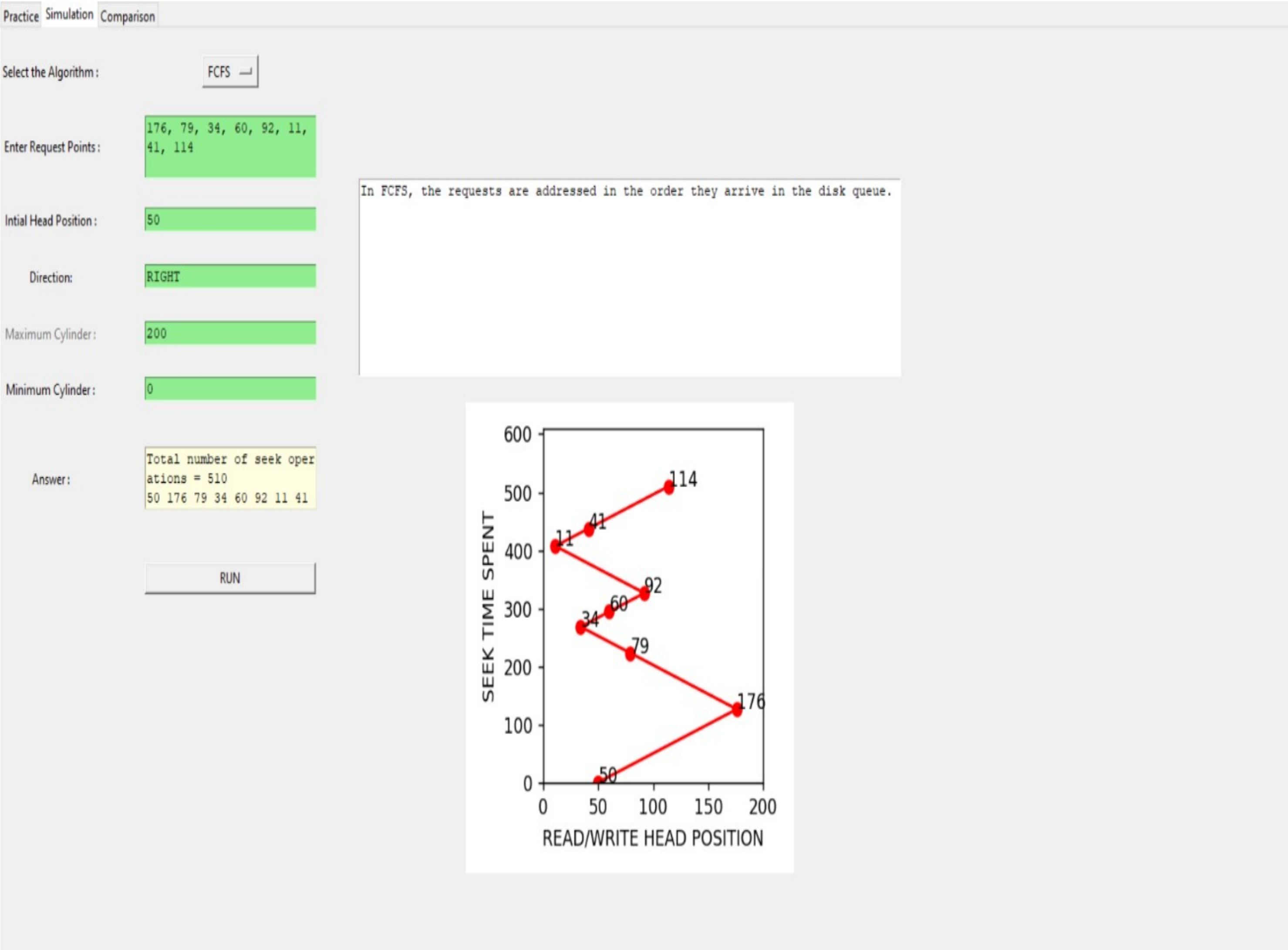
1)First Come First Serve (FCFS):

Algorithm:

- Let Request array represents an array storing indexes of tracks that have been requested in ascending order of their time of arrival. ‘head’ is the position of disk head.
- Let us one by one take the tracks in default order and calculate the absolute distance of the track from the head.
- Increment the total seek count with this distance.
- Currently serviced track position now becomes the new head position.
- Go to step 2 until all tracks in request array have not been serviced.



Select the Algorithm : FCFS

Enter Request Points : 176, 79, 34, 60, 92, 11, 41, 114

Intial Head Position : 50

Direction: RIGHT

Maximum Cylinder : 200

Minimum Cylinder : 0

```
In FCFS, the requests are addressed in the order they arrive in the disk queue.
```

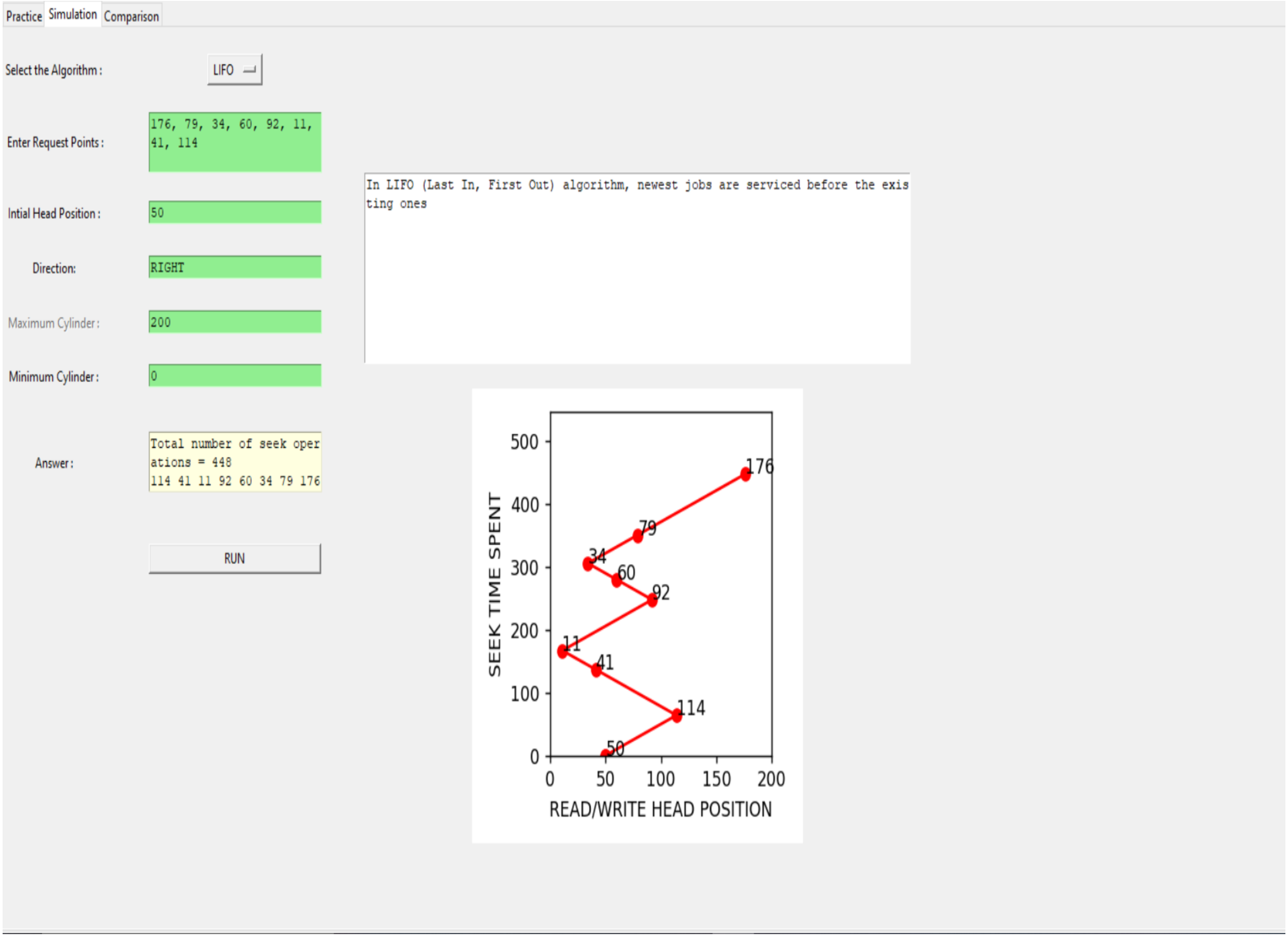
Answer :

```
Total number of seek oper
ations = 510
50 176 79 34 60 92 11 41
```

2)Last In First Out (LIFO):

Algorithm:

- Let Request array represents an array storing indexes of tracks that have been requested in descending order of their time of arrival. ‘head’ is the position of disk head.
- Let us one by one take the tracks in order and calculate the absolute distance of the track from the head.
- Increment the total seek count with this distance.
- Currently serviced track position now becomes the new head position.
- Go to step 2 until all tracks in request array have not been serviced.



Select the Algorithm : LIFO

Enter Request Points : 176, 79, 34, 60, 92, 11, 41, 114

Intial Head Position : 50

Direction: RIGHT

Maximum Cylinder : 200

Minimum Cylinder : 0

Answer :

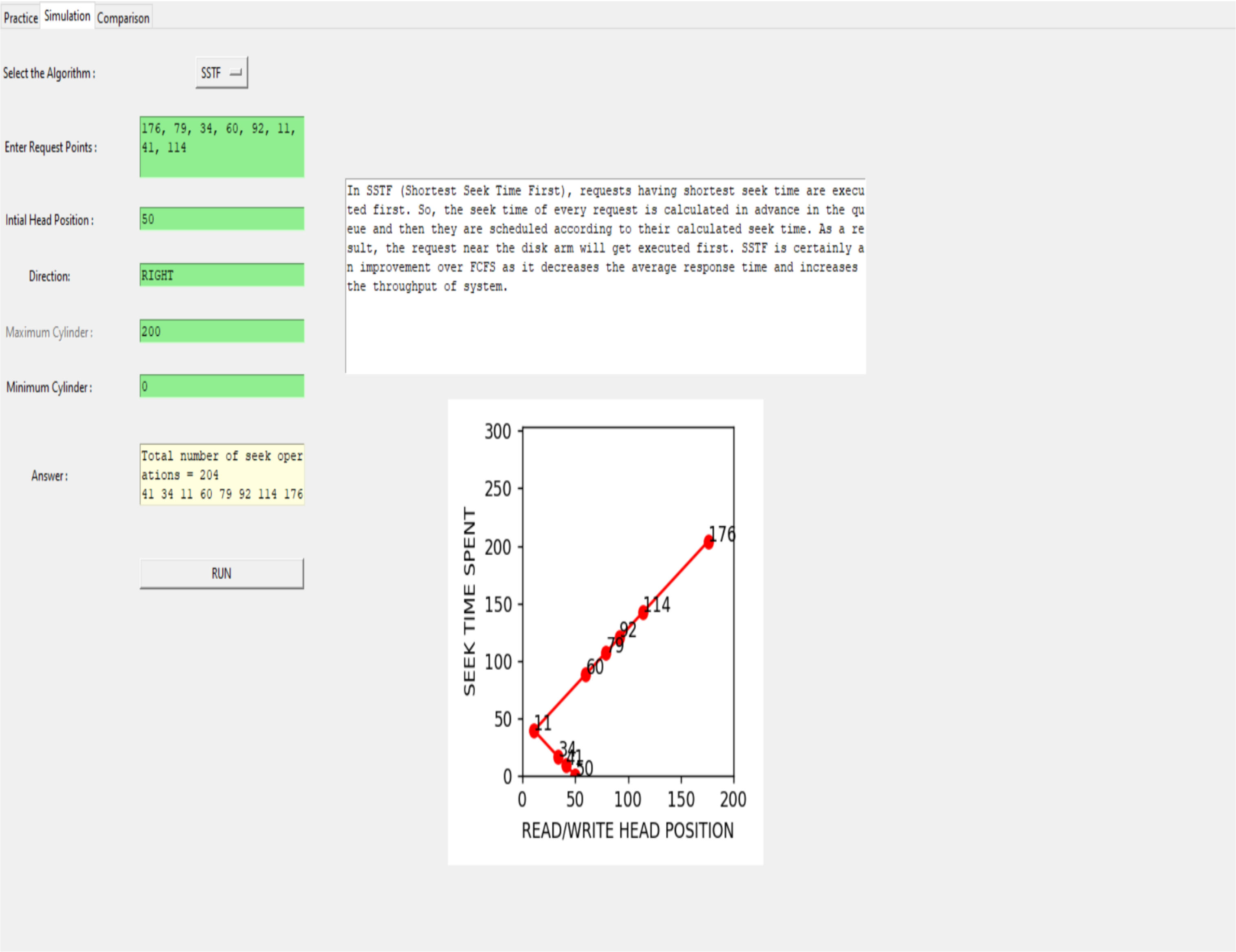
```
Total number of seek operations = 448
114 41 11 92 60 34 79 176
```

```
In LIFO (Last In, First Out) algorithm, newest jobs are serviced before the existing ones
```

3)Shortest Seek Time First (SSTF):

Algorithm:

- Let Request array represents an array storing indexes of tracks that have been requested. ‘head’ is the position of disk head.
- Find the positive distance of all tracks in the request array from head.
- Find a track from requested array which has not been accessed/serviced yet and has minimum distance from head.
- Increment the total seek count with this distance.
- Currently serviced track position now becomes the new head position.
- Go to step 2 until all tracks in request array have not been serviced.



Select the Algorithm : SSTF

Enter Request Points : 176, 79, 34, 60, 92, 11, 41, 114

Intial Head Position : 50

Direction: RIGHT

Maximum Cylinder : 200

Minimum Cylinder : 0

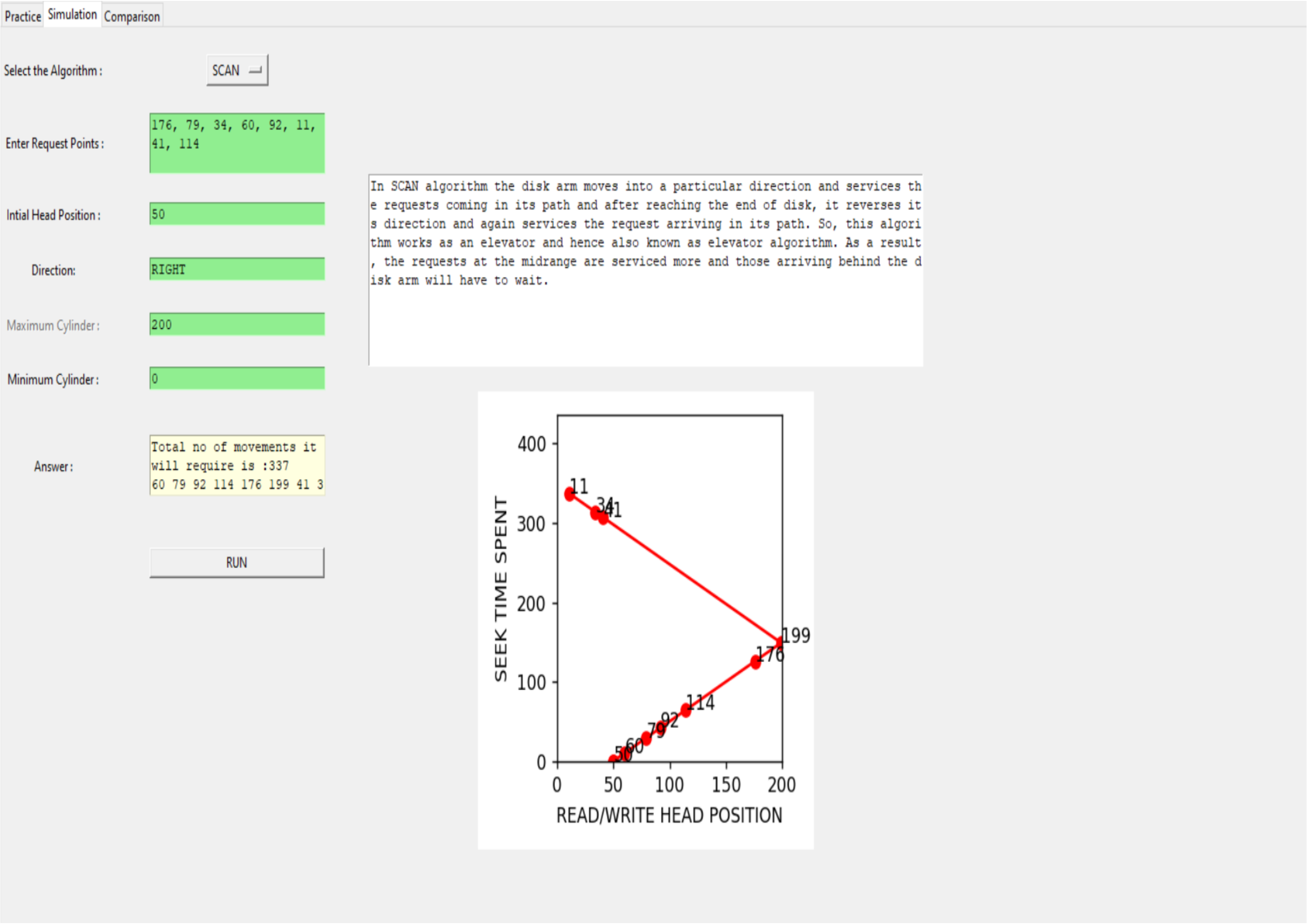
Answer : Total number of seek operations = 204
41 34 11 60 79 92 114 176

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 their 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.

4) SCAN:

Algorithm:

- Let Request array represents an array storing indexes of tracks that have been requested in ascending order of their time of arrival. ‘head’ is the position of disk head.
- Let direction represents whether the head is moving towards left or right.
- In the direction in which head is moving service all tracks one by one.
- Calculate the absolute distance of the track from the head.
- Increment the total seek count with this distance.
- Currently serviced track position now becomes the new head position.
- Go to step 3 until we reach at one of the ends of the disk.
- If we reach at the end of the disk reverse the direction and go to step 2 until all tracks in request array have not been serviced.



Select the Algorithm : SCAN

Enter Request Points : 176, 79, 34, 60, 92, 11, 41, 114

Intial Head Position : 50

Direction: RIGHT

Maximum Cylinder : 200

Minimum Cylinder : 0

Answer :

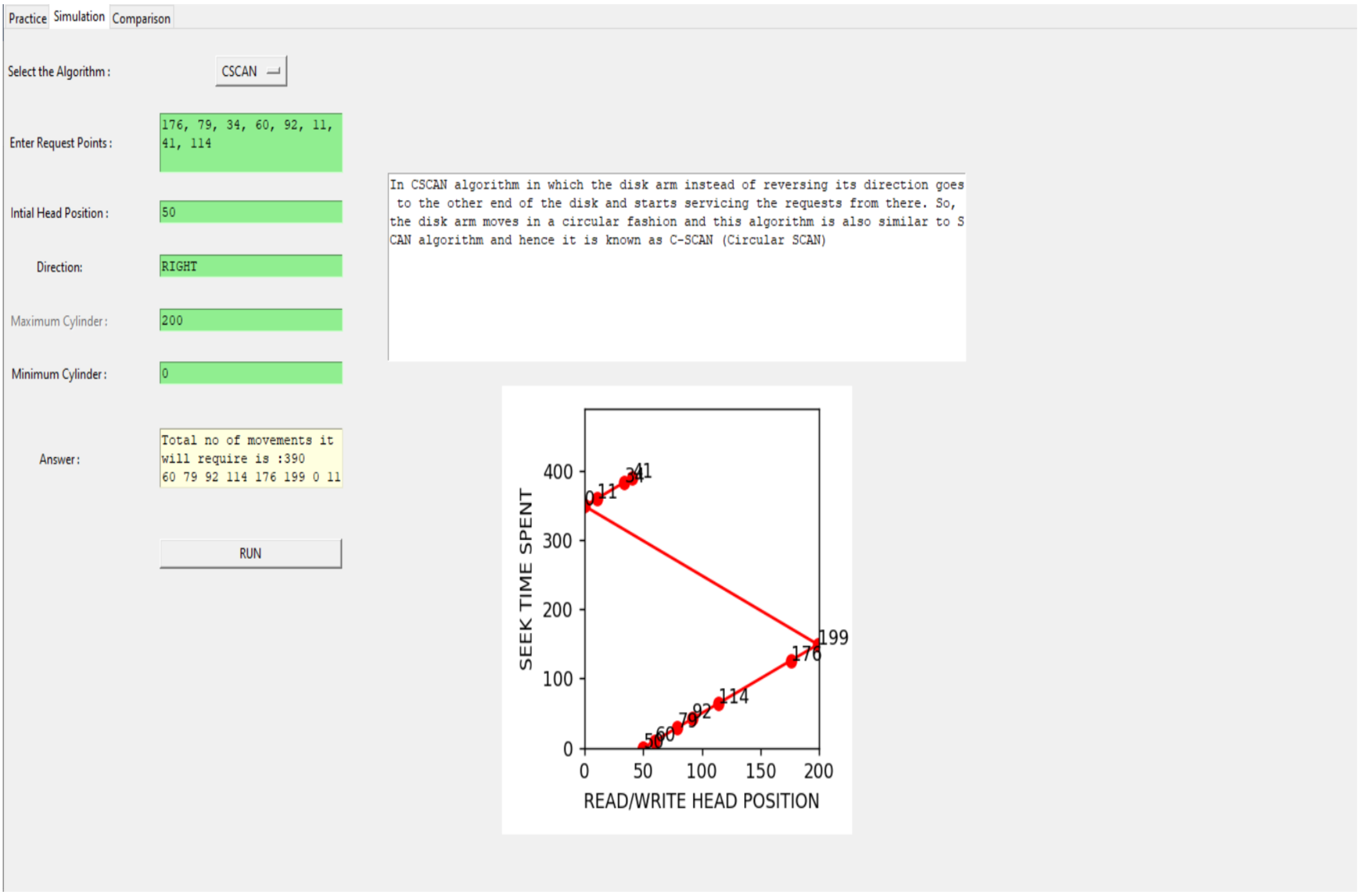
```
Total no of movements it
will require is :337
60 79 92 114 176 199 41 3
```

```
In SCAN algorithm the disk arm moves into a particular direction and services th
e requests coming in its path and after reaching the end of disk, it reverses it
s direction and again services the request arriving in its path. So, this algori
thm works as an elevator and hence also known as elevator algorithm. As a result
, the requests at the midrange are serviced more and those arriving behind the d
isk arm will have to wait.
```

5)C-SCAN :

Algorithm:

- Let Request array represents an array storing indexes of tracks that have been requested in ascending order of their time of arrival. ‘head’ is the position of disk head.
- The head services only in the right direction from 0 to size of the disk.
- While moving in the left direction do not service any of the tracks.
- When we reach at the beginning(left end) reverse the direction.
- While moving in right direction it services all tracks one by one.
- While moving in right direction calculate the absolute distance of the track from the head.
- Increment the total seek count with this distance.
- Currently serviced track position now becomes the new head position.
- Go to step 6 until we reach at right end of the disk.
- If we reach at the right end of the disk reverse the direction and go to step 3 until all tracks in request array have not been serviced.



Select the Algorithm : CSCAN

Enter Request Points : 176, 79, 34, 60, 92, 11, 41, 114

Intial Head Position : 50

Direction: RIGHT

Maximum Cylinder : 200

Minimum Cylinder : 0

Answer :

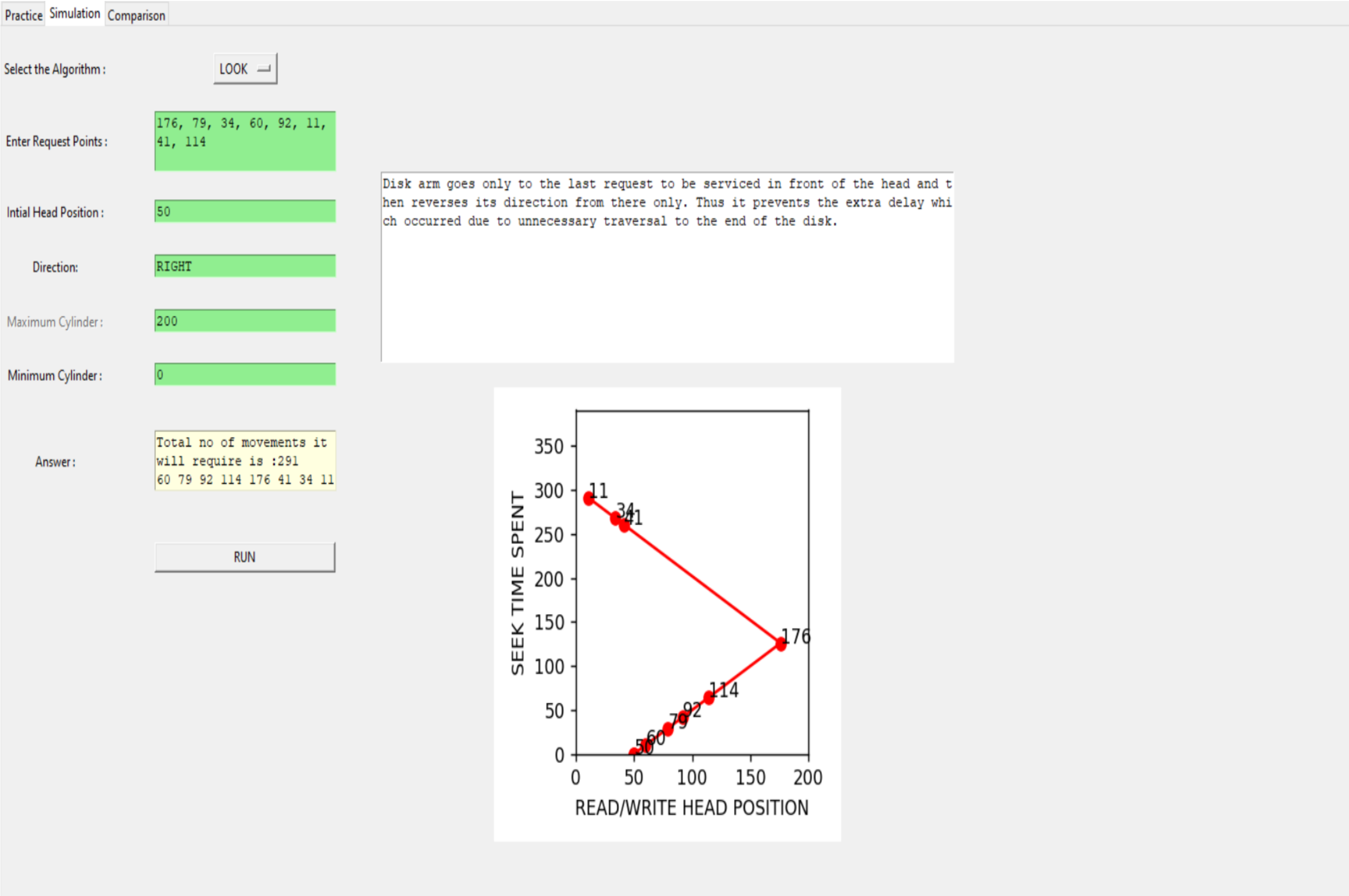
```
Total no of movements it
will require is :390
60 79 92 114 176 199 0 11
```

```
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 S
CAN algorithm and hence it is known as C-SCAN (Circular SCAN)
```

6) **LOOK** :

Algorithm:

- Let Request array represents an array storing indexes of tracks that have been requested in ascending order of their time of arrival. ‘head’ is the position of disk head.
- The initial direction in which head is moving is given and it services in the same direction.
- The head services all the requests one by one in the direction head is moving.
- The head continues to move in the same direction until all the request in this direction are not finished.
- While moving in this direction calculate the absolute distance of the track from the head.
- Increment the total seek count with this distance.
- Currently serviced track position now becomes the new head position.
- Go to step 5 until we reach at last request in this direction.
- If we reach where no requests are needed to be serviced in this direction reverse the direction and go to step 3 until all tracks in request array have not been serviced.



Select the Algorithm : LOOK

Enter Request Points : 176, 79, 34, 60, 92, 11, 41, 114

Intial Head Position : 50

Direction: RIGHT

Maximum Cylinder : 200

Minimum Cylinder : 0

Answer :

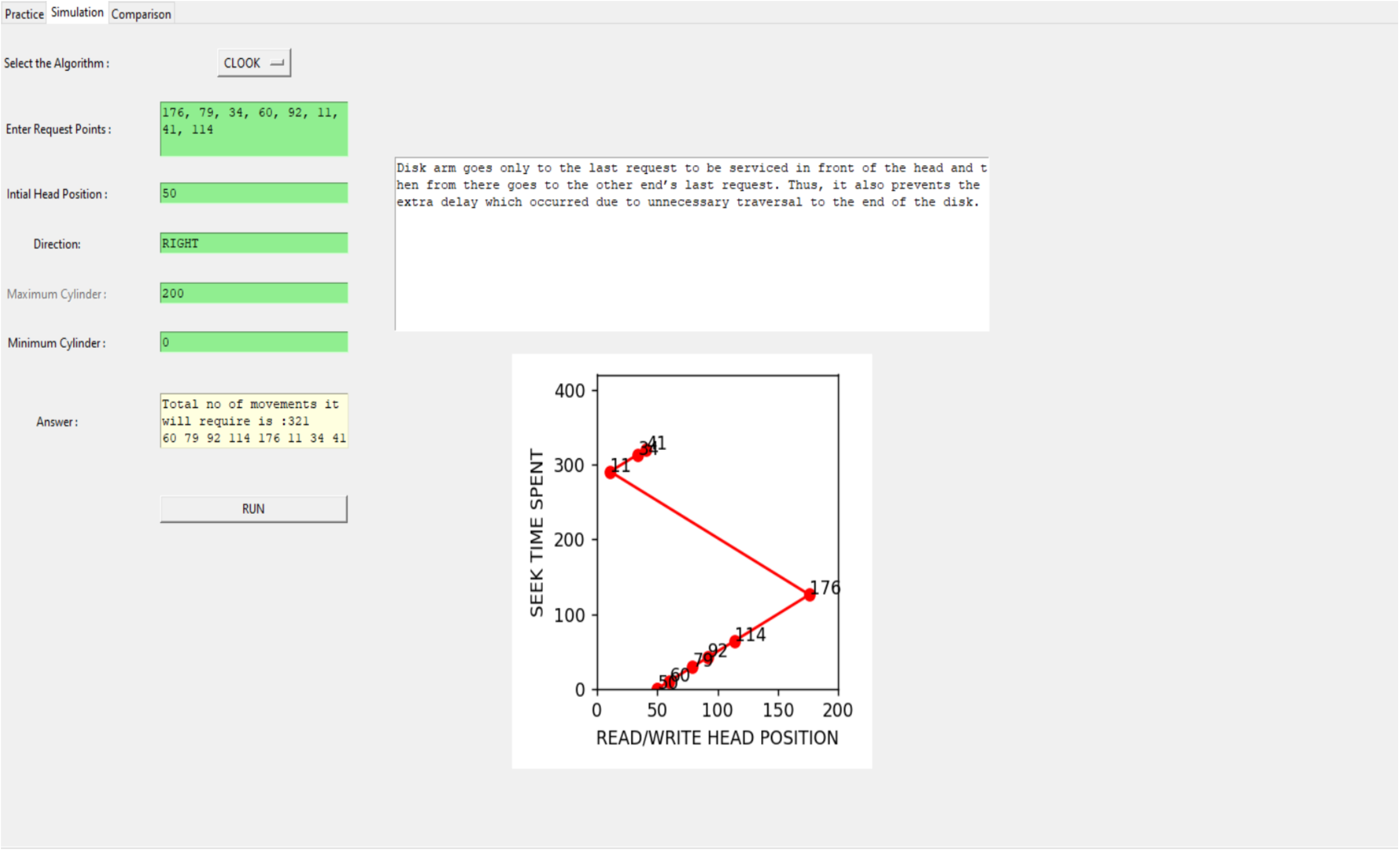
```
Total no of movements it
will require is :291
60 79 92 114 176 41 34 11
```

```
Disk arm goes only to the last request to be serviced in front of the head and t
hen reverses its direction from there only. Thus it prevents the extra delay whi
ch occurred due to unnecessary traversal to the end of the disk.
```

7)C-LOOK :

Algorithm:

- Let Request array represents an array storing indexes of the tracks that have been requested in ascending order of their time of arrival and **head** is the position of the disk head.
- The initial direction in which the head is moving is given and it services in the same direction.
- The head services all the requests one by one in the direction it is moving.
- The head continues to move in the same direction until all the requests in this direction have been serviced.
- While moving in this direction, calculate the absolute distance of the tracks from the head.
- Increment the total seek count with this distance.
- Currently serviced track position now becomes the new head position.
- Go to step 5 until we reach the last request in this direction.
- If we reach the last request in the current direction then reverse the direction and move the head in this direction until we reach the last request that is needed to be serviced in this direction without servicing the intermediate requests.
- Reverse the direction and go to step 3 until all the requests have not been serviced.



Select the Algorithm : CLOOK

Enter Request Points : 176, 79, 34, 60, 92, 11, 41, 114

Intial Head Position : 50

Direction: RIGHT

Maximum Cylinder : 200

Minimum Cylinder : 0

Answer :

```
Total no of movements it
will require is :321
60 79 92 114 176 11 34 41
```

```
Disk arm goes only to the last request to be serviced in front of the head and t
hen 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.
```