Date - 11/9/2020

Experiment No. 5

Disk scheduling algorithms

Aim: To write a program to implement the disk scheduling algorithms

a) FCFS
b) SCAN
c) C-SCAN

Theory: Diskschedulingisdonebyoperatingsystemstoschedulel/Orequestsarrivingfor 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. Thus other I/O requests need to wait in the waiting queue and need to be scheduled.
- Two or more requests may be far from each other so can result in greater disk arm movement.
- Hard drives are one of the slowest parts of the computer system and thus need to be accessed in an efficient manner.

1. FCFS:

It is the simplest form of disk scheduling. The requests are serviced in the order they arrive in the queue. This algorithm is very easy to implement. However, it doesn't reduce the average seek time effectively.

2. SCAN

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

3. C-SCAN

C-SCAN works just like the SCAN to some extent. It begins its scan toward the nearest end and works its way all the way to the end of the system. Once it hits the bottom or top it jumps to the other end and moves in the same direction.

Algorithm:

1. FCFS

```
Step 1: Start.
Step 2: Store the requests in an array named requests.
Step 3: Store the position of the head in the variable head.
Step 4: Add head to requests and set requests[0] = head, set i = 0, time=0
Step 5: Repeat until array end:
                seektime = abs(requests[i+1] - requests[i])
                time += seektime
Step 6:  avg_time = time/length of array requests
Step 7:  Stop the program.
```

2. SCAN

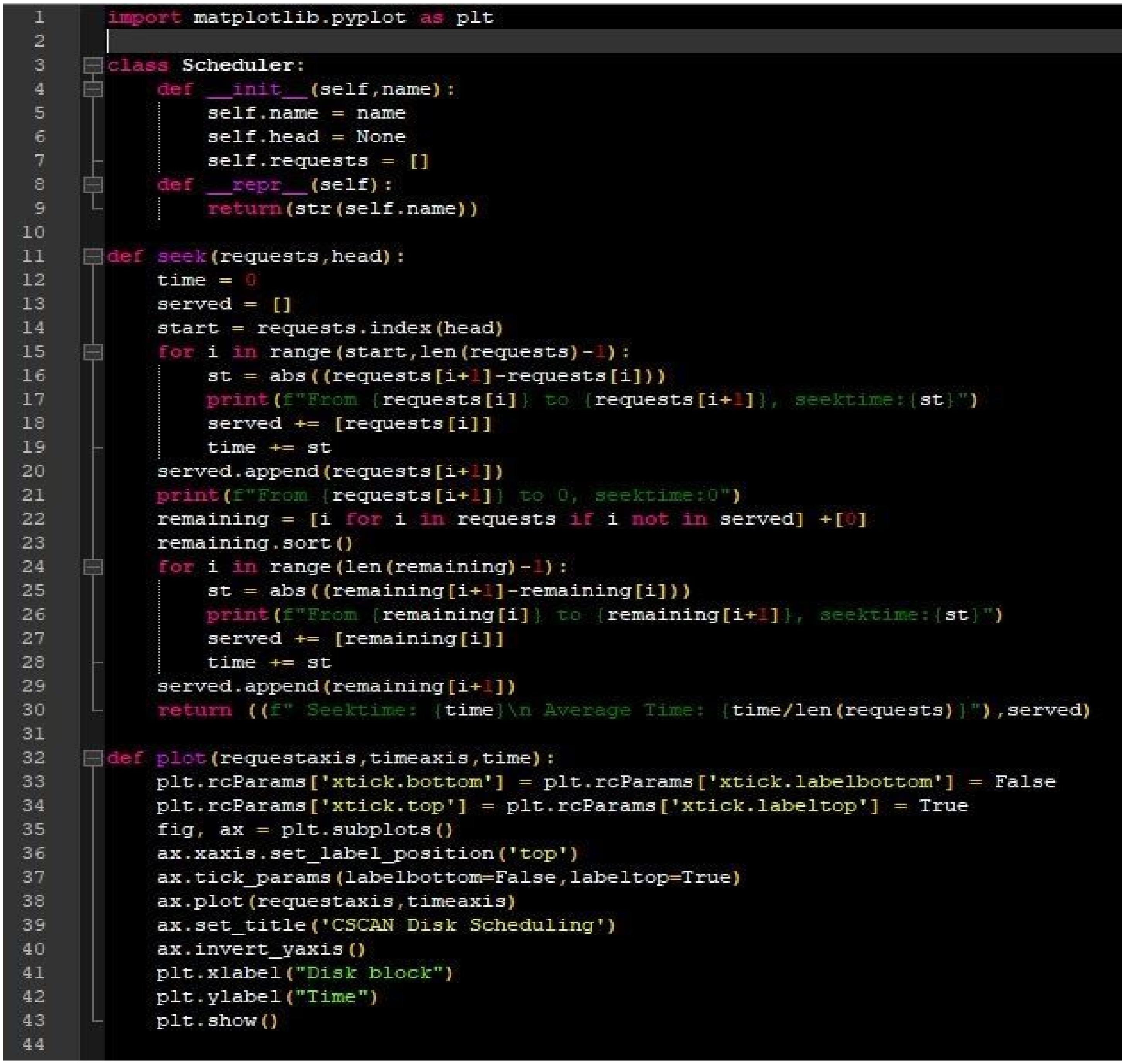
```
Step 1: Start.
Step 2: Store the requests in an array named requests.
Step 3: Store the position of the head in the variable head.
Step 4: Set served = new array(), requests.add(head), i=0
Step 5: Sort requests in ascending order.
Step 6:  Repeat until array end:
                seektime = abs(requests[i+1] - requests[i])
                time += seektime
                served.add(requests[i])
                i+=1
Step 7: Set remaining = new array(), remaining.sort(descending)
        Repeat until array end:
                i=0
                remaining.add(requests[i] if requests[i] not in served)
                i+=1
Step 8: Repeat until array end:
                i=0
                seektime = abs(remaining[i+1] - remaining[i])
                time += seektimeserved.add(remaining[i])
                i+=1
        served.add(remaining[i+1])
Step 9: avg_time = time/length of array served
Step 10: Stop the program.
```

3. C-SCAN

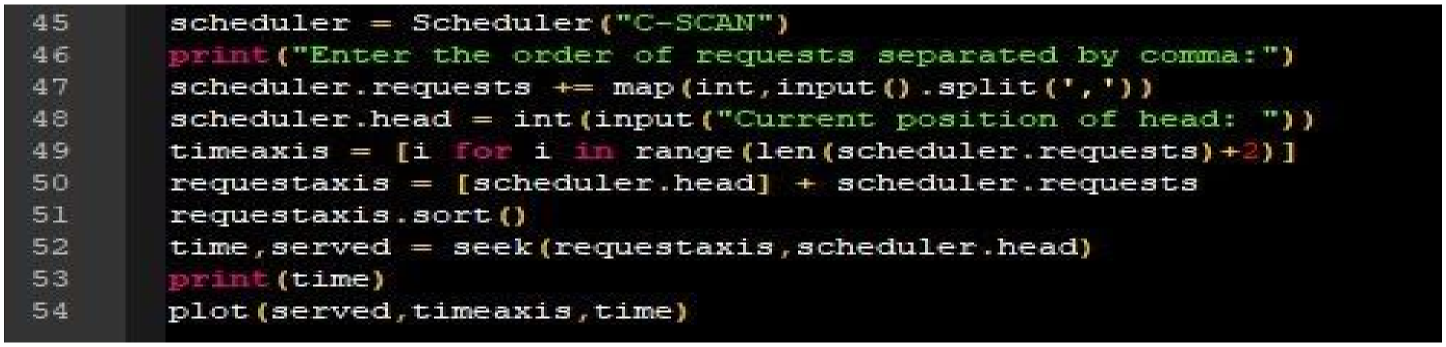
```
Step 1: Start.
Step 2: Store the requests in an array named requests
Step 3: Store the position of the head in the variable head
Step 4:  Set served = new array(), requests.add(head), i=0
Step 5:  Sort requests in ascending order.
Step 6:  Repeat until array end:
                seektime = abs(requests[i+1] - requests[i])
                time += seektime
                served.add(requests[i])
                i+=1
Step 7:  Set remaining = new array(), remaining.add(0)
        Repeat until array end:
                i=0
                remaining.add(requests[i] if requests[i] not in served)
                i+=1
Step 8:  Repeat until array end: i=0, remaining.sort()
                seektime = abs(remaining[i+1] - remaining[i])
                time += seektimeserved.add(remaining[i])
                i+=1
        served.add(remaining[i+1])
Step 9: avg_time = time/length of array served
Step 10: Stop the program.
```

Program:

1. FCFS

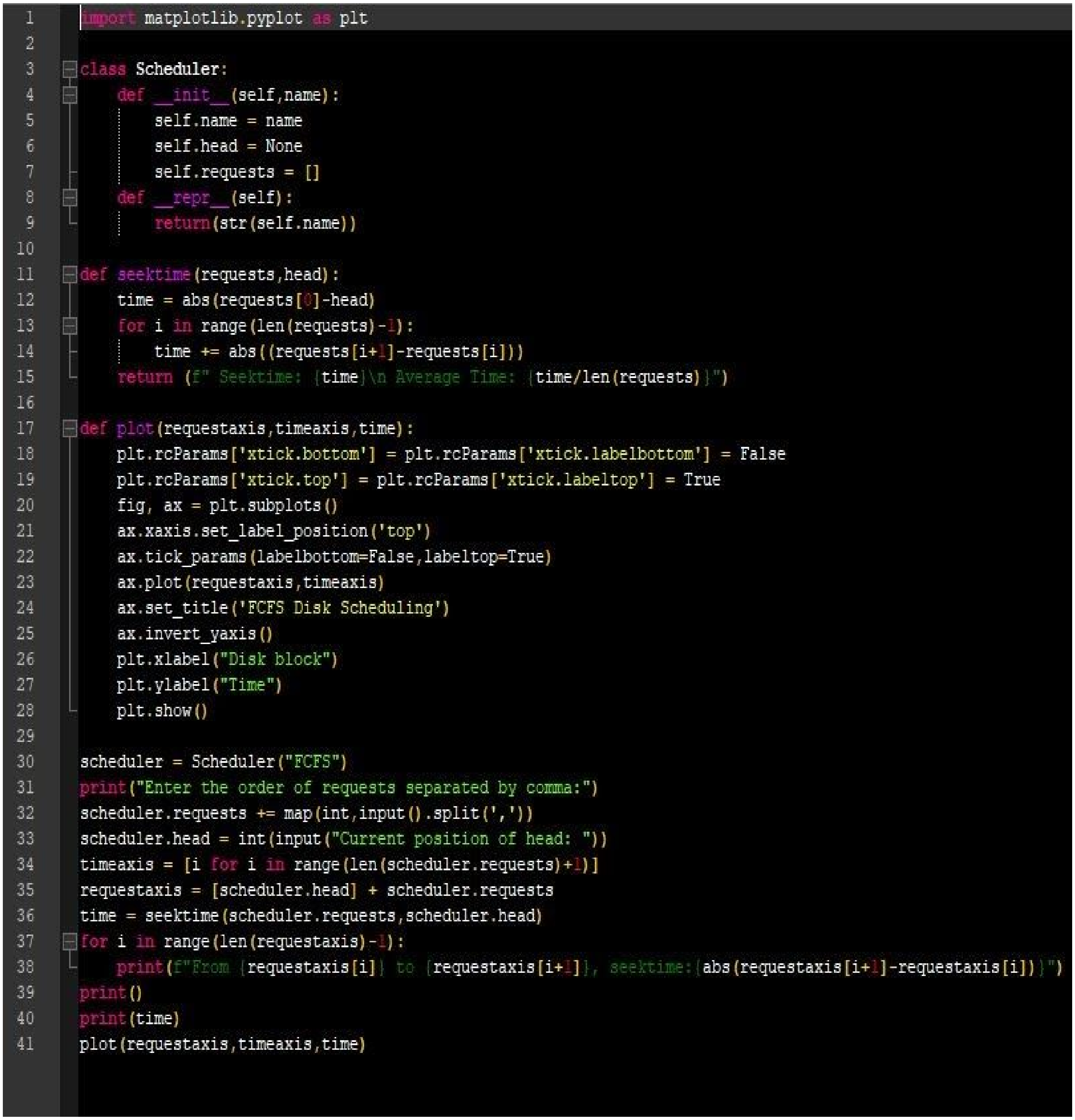


```python
import matplotlib.pyplot as plt

class Scheduler:
    def __init__(self,name):
        self.name = name
        self.head = None
        self.requests = []
    def __repr__(self):
        return(str(self.name))

def seek(requests,head):
    time = 0
    served = []
    start = requests.index(head)
    for i in range(start,len(requests)-1):
        st = abs((requests[i+1]-requests[i]))
        print(f"From {requests[i]} to {requests[i+1]}, seektime:{st}")
        served += [requests[i]]
        time += st
    served.append(requests[i+1])
    print(f"From {requests[i+1]} to 0, seektime:0")
    remaining = [i for i in requests if i not in served] +[0]
    remaining.sort()
    for i in range(len(remaining)-1):
        st = abs((remaining[i+1]-remaining[i]))
        print(f"From {remaining[i]} to {remaining[i+1]}, seektime:{st}")
        served += [remaining[i]]
        time += st
    served.append(remaining[i+1])
    return ((f" Seektime: {time}\n Average Time: {time/len(requests)}"),served)

def plot(requestaxis,timeaxis,time):
    plt.rcParams['xtick.bottom'] = plt.rcParams['xtick.labelbottom'] = False
    plt.rcParams['xtick.top'] = plt.rcParams['xtick.labeltop'] = True
    fig, ax = plt.subplots()
    ax.xaxis.set_label_position('top')
    ax.tick_params(labelbottom=False,labeltop=True)
    ax.plot(requestaxis,timeaxis)
    ax.set_title('CSCAN Disk Scheduling')
    ax.invert_yaxis()
    plt.xlabel("Disk block")
    plt.ylabel("Time")
    plt.show()

```



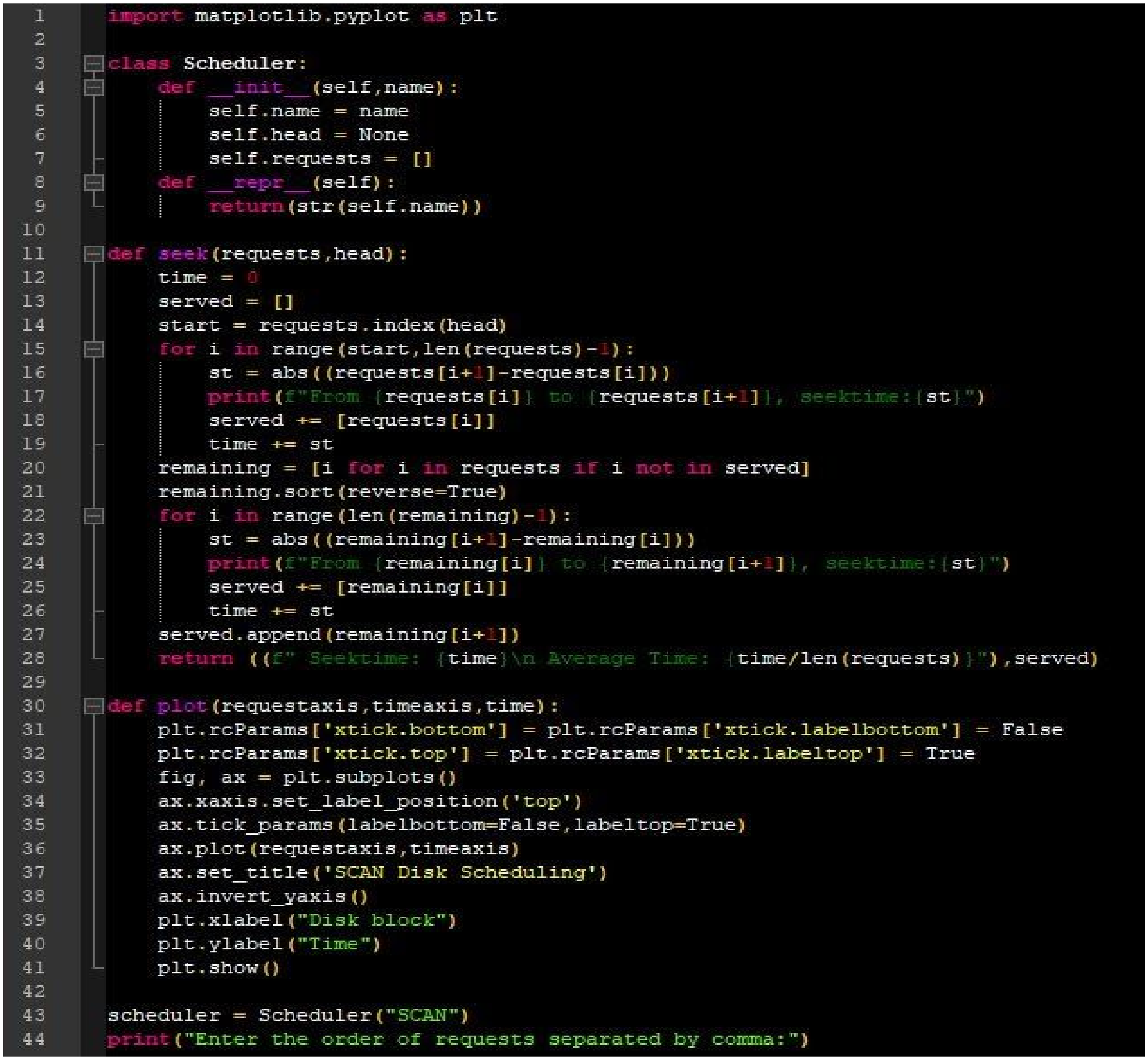
```python
scheduler = Scheduler("C-SCAN")
print("Enter the order of requests separated by comma:")
scheduler.requests += map(int,input().split(','))
scheduler.head = int(input("Current position of head: "))
timeaxis = [i for i in range(len(scheduler.requests)+2)]
requestaxis = [scheduler.head] + scheduler.requests
requestaxis.sort()
time,served = seek(requestaxis,scheduler.head)
print(time)
plot(served,timeaxis,time)
```

2. SCAN

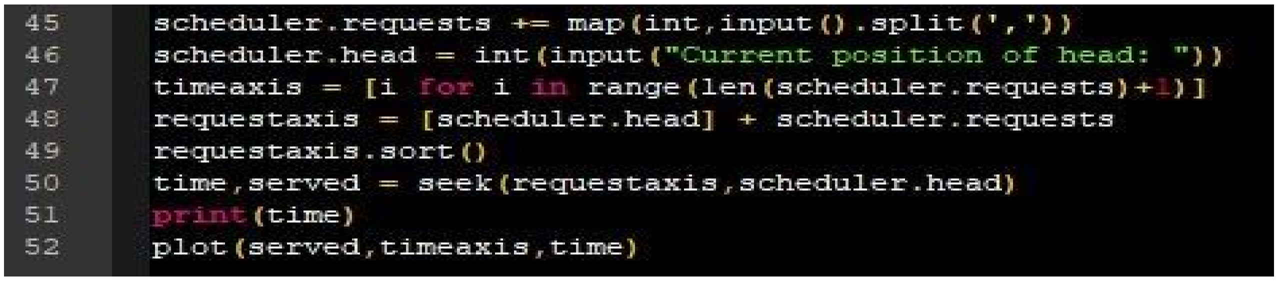


```python
import matplotlib.pyplot as plt

class Scheduler:
    def __init__(self,name):
        self.name = name
        self.head = None
        self.requests = []
    def __repr__(self):
        return(str(self.name))

def seektime(requests,head):
    time = abs(requests[0]-head)
    for i in range(len(requests)-1):
        time += abs((requests[i+1]-requests[i]))
    return (f" Seektime: {time}\n Average Time: {time/len(requests)}")

def plot(requestaxis,timeaxis,time):
    plt.rcParams['xtick.bottom'] = plt.rcParams['xtick.labelbottom'] = False
    plt.rcParams['xtick.top'] = plt.rcParams['xtick.labeltop'] = True
    fig, ax = plt.subplots()
    ax.xaxis.set_label_position('top')
    ax.tick_params(labelbottom=False,labeltop=True)
    ax.plot(requestaxis,timeaxis)
    ax.set_title('FCFS Disk Scheduling')
    ax.invert_yaxis()
    plt.xlabel("Disk block")
    plt.ylabel("Time")
    plt.show()

scheduler = Scheduler("FCFS")
print("Enter the order of requests separated by comma:")
scheduler.requests += map(int,input().split(','))
scheduler.head = int(input("Current position of head: "))
timeaxis = [i for i in range(len(scheduler.requests)+1)]
requestaxis = [scheduler.head] + scheduler.requests
time = seektime(scheduler.requests,scheduler.head)
for i in range(len(requestaxis)-1):
    print(f"From {requestaxis[i]} to {requestaxis[i+1]}, seektime:{abs(requestaxis[i+1]-requestaxis[i])}")
print()
print(time)
plot(requestaxis,timeaxis,time)
```

3. C-SCAN



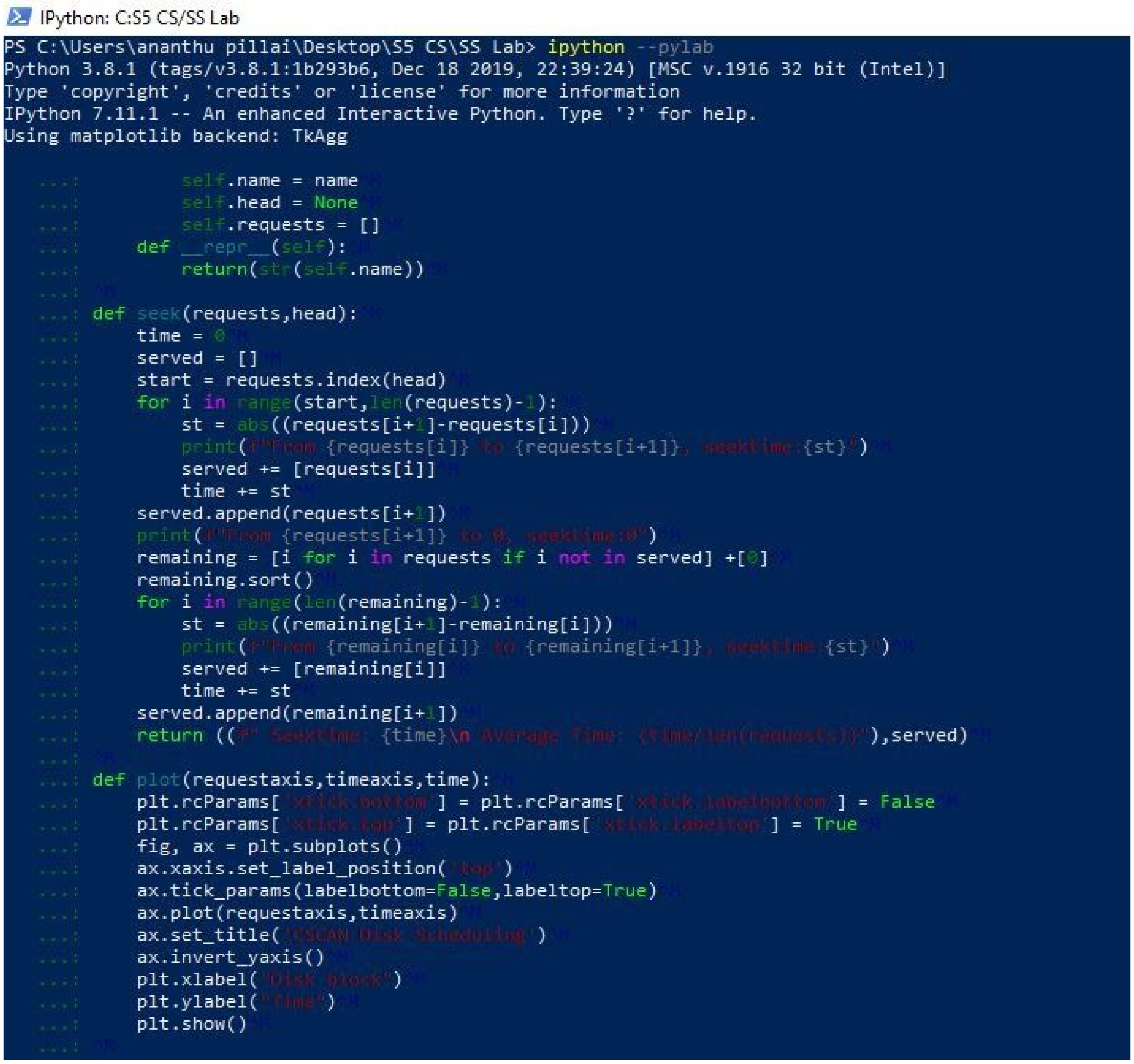
```python
import matplotlib.pyplot as plt

class Scheduler:
    def __init__(self,name):
        self.name = name
        self.head = None
        self.requests = []
    def __repr__(self):
        return(str(self.name))

def seek(requests,head):
    time = 0
    served = []
    start = requests.index(head)
    for i in range(start,len(requests)-1):
        st = abs((requests[i+1]-requests[i]))
        print(f"From {requests[i]} to {requests[i+1]}, seektime:{st}")
        served += [requests[i]]
        time += st
    remaining = [i for i in requests if i not in served]
    remaining.sort(reverse=True)
    for i in range(len(remaining)-1):
        st = abs((remaining[i+1]-remaining[i]))
        print(f"From {remaining[i]} to {remaining[i+1]}, seektime:{st}")
        served += [remaining[i]]
        time += st
    served.append(remaining[i+1])
    return ((f" Seektime: {time}\n Average Time: {time/len(requests)}"),served)

def plot(requestaxis,timeaxis,time):
    plt.rcParams['xtick.bottom'] = plt.rcParams['xtick.labelbottom'] = False
    plt.rcParams['xtick.top'] = plt.rcParams['xtick.labeltop'] = True
    fig, ax = plt.subplots()
    ax.xaxis.set_label_position('top')
    ax.tick_params(labelbottom=False,labeltop=True)
    ax.plot(requestaxis,timeaxis)
    ax.set_title('SCAN Disk Scheduling')
    ax.invert_yaxis()
    plt.xlabel("Disk block")
    plt.ylabel("Time")
    plt.show()

scheduler = Scheduler("SCAN")
print("Enter the order of requests separated by comma:")
```



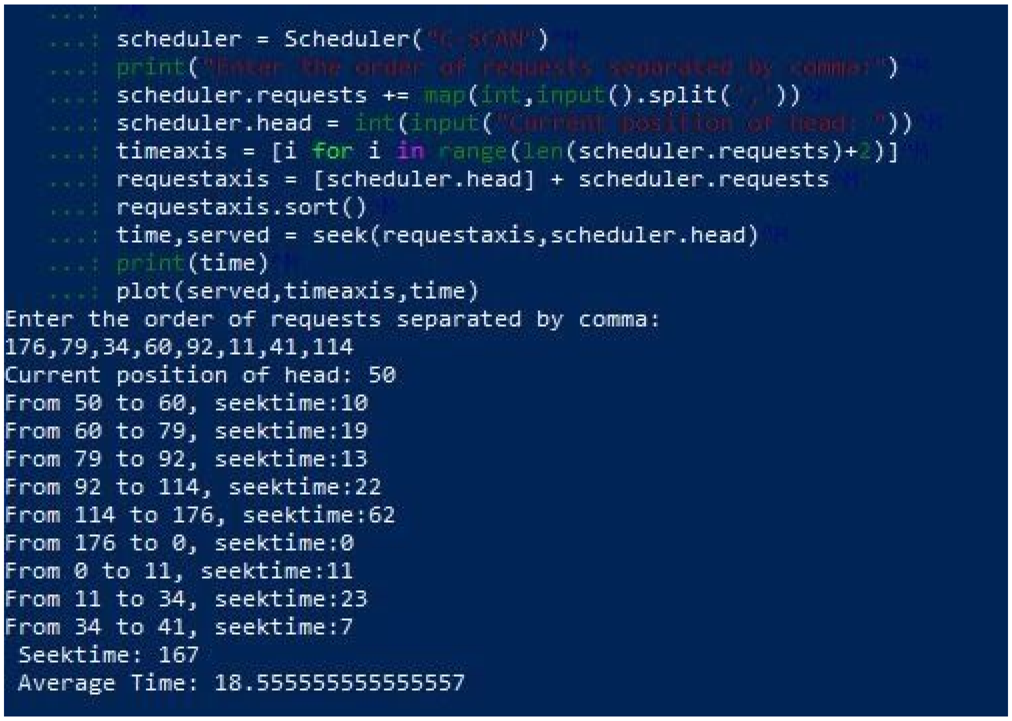
```python
scheduler.requests += map(int,input().split(','))
scheduler.head = int(input("Current position of head: "))
timeaxis = [i for i in range(len(scheduler.requests)+1)]
requestaxis = [scheduler.head] + scheduler.requests
requestaxis.sort()
time,served = seek(requestaxis,scheduler.head)
print(time)
plot(served,timeaxis,time)
```

Output:

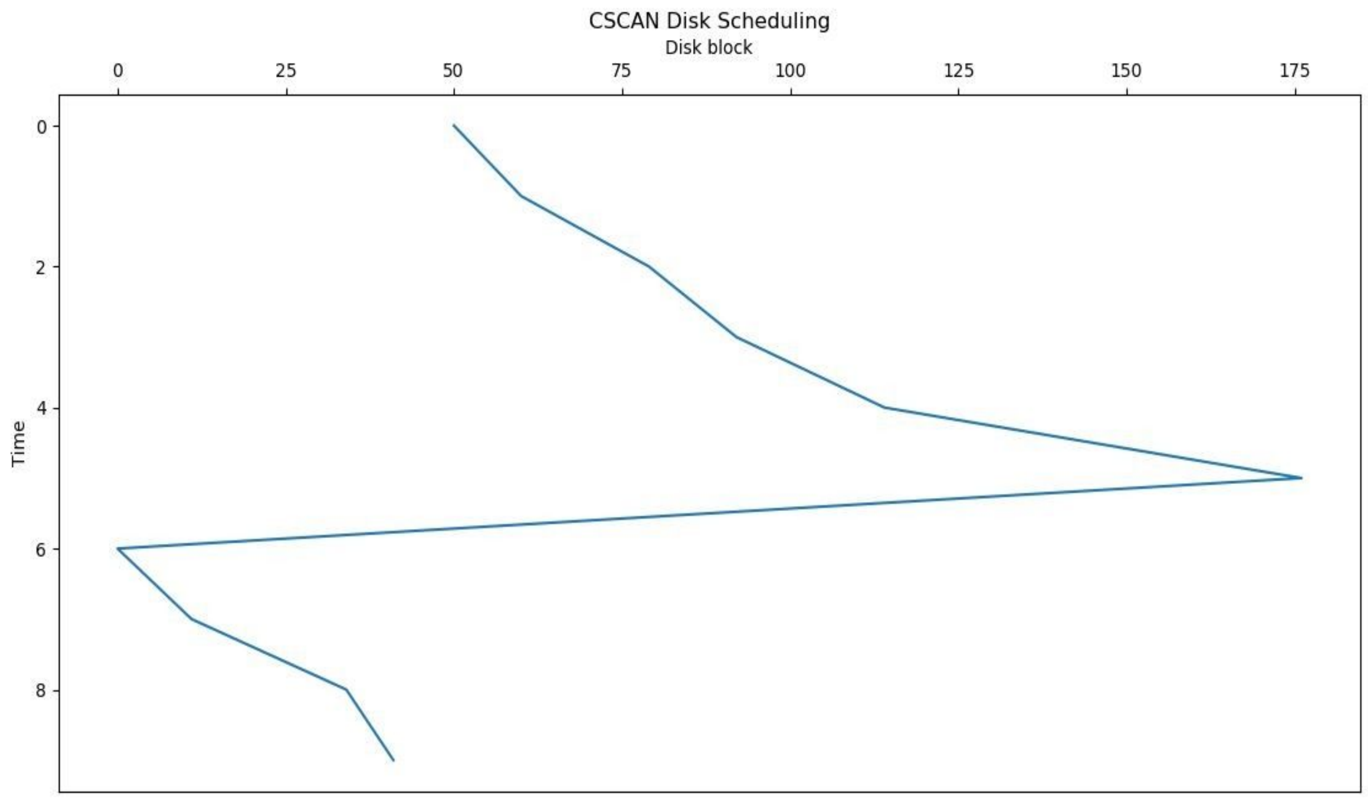
CSCAN



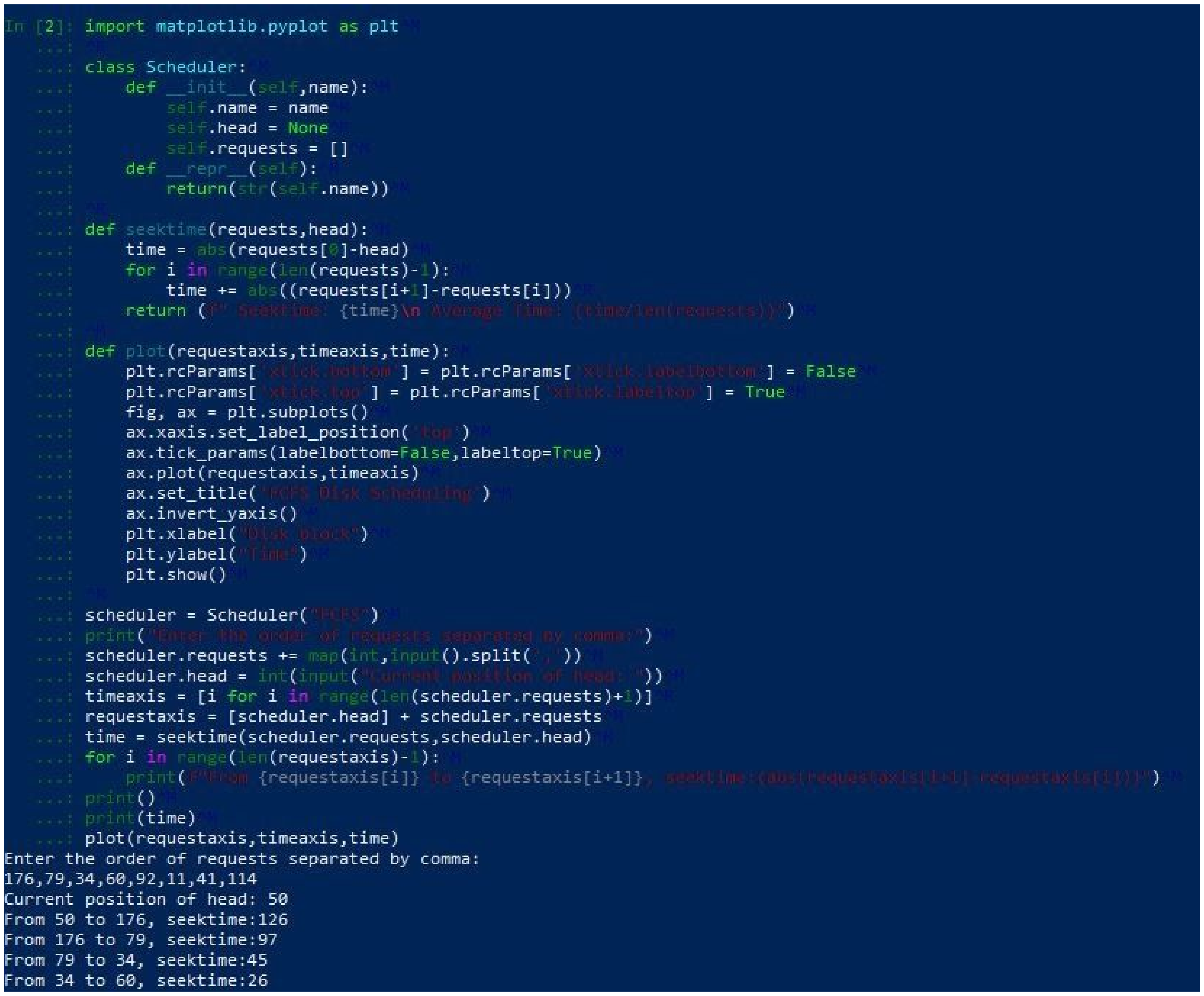
```
PS C:\Users\ananthu pillai\Desktop\S5 CS\SS Lab> ipython --pylab
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 22:39:24) [MSC v.1916 32 bit (Intel)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.11.1 -- An enhanced Interactive Python. Type '?' for help.
Using matplotlib backend: TkAgg

   ...:         self.name = name^M
   ...:         self.head = None^M
   ...:         self.requests = []^M
   ...:     def __repr__(self):^M
   ...:         return(str(self.name))^M
   ...: ^M
   ...: def seek(requests,head):^M
   ...:     time = 0^M
   ...:     served = []^M
   ...:     start = requests.index(head)^M
   ...:     for i in range(start,len(requests)-1):^M
   ...:         st = abs((requests[i+1]-requests[i]))^M
   ...:         print(f"From {requests[i]} to {requests[i+1]}, seektime:{st}")^M
   ...:         served += [requests[i]]^M
   ...:         time += st^M
   ...:     served.append(requests[i+1])^M
   ...:     print(f"From {requests[i+1]} to 0, seektime:0")^M
   ...:     remaining = [i for i in requests if i not in served] +[0]^M
   ...:     remaining.sort()^M
   ...:     for i in range(len(remaining)-1):^M
   ...:         st = abs((remaining[i+1]-remaining[i]))^M
   ...:         print(f"From {remaining[i]} to {remaining[i+1]}, seektime:{st}")^M
   ...:         served += [remaining[i]]^M
   ...:         time += st^M
   ...:     served.append(remaining[i+1])^M
   ...:     return ((f" Seektime: {time}\n Average Time: {time/len(requests)}"),served)^M
   ...: ^M
   ...: def plot(requestaxis,timeaxis,time):^M
   ...:     plt.rcParams['xtick.bottom'] = plt.rcParams['xtick.labelbottom'] = False^M
   ...:     plt.rcParams['xtick.top'] = plt.rcParams['xtick.labeltop'] = True^M
   ...:     fig, ax = plt.subplots()^M
   ...:     ax.xaxis.set_label_position('top')^M
   ...:     ax.tick_params(labelbottom=False,labeltop=True)^M
   ...:     ax.plot(requestaxis,timeaxis)^M
   ...:     ax.set_title('CSCAN Disk Scheduling')^M
   ...:     ax.invert_yaxis()^M
   ...:     plt.xlabel('Disk block')^M
   ...:     plt.ylabel('Time')^M
   ...:     plt.show()^M
   ...: ^M
```



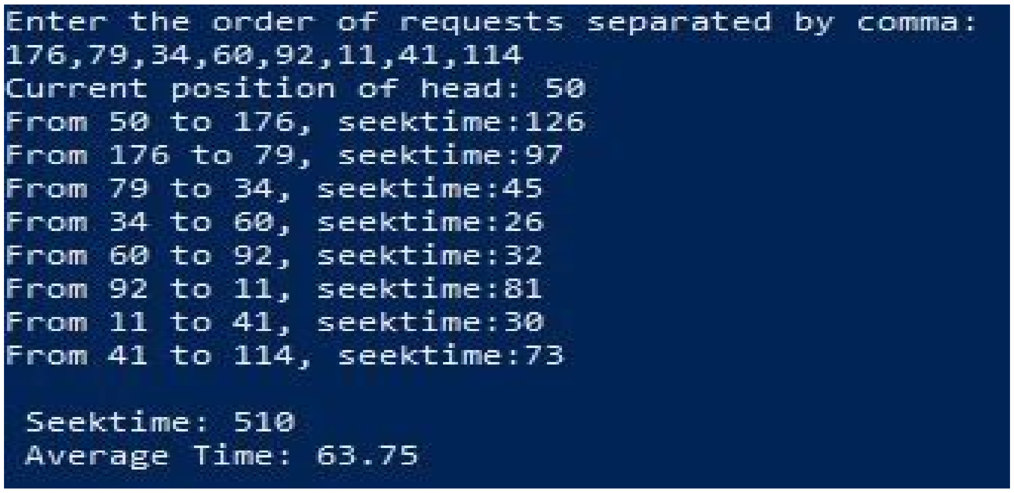
```
   ...:
   ...: scheduler = Scheduler("C-SCAN")
   ...: print("Enter the order of requests separated by comma:")
   ...: scheduler.requests += map(int,input().split(','))
   ...: scheduler.head = int(input("Current position of head: "))
   ...: timeaxis = [i for i in range(len(scheduler.requests)+2)]
   ...: requestaxis = [scheduler.head] + scheduler.requests
   ...: requestaxis.sort()
   ...: time,served = seek(requestaxis,scheduler.head)
   ...: print(time)
   ...: plot(served,timeaxis,time)
Enter the order of requests separated by comma:
176,79,34,60,92,11,41,114
Current position of head: 50
From 50 to 60, seektime:10
From 60 to 79, seektime:19
From 79 to 92, seektime:13
From 92 to 114, seektime:22
From 114 to 176, seektime:62
From 176 to 0, seektime:0
From 0 to 11, seektime:11
From 11 to 34, seektime:23
From 34 to 41, seektime:7
 Seektime: 167
 Average Time: 18.555555555555557
```



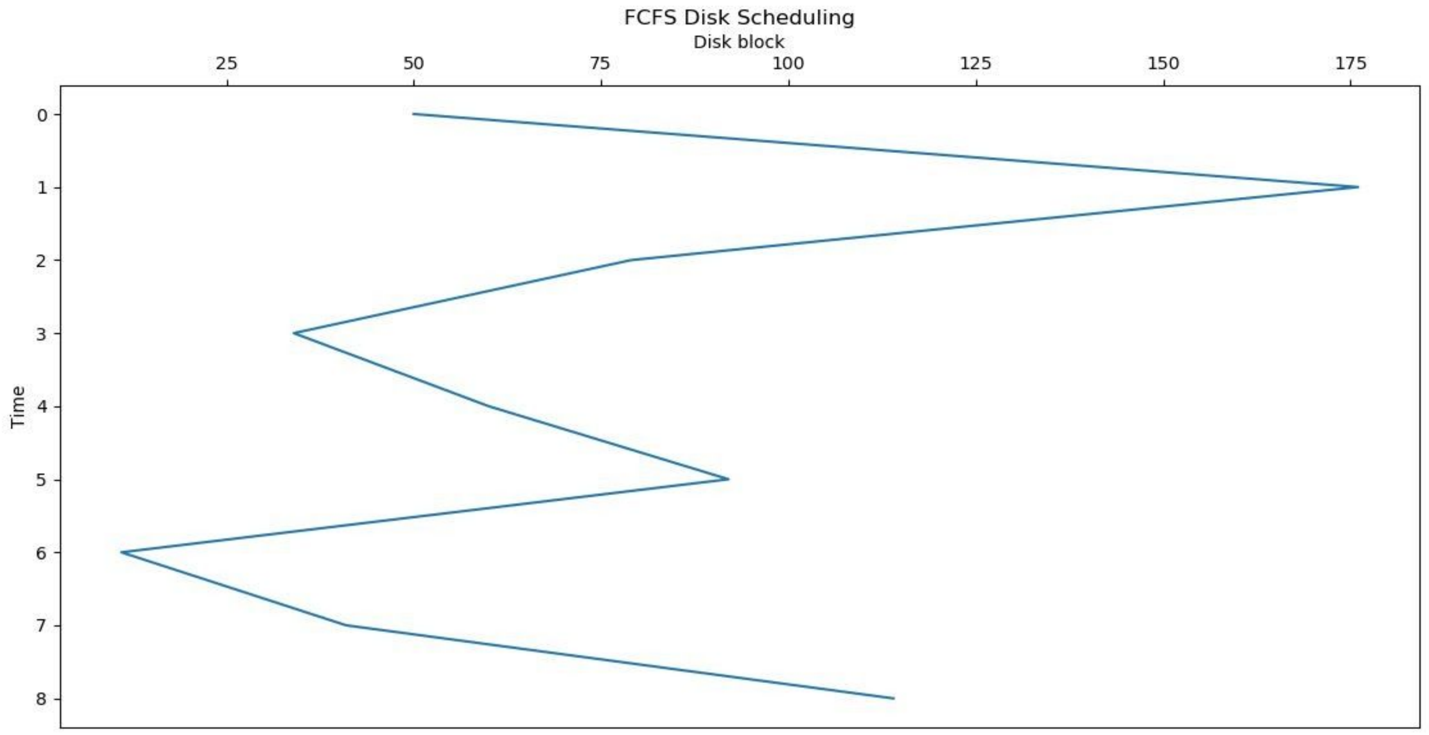
FCFS



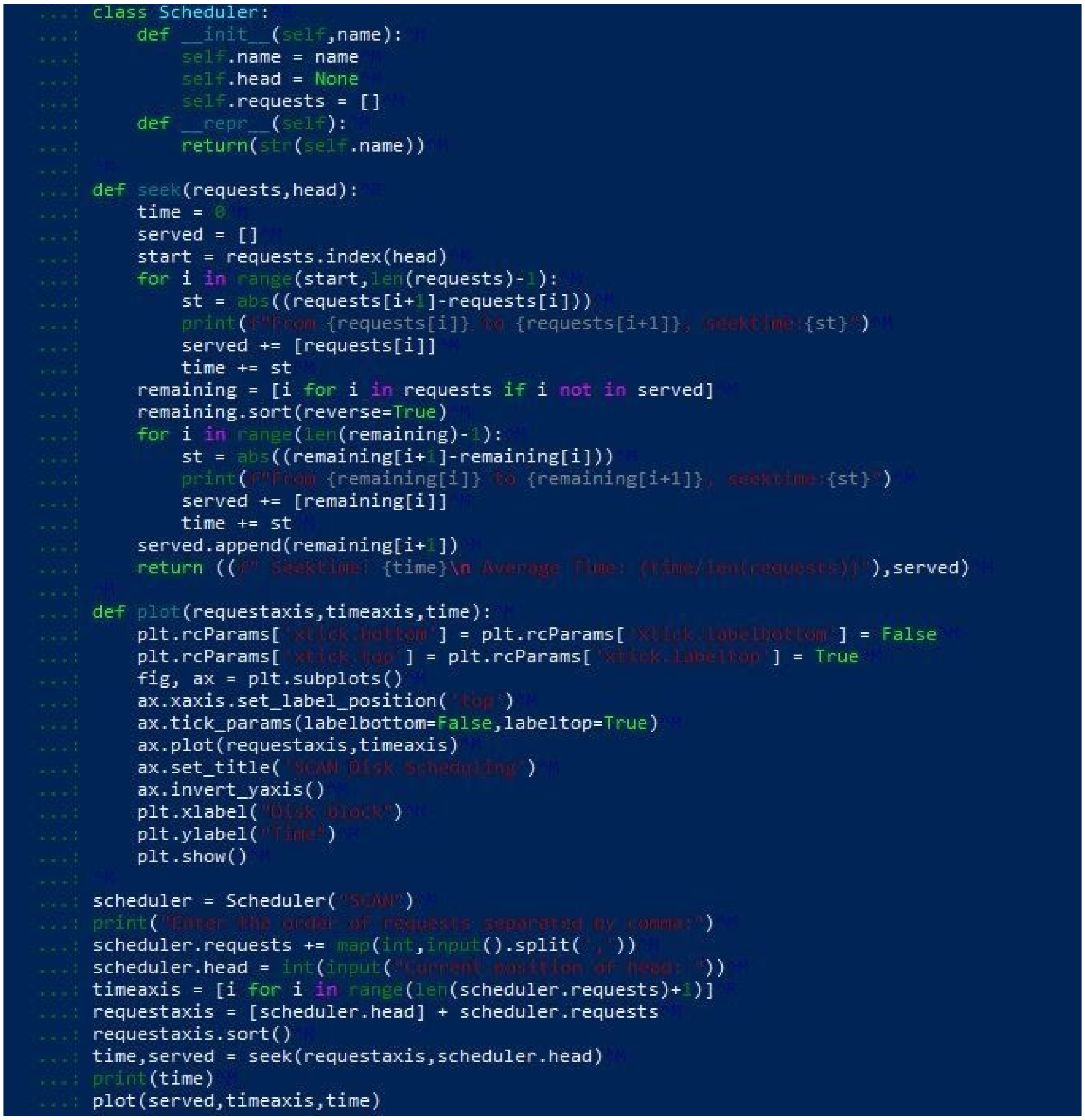
```
In [2]: import matplotlib.pyplot as plt
   ...: 
   ...: class Scheduler:
   ...:     def __init__(self,name):
   ...:         self.name = name
   ...:         self.head = None
   ...:         self.requests = []
   ...:     def __repr__(self):
   ...:         return(str(self.name))
   ...: 
   ...: def seektime(requests,head):
   ...:     time = abs(requests[0]-head)
   ...:     for i in range(len(requests)-1):
   ...:         time += abs((requests[i+1]-requests[i]))
   ...:     return (f" Seektime: {time}\n Average Time: {time/len(requests)}")
   ...: 
   ...: def plot(requestaxis,timeaxis,time):
   ...:     plt.rcParams['xtick.bottom'] = plt.rcParams['xtick.labelbottom'] = False
   ...:     plt.rcParams['xtick.top'] = plt.rcParams['xtick.labeltop'] = True
   ...:     fig, ax = plt.subplots()
   ...:     ax.xaxis.set_label_position('top')
   ...:     ax.tick_params(labelbottom=False,labeltop=True)
   ...:     ax.plot(requestaxis,timeaxis)
   ...:     ax.set_title('FCFS Disk Scheduling')
   ...:     ax.invert_yaxis()
   ...:     plt.xlabel("Disk block")
   ...:     plt.ylabel("Time")
   ...:     plt.show()
   ...: 
   ...: scheduler = Scheduler("FCFS")
   ...: print("Enter the order of requests separated by comma:")
   ...: scheduler.requests += map(int,input().split(','))
   ...: scheduler.head = int(input("Current position of head: "))
   ...: timeaxis = [i for i in range(len(scheduler.requests)+1)]
   ...: requestaxis = [scheduler.head] + scheduler.requests
   ...: time = seektime(scheduler.requests,scheduler.head)
   ...: for i in range(len(requestaxis)-1):
   ...:     print(f"From {requestaxis[i]} to {requestaxis[i+1]}, seektime:{abs(requestaxis[i+1]-requestaxis[i])}")
   ...: print()
   ...: print(time)
   ...: plot(requestaxis,timeaxis,time)
Enter the order of requests separated by comma:
176,79,34,60,92,11,41,114
Current position of head: 50
From 50 to 176, seektime:126
From 176 to 79, seektime:97
From 79 to 34, seektime:45
From 34 to 60, seektime:26
```



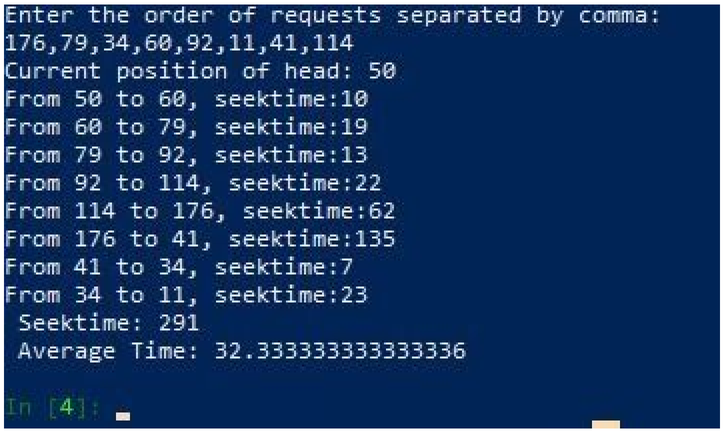
```
Enter the order of requests separated by comma:
176,79,34,60,92,11,41,114
Current position of head: 50
From 50 to 176, seektime:126
From 176 to 79, seektime:97
From 79 to 34, seektime:45
From 34 to 60, seektime:26
From 60 to 92, seektime:32
From 92 to 11, seektime:81
From 11 to 41, seektime:30
From 41 to 114, seektime:73

 Seektime: 510
 Average Time: 63.75
```


FCFS Disk Scheduling
Disk block
25
50
75
100
125
150
175
Time
0
1
2
3
4
5
6
7
8

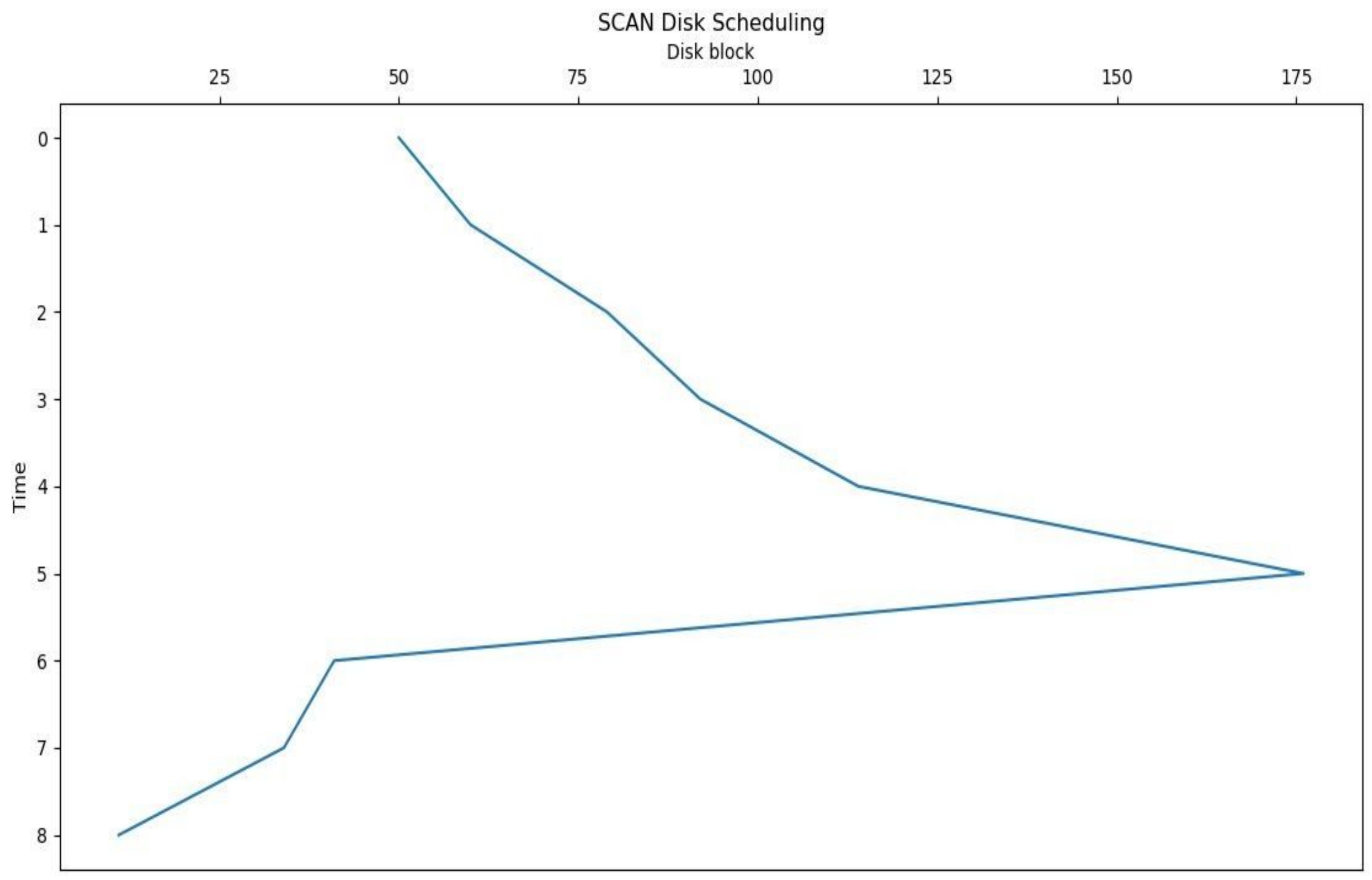
SCAN



```
   ...: class Scheduler:
   ...:     def __init__(self,name):
   ...:         self.name = name
   ...:         self.head = None
   ...:         self.requests = []
   ...:     def __repr__(self):
   ...:         return(str(self.name))
   ...:
   ...: def seek(requests,head):
   ...:     time = 0
   ...:     served = []
   ...:     start = requests.index(head)
   ...:     for i in range(start,len(requests)-1):
   ...:         st = abs((requests[i+1]-requests[i]))
   ...:         print(f"From {requests[i]} to {requests[i+1]}, seektime:{st}")
   ...:         served += [requests[i]]
   ...:         time += st
   ...:     remaining = [i for i in requests if i not in served]
   ...:     remaining.sort(reverse=True)
   ...:     for i in range(len(remaining)-1):
   ...:         st = abs((remaining[i+1]-remaining[i]))
   ...:         print(f"From {remaining[i]} to {remaining[i+1]}, seektime:{st}")
   ...:         served += [remaining[i]]
   ...:         time += st
   ...:     served.append(remaining[i+1])
   ...:     return ((f" Seektime: {time}\n Average Time: {time/len(requests)}"),served)
   ...:
   ...: def plot(requestaxis,timeaxis,time):
   ...:     plt.rcParams['xtick.bottom'] = plt.rcParams['xtick.labelbottom'] = False
   ...:     plt.rcParams['xtick.top'] = plt.rcParams['xtick.labeltop'] = True
   ...:     fig, ax = plt.subplots()
   ...:     ax.xaxis.set_label_position('top')
   ...:     ax.tick_params(labelbottom=False,labeltop=True)
   ...:     ax.plot(requestaxis,timeaxis)
   ...:     ax.set_title('SCAN Disk Scheduling')
   ...:     ax.invert_yaxis()
   ...:     plt.xlabel("Disk Block")
   ...:     plt.ylabel("Time")
   ...:     plt.show()
   ...:
   ...: scheduler = Scheduler("SCAN")
   ...: print("Enter the order of requests separated by comma:")
   ...: scheduler.requests += map(int,input().split(','))
   ...: scheduler.head = int(input("Current position of head: "))
   ...: timeaxis = [i for i in range(len(scheduler.requests)+1)]
   ...: requestaxis = [scheduler.head] + scheduler.requests
   ...: requestaxis.sort()
   ...: time,served = seek(requestaxis,scheduler.head)
   ...: print(time)
   ...: plot(served,timeaxis,time)
```



```
Enter the order of requests separated by comma:
176,79,34,60,92,11,41,114
Current position of head: 50
From 50 to 60, seektime:10
From 60 to 79, seektime:19
From 79 to 92, seektime:13
From 92 to 114, seektime:22
From 114 to 176, seektime:62
From 176 to 41, seektime:135
From 41 to 34, seektime:7
From 34 to 11, seektime:23
 Seektime: 291
 Average Time: 32.333333333333336

In [4]:
```



Result:

The program to simulate the disk scheduling algorithms such as FCFS, SCAN and C-SCAN have been implemented and simulated successfully.