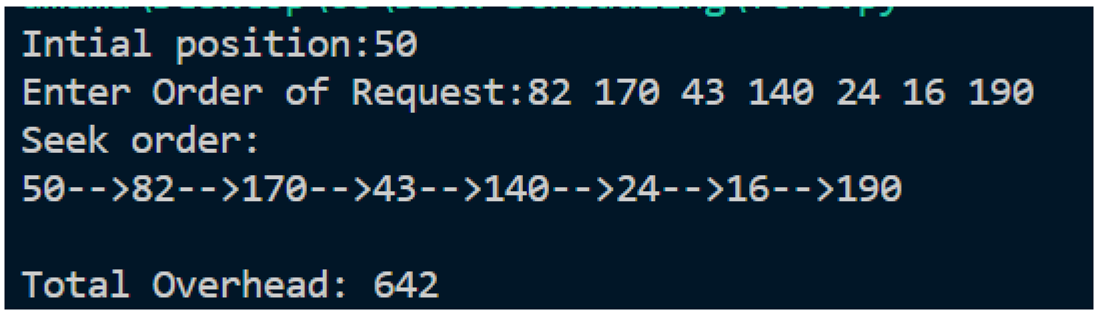
Disk Scheduling

First Come First Serve

Code:

```
intial = int(input("Intial position:"))
order = list(map(int,input("Enter Order of Request:").split()))

#fcfs
final=[intial]

prev=intial
overhead=0

while len(order):
    x=order[0]
    order.pop(0)

    overhead+=abs(x-prev)
    prev=x

    final.append(x)

print("Seek order:")
print("-->".join([str(i) for i in final]))
print("Total Overhead: ",end="")
print(overhead)
```

Output:



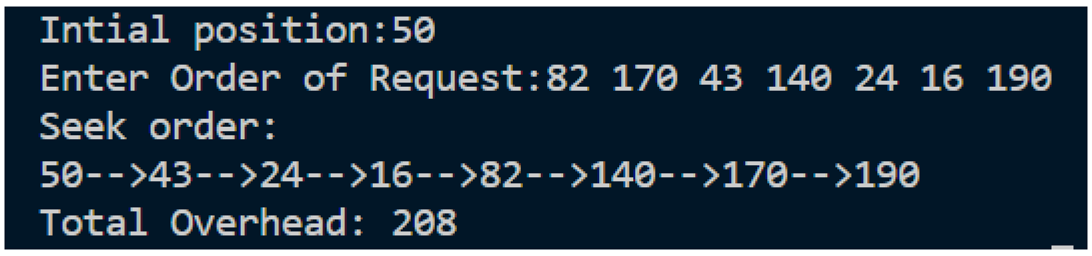
```
Intial position:50
Enter Order of Request:82 170 43 140 24 16 190
Seek order:
50-->82-->170-->43-->140-->24-->16-->190

Total Overhead: 642
```

Shortest Seek Time First

Code:

```
def near_sort(a,k):

    fine=[k]
    prev=k

    while len(a):
        fine.append(min(a,key=lambda x:abs(x-prev)))
        prev=fine[-1]
        a.remove(prev)

    return fine


intial = int(input("Intial position:"))
order = list(map(int,input("Enter Order of Request:").split()))


order=near_sort(order,intial)
final=[]

prev=intial
overhead=0

while len(order):
    x=order[0]
    order.pop(0)

    overhead+=abs(x-prev)
    prev=x
    final.append(x)


print("Seek order:")
print("-->".join([str(i) for i in final]))
print("Total Overhead: ",end="")
print(overhead)
```

Output:



```
Intial position:50
Enter Order of Request:82 170 43 140 24 16 190
Seek order:
50-->43-->24-->16-->82-->140-->170-->190
Total Overhead: 208
```

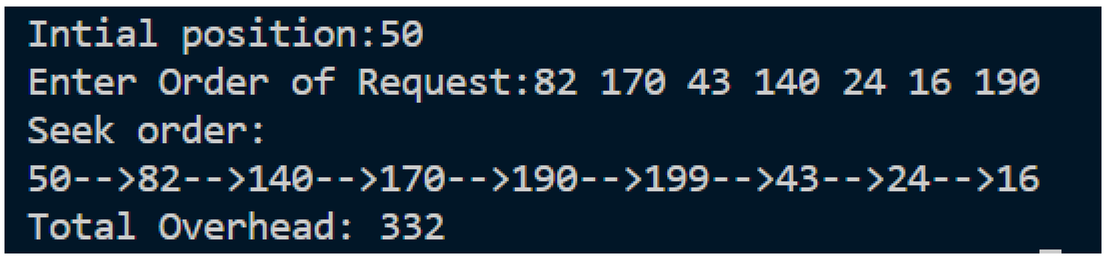
SCAN

Code:

```
def scan_sort(a,k,end):

    first = sorted([i for i in a if i>k])
    middle = [end if end not in first else None]
    last = sorted([i for i in a if i<k],key=lambda x:-x)

    return first+middle+last


intial = int(input("Intial position:"))
order = list(map(int,input("Enter Order of Request:").split()))
```

```
start=0
end=199

order=scan_sort(order,intial,end)
final=[intial]

prev=intial
overhead=0

while len(order):
    x=order[0]
    order.pop(0)

    overhead+=abs(x-prev)
    prev=x
    final.append(x)


print("Seek order:")
print("-->".join([str(i) for i in final]))
print("Total Overhead: ",end="")
print(overhead)
```

Output:



```
Intial position:50
Enter Order of Request:82 170 43 140 24 16 190
Seek order:
50-->82-->140-->170-->190-->199-->43-->24-->16
Total Overhead: 332
```

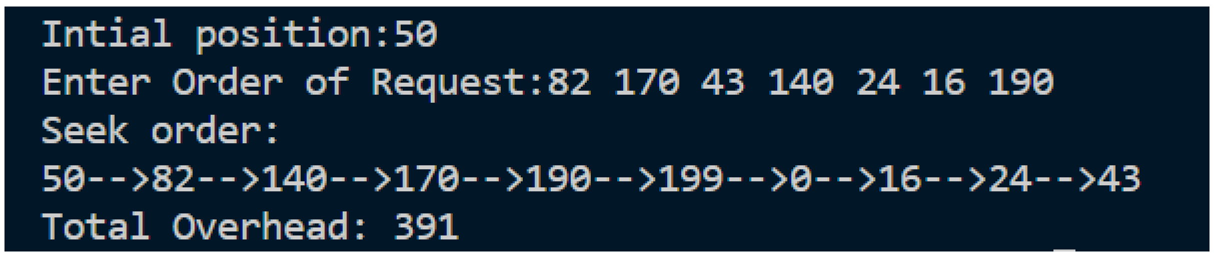
C-SCAN

Code:

```
def scan_sort(a,k,end,start):

    first = sorted([i for i in a if i>k])
    last = sorted([i for i in a if i<k])
    middle = [end if end not in first else None]+[start if start not in last else None]

    return first+middle+last


intial = int(input("Intial position:"))
order = list(map(int,input("Enter Order of Request:").split()))

start=0
end=199

order=scan_sort(order,intial,end,start)
final=[intial]
```

```
prev=intial
overhead=0

while len(order):
    x=order[0]
    order.pop(0)

    overhead+=abs(x-prev)
    prev=x
    final.append(x)


print("Seek order:")
print("-->".join([str(i) for i in final]))
print("Total Overhead: ",end="")
print(overhead)
```

Output:



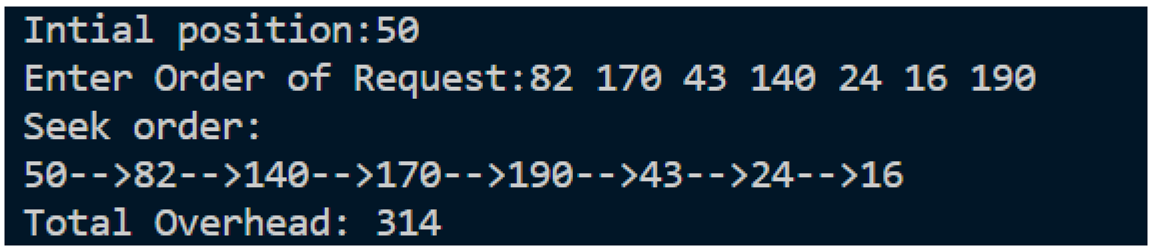
```
Intial position:50
Enter Order of Request:82 170 43 140 24 16 190
Seek order:
50-->82-->140-->170-->190-->199-->0-->16-->24-->43
Total Overhead: 391
```

LOOK

Code:

```
def look_sort(a,k):

    first = sorted([i for i in a if i>k])
    last = sorted([i for i in a if i<k],key=lambda x:-x)

    return first+last


intial = int(input("Intial position:"))
order = list(map(int,input("Enter Order of Request:").split()))

start=0
end=199

order=look_sort(order,intial)
final=[intial]

prev=intial
overhead=0

while len(order):
    x=order[0]
    order.pop(0)
    overhead+=abs(x-prev)
    prev=x
    final.append(x)


print("Seek order:")
print("-->".join([str(i) for i in final]))
print("Total Overhead: ",end="")
print(overhead)
```

Output:



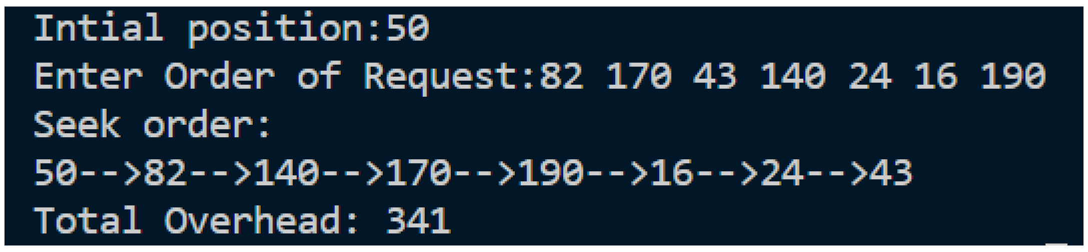
```
Intial position:50
Enter Order of Request:82 170 43 140 24 16 190
Seek order:
50-->82-->140-->170-->190-->43-->24-->16
Total Overhead: 314
```

CLOOK

Code:

```
def look_sort(a,k):

    first = sorted([i for i in a if i>k])
    last = sorted([i for i in a if i<k])

    return first+last


intial = int(input("Intial position:"))
order = list(map(int,input("Enter Order of Request:").split()))

start=0
end=199

order=look_sort(order,intial)
final=[intial]

prev=intial
overhead=0

while len(order):
    x=order[0]
    order.pop(0)

    overhead+=abs(x-prev)
    prev=x
    final.append(x)


print("Seek order:")
print("-->".join([str(i) for i in final]))
print("Total Overhead: ",end="")
print(overhead)
```

Output :



```
Intial position:50
Enter Order of Request:82 170 43 140 24 16 190
Seek order:
50-->82-->140-->170-->190-->16-->24-->43
Total Overhead: 341
```