# GOA COLLEGE OF ENGINEERING

"Bhausaheb Bandodkar Technical Education Complex"

**Experiment No: 6**

**Aim:** To implement Shortest Seek Time First Algorithm

**Theory:**

Shortest seek time first (SSTF) algorithm selects the disk I/O request which requires the least disk arm movement from its current position regardless of the direction. It reduces the total seek time as compared to FCFS. It allows the head to move to the closest track in the service queue. Advantages of Shortest Seek Time First (SSTF) –

1. Better performance than FCFS scheduling algorithm.
2. It provides better throughput.
3. ThisalgorithmisusedinBatchProcessingsystemwherethroughputismoreimportant. 4. It has less average response and waiting time.

Disadvantages of Shortest Seek Time First (SSTF) –

1. Starvation is possible for some requests as it favours easy to reach request and ignores the far away

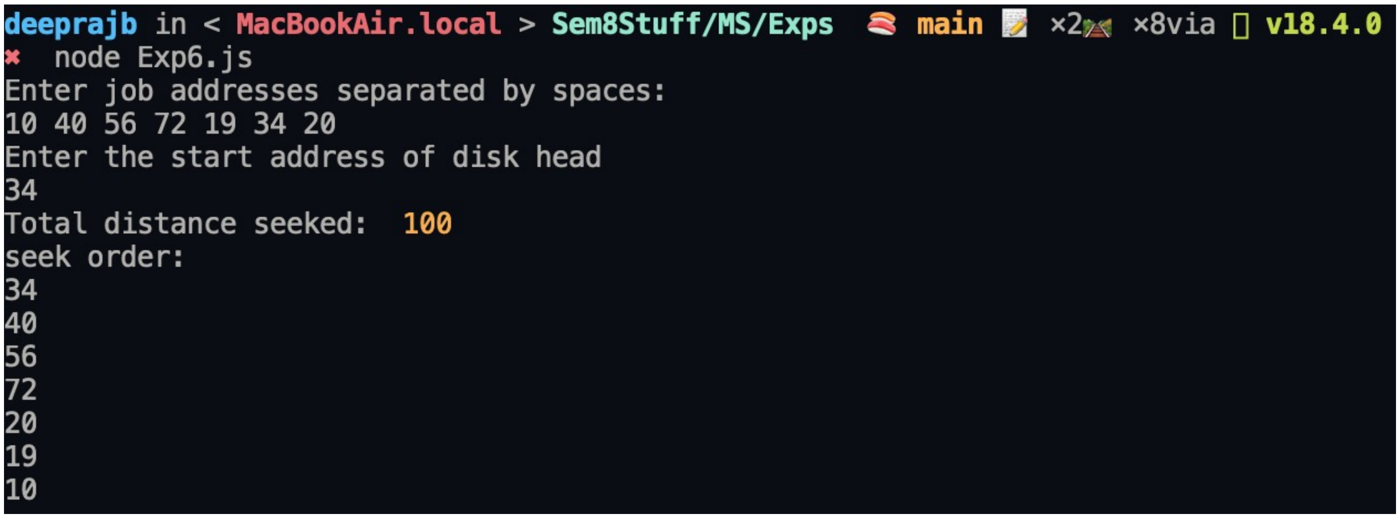
**Code:**

```
const readline = require("readline").createInterface({
input: process.stdin,
output: process.stdout
})
function sstf(start,jobs){
let totalDistance=0
let jobOrder=[]
let currPos=start
while(jobs.length>0){
let closestIdx=-1
let smallestDistance=Infinity
for(let i=0;i<jobs.length;i++){
let currDist=Math.abs(currPos-jobs[i])
if(currDist<smallestDistance){
smallestDistance=currDist
closestIdx=i
} }
jobOrder.push(jobs[closestIdx])
totalDistance+=Math.abs(currPos-jobs[closestIdx])
currPos=jobs[closestIdx]
jobs.splice(closestIdx,1)
}
console.log("Total distance seeked: ",totalDistance)
console.log("seek order:")
for(let i=0;i<jobOrder.length;i++){
```

```
console.log(jobOrder[i])
} }
async function getJobs(){
let joblist= await new Promise(resolve=>{
readline.question("Enter job addresses separated by spaces:\n",
resolve)
})
return joblist.split(" ")
}
async function getStart(){
let start=await new Promise(resolve=>{
readline.question("Enter the start address of disk head\n",
resolve)
})
return Number(start)}

const main = async() => {
let jobs=await getJobs()
let start=await getStart()
sstf(start,jobs)
process.exit(0)
}
main()
```

**Output:**



```
deeprajb in < MacBookAir.local > Sem8Stuff/MS/Exps  🍣 main 📝 ×2 ×8via  v18.4.0
✖  node Exp6.js
Enter job addresses separated by spaces:
10 40 56 72 19 34 20
Enter the start address of disk head
34
Total distance seeked:  100
seek order:
34
40
56
72
20
19
10
```

**Conclusion:** A program implementing Shortest Seek Time First Algorithm was successfully implemented and executed.