Name:- Pratiksingh Rajeshsingh Thakur
Registration No.: 2019BCS133
Roll No.: A 63
Division : A
Subject : Advanced Algorithm

Practical 9

Aim:- Implement the job scheduling algorithm.

Theory:-
Job scheduling algorithm is ~~and~~ one of the technique available in Greedy approach to solve the problem of scheduling a list of jobs in an order such that maxim-um number of jobs can be performed with a constraint that only one job can be executed at any time.

The problem can be solved by using the Greedy approach. In general the greedy strategy sort the list of job in an order ~~sta~~ such that the jobs with earlier start time are organised earlier. And then two tasks are compared on the basis of end time of ith job and start time of jth job such that i < j. If the start time of jth job is more than or equal to the end time of ith job then the jobs ith and jth job can be executed consecutively.

Steps involved in the algorithm:

1) Sort the list of job such that jobs with earlier start time are positioned before the jobs starting later

2) traverse the list of jobs from the starting end such that the start time of the current job should be more than or equal to the end time of the last job in the required sequence.

3) Once the loop is executed display the list.

pseudo code for the algorithm

```
job_list = [...., [start_i, end_i], ...]

sort (job_list)
/* sorting in the order as mentioned in step 1 */

list = [ ]
```

```
list.append([start_0, end_0])
t=0 , i=0
for t++ < job_list.size() do
    if job_list[t][0] >= list[i][1]:
        list.append([start_t, end_t])
        i++
end for
for j in list
    print(j)
end for
```

Proof of time complexity:

The step no. 2 and step no. 3 in the algorithm requires at most 'n' no. of iteration here n is the no. of elements in the array.

∴ time complexity of step no. 2 & 3 = O(n)

Step no. 1 requires n log (n) time to sort the list

hence
$T(n) = O(n \log n) + O(n) + O(n)$

$\therefore T(n) = O(n \log n)$
hence the overall time complexity of the algorithm is $O(n \log n)$

Conclusion:- The Greedy approach for the job scheduling algorithm is studied, and the time etc complexity proof is also studied