

CS7200: Algorithm Design and Analysis

Assignment-2

Sai Vijay Kumar Surineela

U01096171

Surineela.2@wright.edu

Manoj Kumar Reddy Avula

U01067535

avula.25@wright.edu

Prabhu Charan Murari

U01099304

murari.4@wright.edu

Pseudo code: -

PsuedoCode

The input file will have # Of jobs, JobId, startTime, endTime.

The output file should have # of jobs possible, jobs possible on M1 and jobs possible on M2 on 2 Machines with least overlapping.

All the Jobs are sorted on earliest End time using sort() which has a time complexity of $O(n\log(n))$, relies on Timesort sorting technique.

""""

Pusedo code starts...

Sort all the jobs based on end time in ascending manner.

Initate a empty list for Jobs on Machine1.

Initate a empty list for Jobs on Machine2.

Initialise a int variable to zero which stores the last end time of the job that run on Machine1, machine1LastEndTime.

Initialise a int variable to zero which stores the last end time of the job that run on Machine2, machine2LastEndTime.

for indiviual job in sorted Jobs:

 store JobID, JobStartTime, JobEndTime from indiviual job.

 if a new job start time is equal or greater than machine1 LastEndTime:

 update the machine1 LastEndTime to new job end time.

 add the new jobs ID to List of jobs that can run Machine1.

 else:

 update the machine2 LastEndTime to new job end time.

 add the new jobs ID to List of jobs that can run Machine2.

totalJobs is the toatl # of jobs that can run on 2 machine is summation of elements in Machine1 & Machine2.

Join all jobs in machine 1 with a space gap, which is to be printed in output file.

Join all jobs in machine 1 with a space gap, which is to be printed in output file.

""""

Greedy Heuristics used is Earliest Finish Time Maximizing the number of non-overlapping jobs on two machines, with a time complexity of $O(n \log(n))$.

"""

Example to Illustrate Correctness:

Consider the following list of jobs:

5

1 5 7

2 0 3

3 6 8

4 3 6

5 1 4

Execution:

1. Sort the Jobs by End Time:

- Sorted Job Ids: 2, 5, 4, 1, 3.
- Job 2 is assigned to M1.
- Job 5 and Job 2 overlaps, so it is assigned to M2.
- Job 4, after Job 2 ends, assigned to M1.
- Job 1, after Job 5 ends, assigned to M2.
- Job 3, after Job 4 ends, assigned to M1.

Result:

- 5
- M1: Jobs [2, 4, 3]
- M2: Jobs [1, 5]

"""**Note: same Pseudo code is provided in the code itself.**

Time complexity: The time complexity of the code is $O(n \log(n))$.