

<sup>9</sup> IT081 - Pankhania Anand R.

<sup>10</sup> DAA - Job Scheduling With Deadlines

<sup>11</sup> Problem Analysis :-

<sup>12</sup> In this problem, there is a list of jobs given. In the list deadlines & profits are also given for each job. And we're assuming that every job will take a single unit of time, then so the minimum deadline for a job is 1. If only one job ~~is~~ can be scheduled at a time, then maximize the profit.

Approach:-

To solve this problem, all subset of the set of jobs are generated to check whether the individual subset is feasible or not. Also, we have to keep track of on maximum profit for all feasible subset that has generated.

04 DAY 308.057  
SUNDAY

J/P:- list of jobs, no. of jobs present in list. their deadlines & profits.

10

O/P:- Job sequence, how jobs are taken & profit would be max. in it.

11

ex.

|           |    |     |    |    |    |  |
|-----------|----|-----|----|----|----|--|
| Jobs      | A  | B   | C  | D  | E  | $\left. \begin{array}{l} 0-1 \\ 1-2 \\ 2-3 \end{array} \right\} \text{slots possible}$ |
| Profits   | 19 | 100 | 27 | 25 | 15 |  |
| Deadlines | 1  | 2   | 2  | 1  | 3  |  |

1

| <sup>2</sup> Job consider | slot assign     | sol'n   | Profit    |
|---------------------------|-----------------|---------|-----------|
| 3 —                       | —               | $\phi$  | 0         |
| A                         | [0-1]           | A       | 19        |
| 4 B                       | [0-1][1-2]      | A, B    | 19+100    |
| C x(2)                    | [0-1][1-2]      | A, B    | 19+100    |
| 5 D x(1)                  | [0-1][1-2]      | A, B    | 19+100    |
| E                         | [0-1][1-2][2-3] | A, B, E | 19+100+15 |

6

134

### Algorithm:-

- 1) Begin
- 2) Sort the jobs in joblist according to their profit create a list of segn and slot to track free time slots
- 3) Initially make all slots  $\rightarrow$  free
- 4) for all given jobs  $i$  do
- 5)     for all jobs in list from ending of list  $j$  do
- 5)         if slot  $[j]$  is free then
- 6)             jobsequence  $[j] := i$
- 7)             make ~~at~~ slot  $[j] :=$  fill
- 8)             break the loop
- 9)     done
- 10) done
- 11) for all slots when it is not free do
- 12)     print id of job using joblist[jobseqn  $[i]$ ]
- 13) done
- 14) End

## Complexity :-

Time complexity :-  $O(n^2)$   $\left[ \because \underbrace{O(n \log n)}_{\text{(high)}} + O(n^2) \right]$

→ sort job acc. to decreasing order of deadline  $\Downarrow$   
 $= O(n \log n)$

→ for each job find slot in array size of  $n = O(n^2)$