Job Sequencing with Deadlines

The problem is stated as below

i. There are 'n' jobs to be processed on a machine.

ii. Each job 'i' has a deadline $\mathbf{d_i \geq 1}$ and profit $\mathbf{p_i \geq 0}$.

iii. $\mathbf{p_i}$ is earned if and only if the job is completed by its' deadline.

iv. processing time for each job on a machine is one unit time(say 1μs or 1ms).

v. Only one machine is available for processing jobs.

vi. only one job is processed at a time on the machine.

vii. A feasible solution is a subset of jobs 'J' such that

viii. An optimal solution is a feasible solution with maximum profit value.

Problem1

Let n = 4, (P1, P2, P3, P4,) = (100, 10, 15, 27) and (d1 d2 d3 d4) = (2, 1, 2, 1). The feasible solutions and their values are:

Solution

Sl. No.	Feasible Solution	Processing Sequence	Value	Remarks
1	1,2	2,1	110	
2	1,3	1,3 or 3,1	115	
3	**1,4**	**4,1**	**127**	**optimal**
4	2,3	2,3	25	
5	3,4	4,3	42	
6	1	1	100	
7	2	2	10	
8	3	3	15	
9	4	4	27	

Optimal solution

To solve this problem, the given jobs are sorted according to their profit in a descending order. Hence, after sorting, the jobs are ordered as shown in the following table.

Jobs	J1	J4	J3	J2
Deadlines	2	1	2	1
Profits	100	27	15	10

From this set of jobs, first we select *J1*, as it can be completed within its deadline and contributes maximum profit.

Next J4 contributes 2nd maximum profit but its' deadline is 1. If J1 will execute first then J4 can not be execute. Only one possibility left to earn the maximum profit i.e. to change the processing sequence of jobs of their execution i.e. in 1st slot J4 will be executed then J1(its' deadline 2). So the sequence is 4,1.

Next J3(deadline 2)will not occupied the slot because 1st two slots already occupied.

Next J2 similar as J3.

Algorithm

Job-Sequencing-With-Deadline (D, J, n)

```
{//D[i] ≥ 1, 1 ≤ i ≤ n are the deadlines, n ≥ 1. The jobs are order such that
//p[1] ≥ p[2] ≥ .....≥ p[n]. J[i] is the ith job in the optimal solution, 1 ≤ i ≤ k. Also, at
//termination D[J[i]] ≤ D[J[i + 1]],  1 ≤ i ≤ k//
    D[0] = J[0] = 0;
     k = 1;
    J[1] = 1;                    // job one is inserted into J//
    for i = 2 to n do
    {//consider jobs in decreasing order of p[i]. Find position for i and check
//feasibility for insertion.
        r = k;                   // r and k are indices for existing job in J//
        while((D(J[r]) > D[i]) && (D(J[r]) ≠ r) do     // find r such that i can be
                                                        inserted after r //
        r= r – 1;    // job r can be processed after i and deadline of job r is not
                     exactly r //
        if((D(J[r]) ≤ D[i]) && (D[i] > r) then
        {// the new job i can come after existing job r; insert i into J at position
        r+1 //
            for q = k to r + 1 step -1 do
            J[q + 1] = J[q];    // shift jobs(r+1) to k right by one position //
            J[r + 1] = i; k = k + 1;    // i is inserted at position r+1 and total
                                        jobs in J are increased by one //
        }
    }
    return k;
}
```

Time Complexity: Two nested loop used so $O(n^2)$.

Problem2

Given the jobs, their deadlines and associated profits as shown

Jobs	J1	J2	J3	J4	J5	J6
Deadlines	5	3	3	2	4	2
Profits	200	180	190	300	120	100

Answer the following questions-

1. Write the optimal schedule that gives maximum profit.
2. Are all the jobs completed in the optimal schedule?
3. What is the maximum earned profit?

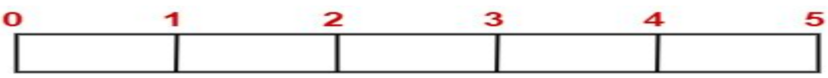
Solution

1. Sort all the given jobs in decreasing order of their profit-

Jobs	J4	J1	J3	J2	J5	J6
Deadlines	2	5	3	3	4	2
Profits	300	200	190	180	120	100

2. Value of maximum deadline = 5.

So, draw a Gantt chart with maximum time on Gantt chart = 5 units as shown-



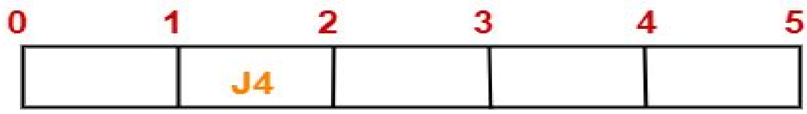
Gantt Chart

Now,

- We take each job one by one in the order they appear in Step1.
- We place the job on Gantt chart as far as possible from 0.

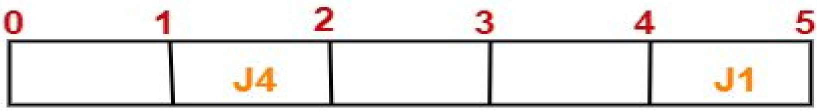
3.

- We take job J4.
- Since its deadline is 2, so we place it in the first empty cell before deadline 2 as-



4.

- We take job J1.
- Since its deadline is 5, so we place it in the first empty cell before deadline 5 as-



5.

- We take job J3.
- Since its deadline is 3, so we place it in the first empty cell before deadline 3 as-

6.

- We take job J2.
- Since its deadline is 3, so we place it in the first empty cell before deadline 3.
- Since the second and third cells are already filled, so we place job J2 in the first cell as-

7.

- Now, we take job J5.
- Since its deadline is 4, so we place it in the first empty cell before deadline 4 as-

Now,

- The only job left is job J6 whose deadline is 2.
- All the slots before deadline 2 are already occupied.
- Thus, job J6 can not be completed.

The optimal schedule is-

J2, J4, J3, J5, J1

This is the required order in which the jobs must be completed in order to obtain the maximum profit.

- All the jobs are not completed in optimal schedule.
- This is because job J6 could not be completed within its deadline.

Maximum earned profit
= Sum of profit of all the jobs in optimal schedule

= Profit of job J2 +Profit of job J4 +Profit of job J3 +Profit of job J5 +Profit of job J1

= 180 + 300 + 190 + 120 + 200

= 990