JOB SEQUENCING WITH DEADLINES

The problem is stated as below.

- There are n jobs to be processed on a machine.
- Each Job_i has a deadline $d_i \geq 1$ and profit $p_i \geq 1$.
- Pi is earned iff the job is completed by its deadline.
- The job is completed if it is processed on a machine for unit time.
- Only one machine is available for processing jobs.
- Only one job is processed at a time on the machine.

JOB SEQUENCING WITH DEADLINES (Contd..)

- A feasible solution is a subset of jobs J such that each job is completed by its deadline.
- An optimal solution is a feasible solution with maximum profit value.

Example : Let n = 4, $(p_1,p_2,p_3,p_4) = (100,10,15,27)$, $(d_1,d_2,d_3,d_4) = (2,1,2,1)$

JOB SEQUENCING WITH DEADLINES (Contd..)

Sr.No.	Feasible Solution	Processing Sequence	Profit value
(i)	(1,2)	(2,1)	110
(ii)	(1,3)	(1,3) or (3,1)	115
(iii)	(1,4)	(4,1)	127 is the optimal one
(iv)	(2,3)	(2,3)	25
(v)	(3,4)	(4,3)	42
(vi)	(1)	(1)	100
(vii)	(2)	(2)	10
(viii)	(3)	(3)	15
(ix)	(4)	(4)	27

GREEDY ALGORITHM TO OBTAIN AN OPTIMAL SOLUTION

- Consider the jobs in the non increasing order of profits subject to the constraint that the resulting job sequence J is a feasible solution.
- In the example considered before, the non-increasing profit vector is

$$\begin{array}{cccc} (100 & 27 & 15 & 10) \\ p_1 & p_4 & p_3 & p_2 \end{array} \qquad \begin{array}{cccc} (2 & 1 & 2 & 1) \\ d_1 & d_4 & d_3 & d_2 \end{array}$$

GREEDY ALGORITHM TO OBTAIN AN OPTIMAL SOLUTION (Contd..)

J = { 1} is a feasible one

J = { 1, 4} is a feasible one with processing sequence (4,1)

J = { 1, 3, 4} is not feasible

J = { 1, 2, 4} is not feasible

J = { 1, 4} is optimal

Job	**Profit**	**Deadline**
J1	10	2
J2	30	3
J3	60	1
J4	40	3

According to Job order on profits in non increasing order is J_3, J_4, J_2, J_1

Job	**Deadline**	**Operation**	**Slot**	**Total Profit**
Initialization			[0, 0, 0]	0
J_3	1	Assign J_3 to $Slot_1$	[J_3, 0, 0]	60
J_4	3	Assign J_4 to $Slot_3$.	[J_3, 0,J_4]	100
J_2	3	Assign J_2 to $Slot_2$ bcoz $Slot_3$ is not empty.	[J_3, J_2, J_4]	130
J_1	2	Reject J_1 bcoz no empty Slots before deadline.	[J_3, J_2, J_4]	130

Job	**Profit**	**Deadline**
J_1	15	7
J_2	20	2
J_3	30	5
J_4	18	3
J_5	18	4
J_6	10	5
J_7	23	2
J_8	16	7
J_9	25	3

```
Algorithm JobSequencing(J,p,d,n,slot)
//n is the number of jobs J[1:n], p[1:n], d[1:n]
//contains Job numbers, profits, deadlines
//slot[1:n] is the solution list.
{
      profit=0;
      for i=1 to n do slot[i]=0;
      for i=1 to n do
      {
            j=d[i];
            while(j>0) do
            {
                  if (slot[j] == 0) then
                  {
                        slot[j] = J[i]
                        profit = profit + p[i]
                        break;
                  }
                  j=j-1
            }
      }
      print slot & profit;
}
```