Job sequencing with deadlines

- Given a set of n jobs and one machine for processing jobs , and each job contains a deadline $d_i > 0$ (where d_i is an integer) and a profit $p_i > 0$. For any job *i* the profit p_i is earned iff the job is completed by its deadline. To complete a job, one has to process the job on a machine for one unit of time.
- The problem is to find an optimal subset of jobs that can be scheduled on a machine and all the jobs in the selected subset can be completed by its deadline.
- A **feasible solution** for this problem is a **subset J of jobs** such that each job in this subset can be **completed by its deadline**.
- .The value of a **feasible solution J** is the sum of the profits of the jobs in J, or $\sum_{i \in J} p_i$
- An **optimal solution** is a feasible solution with **maximum value**.

Example

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

S. No	Possibilities	Feasible ?
1	{1}	Yes
2	{2}	Yes
3	{3}	Yes
4	{4}	Yes
5	{1, 2}	Yes
6	{1, 3}	Yes
7	{1, 4}	Yes
8	{2, 3}	Yes
9	{2, 4}	No
10	{3, 4}	Yes
11	{1, 2, 3}	No
12	{1, 2, 4}	No
13	{1, 3, 4}	No
14	{2, 3, 4}	No
15	{1, 2, 3, 4}	No

machine

slot1	slot2

0 1 2

S.No	Feasible Solution	Processing Sequence	Value
1	(1,2)	2,1	110
2	(1,3)	1,3 or 3,1	115
3	**(1,4)**	**4,1**	**127**
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

Greedy Solution to JS with deadlines

Let n=4 , (p1,p2,p3,p4)= (100,10,15,27) and (d1,d2,d3,d4)= (2,1,2,1).

- Arrange the jobs in the decreasing (non-increasing) order of their profits. So, the job selection order is (J_1, J_4, J_3, J_2)
- Number of slots of the machine = min(n , max(deadline of jobs)) = min(4, 2) = 2

S.No	Job Selected	Slot allotted	Slots Available	Profit
			0 1 2	0
1	J_1	Is feasible? Yes Allot : [0,1]	J_1 0 1 2	100
2	J_4	Is feasible? Yes Move the job J_1 to [1,2] slot and allot [0,1] to job J_4	J_4 J_1 0 1 2	127

Job	P_i	d_i
J_1	100	2
J_4	27	1
J_3	15	2
J_2	10	1

```
Algorithm JS(d,j,n)
// d[i]>=1,1<=i<=n arethe deadlines, n>=1.The jobs are ordered 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;        // Initialize.
        J[1]:=1;              //  Include job1.
        k:=1;
        for i :=2 to n do
        {
        // Consider jobs in nonincreasing order of p[i]. Find position for i and check feasibility of insertion.
                r :=k;
                while ((d[J[r]]>d[i]) and (d[J[r]]!= r)) do r :=r -1;
                if ((d[J[r]]<=d[i])and (d[i] >r)) then
                {
                        // Insert i into J[].
                        for q :=k to (r + 1) step -1 do
                                J[q+ 1]:=J[q];
                        J[r+ 1]:= i; k :=k + 1;
                }
        }
        return k;
}
```