Experiment No: 6

Title: Write a program to find solution to Job Sequencing with Deadlines Problem Instance.

Theory/Description:

This problem consists of n jobs each associated with a deadline and profit and our objective is to earn maximum profit. We will earn profit only when job is completed on or before deadline. We assume that each job will take unit time to complete.

In this problem we have n jobs j1, j2, … jn each has an associated deadline d1, d2, … dn and profit p1, p2, ... pn.

Profit will only be awarded or earned if the job is completed on or before the deadline.

We assume that each job takes unit time to complete.

The objective is to earn maximum profit when only one job can be scheduled or processed at any given time.

Consider the following 5 jobs and their associated deadline and profit.

index	1	2	3	4	5
JOB	j1	j2	j3	j4	j5
DEADLINE	2	1	3	2	1
PROFIT	60	100	20	40	20

Sort the jobs according to their profit in descending order

Note! If two or more jobs are having the same profit then sort them as per their entry in the job list.

index	1	2	3	4	5
JOB	j2	j1	j4	j3	j5
DEADLINE	1	2	2	3	1
PROFIT	100	60	40	20	20

Find the maximum deadline value

Looking at the jobs we can say the max deadline value is 3.

So, dmax = 3

As dmax = 3 so we will have THREE slots to keep track of free time slots. Set the time slot status to EMPTY

time slot	1	2	3
status	EMPTY	EMPTY	EMPTY

Total number of jobs is 5.

So we can write n = 5

Note!

If we look at job j2, it has a deadline 1. This means we have to complete job j2 in time slot 1 if we want to earn its profit.

Similarly, if we look at job j1 it has a deadline 2. This means we have to complete job j1 on or before time slot 2 in order to earn its profit.

Similarly, if we look at job j3 it has a deadline 3. This means we have to complete job j3 on or before time slot 3 in order to earn its profit.

Our objective is to select jobs that will give us higher profit.

Sort the jobs based on decreasing order of their profits

```
Algorithm JobSequenceDeadlines(d,j,n)
//d[i] is a deadline for ith job and jobs are sorted and stored in p[] where p[1]>=p[2]
{
d[0]=0;
J[0]=0;
J[1]=1;
k=1;
for i:=2 to n do
{
  //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
        {
        J[q+1]:=J[q];
        }
        J[r+1]:=I;
        K:=k+1;
}
}
Return k;
}
```

Complexity: $O(n^2)$

Self-Study: Prove that complexity of of Job Sequencing with deadline is $O(n^2)$