11. Problem Statement: Job Sequence with Deadlines

**Problem Analysis:**

In job sequencing problem, the objective is to find a sequence of jobs, which is completed within their deadlines and gives maximum profit.

Let us consider, a set of **n** given jobs which are associated with deadlines and profit is earned, if a job is completed by its deadline. These jobs need to be ordered in such a way that there is maximum profit.

It may happen that all of the given jobs may not be completed within their deadlines.

Assume, deadline of **ith** job **Ji** is **di** and the profit received from this job is **pi**. Hence, the optimal solution of this algorithm is a feasible solution with maximum profit.

Thus, D(i)>0 for 1⩽i⩽n.

Initially, these jobs are ordered according to profit, i.e. p1⩾p2⩾p3⩾...⩾pn.

**Algorithm:**

Algorithm JS (d, j, n)

*// d [i] ≥ 1, 1 ≤ i ≤ n are the 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 job 1.*

k := 1;

for i := 2 to n do

{

*// Consider jobs in non-increasing 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;

}

**Source Code:**

#include<stdio.h>

int JobSequence(int d[], int j[], int n)

{

int q,i,r,k;

d[0]=j[0]=0;

j[1]=1;

k=1;

for(i=2; i<=n; i++)

{

r=k;

while((d[j[r]]>d[i])&&(d[j[r]]!=r))

r=r-1;

if((d[j[r]]<=d[i])&&(d[i]>r))

{

for(q=k;q<=r+1;q++)

{

j[q+1]=j[q];

j[r+1]=i;

k=k+1;

}

}

}

return k;

}

int main()

{

int n,i,k,t;

int d[10],j[10];

printf("Enter the no.of jobs:\n");

scanf("%d",&n);

printf("Enter the profit of jobs: \n",i+1);

for(t=0;t<n;t++)

scanf("%d",&j[i]);

printf("Enter the deadline of job: \n",i+1);

for(t=0;t<n;t++)

scanf("%d",&d[i]);

k=JobSequence(d,j,n);

printf("Solution: \n%d\n",k);

return 0;

}

**Sample Input:**

Enter the no.of jobs:

4

Enter the profit of jobs:

100 10 15 27

Enter the deadline of job:

2 1 2 1

**Sample Output:**

Solution:

3