**Assignment No.3 (d)**

**Job Scheduling Problem**

Given an array of jobs where every job has a deadline and associated profit if the job is finished before the deadline. It is also given that every job takes a single unit of time, so the minimum possible deadline for any job is 1. Maximize the total profit if only one job can be scheduled at a time.

**Examples:**

**Input:** Four Jobs with following deadlines and profits

JobID  Deadline  Profit

a           4          20     
  b           1          10  
  c           1          40    
  d          1          30

**Output:** Following is maximum profit sequence of jobs: c, a

**Input:**  Five Jobs with following deadlines and profits

JobID   Deadline  Profit

  a            2          100  
  b            1          19  
  c            2          27  
  d            1          25  
  e            3          15

**Output:** Following is maximum profit sequence of jobs: c, a, e

[**Greedy**](https://www.geeksforgeeks.org/greedy-algorithms-general-structure-and-applications/)**approach for job scheduling problem**

Greedily choose the jobs with maximum profit first, by sorting the jobs in decreasing order of their profit. This would help to maximize the total profit as choosing the job with maximum profit for every time slot will eventually maximize the total profit

**Algorithm :**

* Sort all jobs in decreasing order of profit.
* Iterate on jobs in decreasing order of profit. For each job , do the following :
  + Find a time slot i, such that slot is empty and i < deadline and i is greatest. Put the job in   
    this slot and mark this slot filled.
  + If no such i exists, then ignore the job.

**Time Complexity:** O(N2)  
**Auxiliary Space:** O(N)

**Conclusion :**

**Implementation :**

# function to schedule the jobs take 2

# arguments array and no of jobs to schedule

def printJobScheduling(arr, t):

# length of array

n = len(arr)

# Sort all jobs according to

# decreasing order of profit

for i in range(n):

for j in range(n - 1 - i):

if arr[j][2] < arr[j + 1][2]:

arr[j], arr[j + 1] = arr[j + 1], arr[j]

# To keep track of free time slots

result = [False] \* t

# To store result (Sequence of jobs)

job = ['-1'] \* t

# Iterate through all given jobs

for i in range(len(arr)):

# Find a free slot for this job

# (Note that we start from the

# last possible slot)

for j in range(min(t - 1, arr[i][1] - 1), -1, -1):

# Free slot found

if result[j] is False:

result[j] = True

job[j] = arr[i][0]

break

# print the sequence

print(job)

# Driver's Code

if \_\_name\_\_ == '\_\_main\_\_':

arr = [['a', 2, 100], # Job Array

['b', 1, 19],

['c', 2, 27],

['d', 1, 25],

['e', 3, 15]]

print("Following is maximum profit sequence of jobs")

# Function Call

printJobScheduling(arr, 3)

**Output**

Following is maximum profit sequence of jobs

c a e