|  |  |
| --- | --- |
|  | DEPARTMENT OF ARTIFICIAL INTELLIGNECE & DATA SCIENCE |

|  |  |
| --- | --- |
| Subject: Analysis of Algorithm | Course Code: CSC402 |
| Semester: 4 | Course: AI & DS |
| Laboratory No: 205 | Name of Subject Teacher: Pramod Bhavarthe |
| Name of Student: Sahil Shaikh | Roll Id: VU2S2223012 |

|  |  |
| --- | --- |
| Title of Practical | Job sequencing with deadlines. |

**Theory –**

**Job sequencing with deadlines** is a problem in which a set of jobs with different deadlines and profits associated with them need to be scheduled on a single machine. The objective is to maximize the total profit by scheduling the jobs in such a way that the deadlines are met.

The problem can be formally defined as follows: Given n jobs, where each job i has a deadline di and a profit pi, schedule the jobs on a single machine to maximize the total profit, subject to the constraint that each job must be completed before its deadline.

This problem can be solved using a greedy algorithm. The algorithm works as follows:

* Sort the jobs in decreasing order of their profits.
* For each job, starting from the highest profit job, assign the job to the latest possible time slot before its deadline that is still available.
* This algorithm guarantees that the maximum profit is achieved while meeting all the deadlines. The time complexity of this algorithm is O(nlogn) due to the sorting step.
* However, it is important to note that this algorithm assumes that there is only one machine available for scheduling the jobs. If there are multiple machines available, then the problem becomes more complex and requires a different algorithm to solve.

**Program –**

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\_\_':

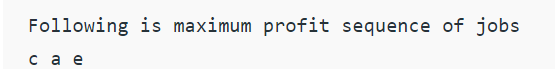
    arr = [['a', 2, 100], ['b', 1, 19], ['c', 2, 27],['d', 1, 25], ['e', 3, 15]]

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

    # Function Call

    printJobScheduling(arr, 3)

**Output –**

****

**Conclusion –**

**Therefore, we have successfully understood and Implemented Job sequencing with deadlines.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Grade and Dated Signature of Teacher** | **Total (10)** | **Remark** | **Dated signature of teacher** |
|  |  |  |