
SCHOOL OF COMPUTING

# DESIGN AND ANALYSIS OF ALGORITHMS

## LAB WORKBOOK

## WEEK - 7

NAME : VIGHRANTH

ROLL NUMBER : CH.SC.U4CSE24149

CLASS : CSE-B

**Question 1:** To implement the Greedy algorithm for the Job Sequencing with Deadlines problem and determine the optimal sequence of jobs that maximizes total profit.

Consider a set of 14 jobs, where each job requires one unit of time for completion. Each job has an associated profit and deadline. A job must be completed on or before its deadline to earn the profit.

The profits of the jobs are:

{22, 19, 29, 28, 30, 21, 27, 25, 24, 26, 14, 27, 19, 11}

The deadlines of the jobs are:

{3, 3, 8, 6, 7, 5, 10, 4, 6, 12, 13, 2, 14, 1}

**CODE:**

```c
//CH.SC.U4CSE24149 VIGHRANTH SK
#include <stdio.h>
#define MAX 50
struct Job {
    int id;
    int profit;
    int deadline;
};
int main() {
    printf("CH.SC.U4CSE24149 VIGHRANTH \n");
    int n, i, j;
    printf("Enter number of jobs: ");
    scanf("%d", &n);
    struct Job jobs[MAX];
    for(i = 0; i < n; i++) {
        jobs[i].id = i + 1;
        printf("\nEnter profit for Job %d: ", i+1);
        scanf("%d", &jobs[i].profit);
        printf("Enter deadline for Job %d: ", i+1);
```

```
        scanf("%d", &jobs[i].deadline);
    }
    for(i = 0; i < n-1; i++) {
        for(j = 0; j < n-i-1; j++) {
            if(jobs[j].profit < jobs[j+1].profit) {
                struct Job temp = jobs[j];
                jobs[j] = jobs[j+1];
                jobs[j+1] = temp;
            }
        }
    }
    int maxDeadline = 0;
    for(i = 0; i < n; i++) {
        if(jobs[i].deadline > maxDeadline)
            maxDeadline = jobs[i].deadline;
    }
    int slot[MAX];
    int totalProfit = 0;
    for(i = 0; i <= maxDeadline; i++)
        slot[i] = -1;
    for(i = 0; i < n; i++) {
        for(j = jobs[i].deadline; j > 0; j--) {
            if(slot[j] == -1) {
                slot[j] = jobs[i].id;
                totalProfit += jobs[i].profit;
                break;
            }
        }
    }
    printf("\n\nTime Slot\tJob\n");
    for(i = 1; i <= maxDeadline; i++) {
        if(slot[i] != -1)
            printf("%d\t\tJ%d\n", i, slot[i]);
```

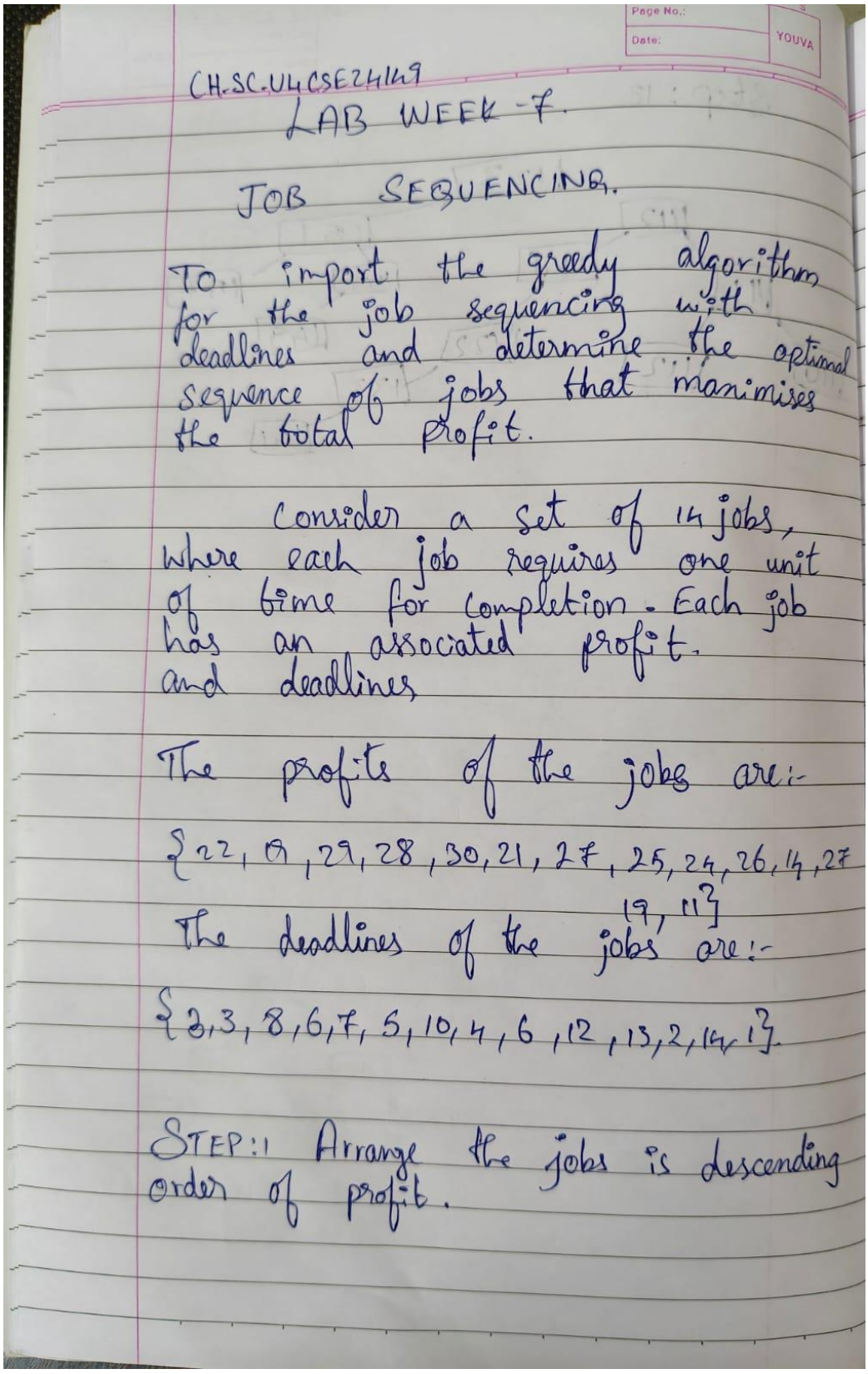
```
        else
            printf("%d\t\t--\n", i);
    }
    printf("\nMaximum Profit = %d\n", totalProfit);
    return 0;
}
```

**OUTPUT:**

```
E:\VIGHRANTH SK>gcc JobSequencing.c

E:\VIGHRANTH SK>a
CH.SC.U4CSE24149 - Vighranth Sk
Enter number of jobs: 10

Enter profit for Job 1: 45
Enter deadline for Job 1: 5

Enter profit for Job 2: 72
Enter deadline for Job 2: 5

Enter profit for Job 3: 94
Enter deadline for Job 3: 6

Enter profit for Job 4: 45
Enter deadline for Job 4: 6

Enter profit for Job 5: 94
Enter deadline for Job 5: 5

Enter profit for Job 6: 46
Enter deadline for Job 6: 5

Enter profit for Job 7: 31
Enter deadline for Job 7: 5

Enter profit for Job 8: 79
Enter deadline for Job 8: 4

Enter profit for Job 9: 93
Enter deadline for Job 9: 5

Enter profit for Job 10: 78
Enter deadline for Job 10: 1
```

```
Time Slot        Job
1                J10
2                J2
3                J8
4                J9
5                J5
6                J3

Maximum Profit = 510
```

**WORKING:**



CH.SC.U4CSE24149

LAB WEEK - 7.

JOB SEQUENCING.

To import the greedy algorithm for the job sequencing with deadlines and determine the optimal sequence of jobs that maximises the total profit.

Consider a set of 14 jobs, where each job requires one unit of time for completion. Each job has an associated profit. and deadlines

The profits of the jobs are:-

{22, 9, 29, 28, 30, 21, 27, 25, 24, 26, 14, 27, 19, 11}

The deadlines of the jobs are:-

{3, 3, 8, 6, 7, 5, 10, 4, 6, 12, 13, 2, 14, 1}.

STEP:1 Arrange the jobs is descending order of profit.

24149

| JOB | PROFIT | DEADLINE |
|---|---|---|
| $J_5$ | 30 | 7 |
| $J_3$ | 29 | 8 |
| $J_4$ | 28 | 6 |
| $J_7$ | 27 | 10 |
| $J_{12}$ | 27 | 2 |
| $J_{10}$ | 26 | 12 |
| $J_8$ | 25 | 4 |
| $J_9$ | 24 | 6 |
| $J_1$ | 22 | 3 |
| $J_6$ | 21 | 5 |
| $J_2$ | 19 | 3 |
| $J_{13}$ | 19 | 14 |
| $J_{11}$ | 14 | 18 |
| $J_{14}$ | 11 | 1 |

Since the maximum value of deadlines is 14, forteen times 2 lots are created

Step-2:- Assign jobs at the optimal time slots.

01) Add $J_5$ (30, 7)

→ $J_5$ can be added in slot 7 since it is empty

Sequence is $\{J_5\}$ and the profit is 30

Date: YOUVA

24149

2) Add $J_3$ (29,8)
→ $J_3$ can be added in slot 8 since it is empty

→ Sequence in $\{J_5, J_3\}$ and the profit is 59

3) Add $J_3$ (28,6)
→ $J_4$ can be added in slot 6 since it is empty

→ Sequence is $\{J_5, J_3, J_4\}$ and the profit is 87

4) Add $J_7$ (27,10)
→ $J_7$ can be added is slot 10 since it is empty

→ Sequence is $\{J_5, J_3, J_4, J_7\}$ and the profit is 114

5) Add $J_{12}$ (27,2)
→ $J_{12}$ can be added in slot 2 since it is empty

→ Sequence is $\{J_5, J_3, J_4, J_7, J_{12}\}$ and then profit 141

6) Add $J_{10}$ (26,4)
→ $J_{10}$ can be added in slot 4 since it is empty

→ Sequence is $\{J_5, J_3, J_4, J_7, J_{12}, J_{10}\}$ and the profit is 166

7) Add $J_8$ (25,4)
→ $J_8$ can be added in slot 4 since it is empty

→ Sequence is $\{J_5, J_3, J_4, J_7, J_{12}, J_{10}, J_8\}$ & profit 192

Page No.:
Date:
YOUVA

8) Add $J_9$ (24, 6)
→ $J_9$ cannot be added in slot 6 since it is occupied
→ But it can be added to slot-5 since it is empty.

→ Sequence is $\{J_5, J_3, J_4, J_7, J_{12}, J_{10}, J_8, J_9\}$ and p-216

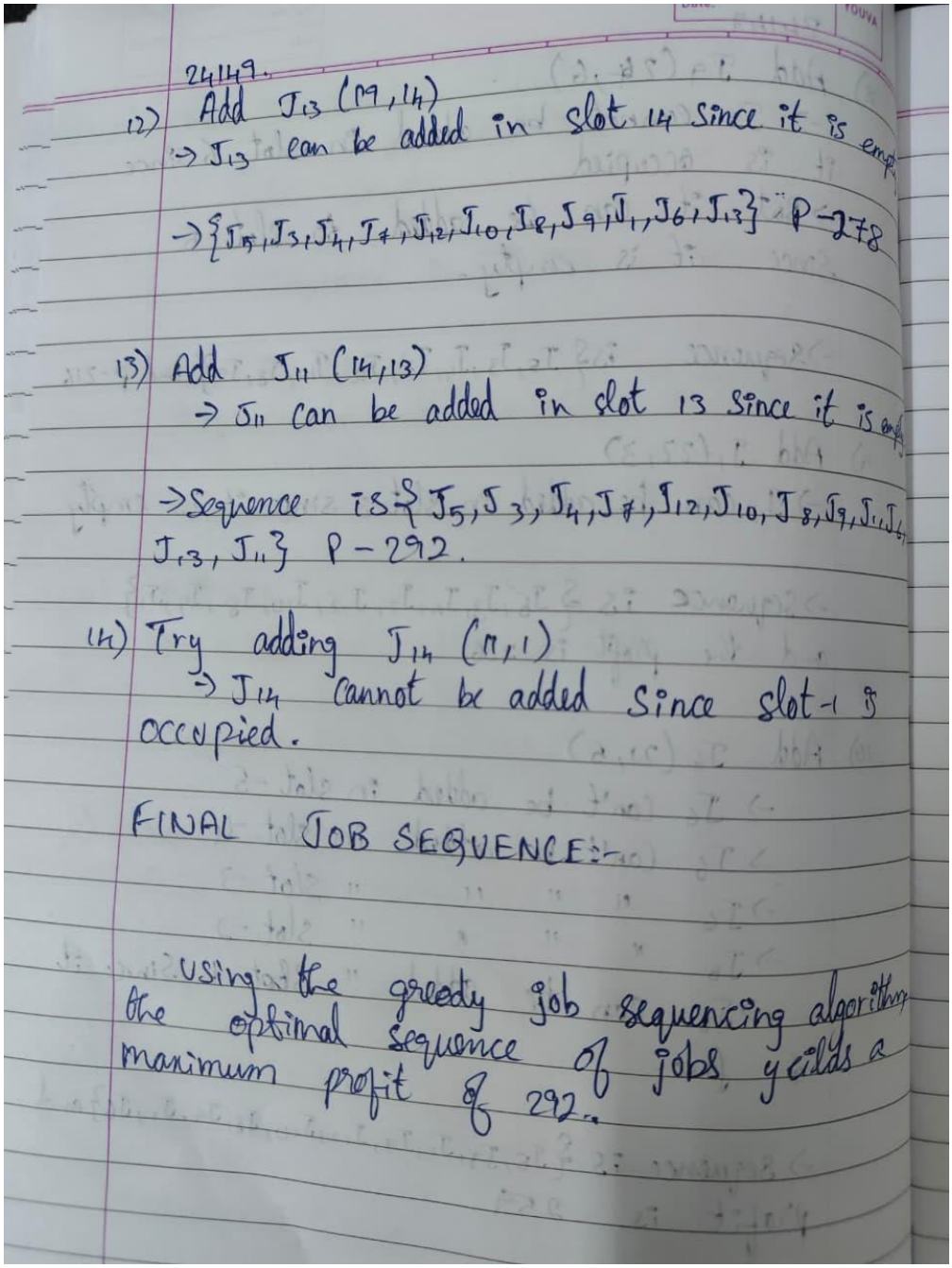
9) Add $J_1$ (22, 3)
→ $J_1$ can be added in slot 3 since it is empty

→ Sequence is $\{J_5, J_3, J_4, J_7, J_{12}, J_{10}, J_8, J_9, J_1\}$ and the profit is 236.

10) Add $J_6$ (21, 5)
→ $J_6$ can't be added in slot-5
→ $J_6$ can't be added in slot-4
→ $J_6$ " " " " slot-3
→ $J_6$ " " " " slot-2
→ $J_6$ can be added " slot-1 since it is free

→ sequence is $\{J_5, J_3, J_4, J_7, J_{12}, J_{10}, J_8, J_9, J_1, J_6\}$ and profit is 259

11) Add $J_2$ (19, 3)
→ $J_2$ can't be added in slot-3
→ " " " " " slot-2.
also in slot-1
∴ $J_2$ is being skipped




**Time Complexity: O($N^2$) :** The time complexity of the program is O($N^2$) because it uses bubble sort to arrange jobs by profit and a nested loop to assign jobs to time slots in the worst case.

**Space Complexity: O(N) :** The space complexity is O(N) since we use arrays to store the jobs and time slots proportional to the number of jobs.