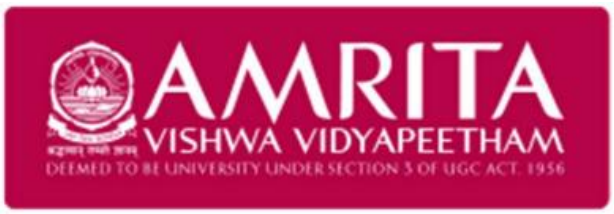


SCHOOL OF COMPUTING

DESIGN AND ANALYSIS OF ALGORITHMS

LAB WORKBOOK

WEEK - 6

NAME : Ganath Avinash G.R

ROLL NUMBER : CH.SC.U4CSE24118

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:

```
#include <stdio.h>
struct Job{
   int id;
   int p;
   int d;
};
void sort(struct Job j[], int n){
   struct Job temp;
   for(int i = 0; i < n-1; i++){
      for(int k = 0; k < n-i-1; k++){
         if(j[k].p < j[k+1].p){
            temp = j[k];
            j[k] = j[k+1];
            j[k+1] = temp;
         }
      }
```

```
    }
}
int main(){
    int n = 0;
    printf("Enter no of Jobs:\n");
    scanf("%d", &n);
    struct Job j[n];
    for (int i = 0; i < n; i++){
        printf("Enter job %d profit and deadline:\n", i+1);
        j[i].id = i+1;
        scanf("%d %d", &j[i].p, &j[i].d);
    }
    sort(j, n);
    int max = 0;
    for(int i = 0; i < n; i++){
        if(j[i].d > max)
            max = j[i].d;
    }
    int slot[max+1];
    int totalProfit = 0;
    for(int i = 0; i <= max; i++)
        slot[i] = -1;
    for(int i = 0; i < n; i++){
        for(int k = j[i].d; k > 0; k--){
            if(slot[k] == -1){
                slot[k] = j[i].id;
                totalProfit += j[i].p;
                break;
```

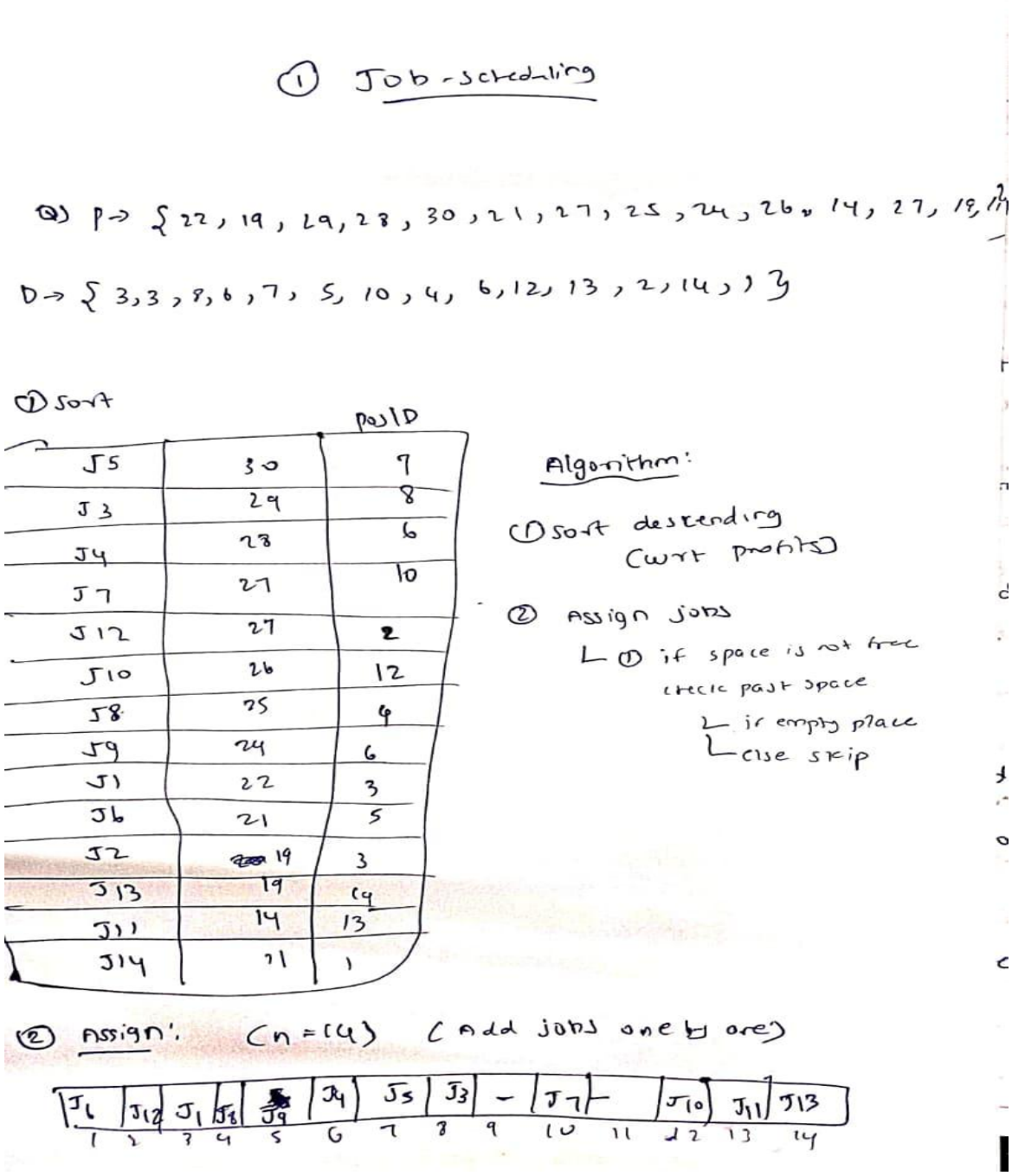
```
            }
        }
    }

    printf("\nSelected Jobs:\n");
    for(int i = 1; i <= max; i++){
        if(slot[i] != -1)
            printf("Job %d\n", slot[i]);
        else{
            printf("-");
        }
    }

    printf("Total Profit = %d\n", totalProfit);

    return 0;
}
```

OUTPUT:

```
PS C:\Users\Ganath Avinash\OneDriv
jobsequence.c -o jobsequence } ; i
Enter no of Jobs:
14
Enter job 1 profit and deadline:
22 3
Enter job 2 profit and deadline:
19 3
Enter job 3 profit and deadline:
29 8
Enter job 4 profit and deadline:
28 6
Enter job 5 profit and deadline:
30 7
Enter job 6 profit and deadline:
21 5
Enter job 7 profit and deadline:
27 10
Enter job 8 profit and deadline:
25 4
Enter job 9 profit and deadline:
24 6
Enter job 10 profit and deadline:
26 12
Enter job 11 profit and deadline:
14 13
Enter job 12 profit and deadline:
27 2
Enter job 13 profit and deadline:
19 14
Enter job 14 profit and deadline:
11 1
```

```
Selected Jobs:
Job 6
Job 12
Job 1
Job 8
Job 9
Job 4
Job 5
Job 3
-Job 7
-Job 10
Job 11
Job 13
Total Profit = 292
PS C:\Users\Ganath Avinash\OneDriv
```

WORKING:



Time Complexity: O(N²) : The time complexity of the program is O(N²) because it uses bubble sort.

Space Complexity: O(N) : The space complexity is O(N) since we use arrays to store the jobs .