DESIGN AND ANALYSIS OF ALGORITHMS

LAB WORKBOOK WEEK – 7

NAME: CHANTATI SAI PURNISHA MAHI

ROLL NUMBER: CH.SC.U4CSE24156

CLASS: CSE-B

DATE: 15-02-2026

JOB SCHEDULING (GREEDY ALGORITHM):-

CODE:

```c
#include <stdio.h>

#define MAX 100

struct Job
{
    int profit;
    int deadline;
};

// Sort jobs by profit (descending)
void sortJobs(struct Job jobs[], int n)
{
    struct Job temp;

    for(int i = 0; i < n - 1; i++)
    {
        for(int j = 0; j < n - i - 1; j++)
        {
            if(jobs[j].profit < jobs[j + 1].profit)
            {
                temp = jobs[j];
                jobs[j] = jobs[j + 1];
                jobs[j + 1] = temp;
            }
        }
    }
}
```

```c
int main()
{
    struct Job jobs[MAX];
    int n;

    printf("Enter number of jobs: ");
    scanf("%d", &n);

    // Input profits
    printf("\nEnter profits of %d jobs:\n", n);
    for(int i = 0; i < n; i++)
        scanf("%d", &jobs[i].profit);

    // Input deadlines
    printf("\nEnter deadlines of %d jobs:\n", n);
    for(int i = 0; i < n; i++)
        scanf("%d", &jobs[i].deadline);

    // Sort by profit
    sortJobs(jobs, n);

    // Find maximum deadline
    int maxDeadline = 0;
    for(int i = 0; i < n; i++)
        if(jobs[i].deadline > maxDeadline)
            maxDeadline = jobs[i].deadline;
```

```
    int slots[MAX] = {0};   // stores profit in slot
    int totalProfit = 0;

    // Greedy scheduling
    for(int i = 0; i < n; i++)
    {
        for(int j = jobs[i].deadline - 1; j >= 0; j--
        {
            if(slots[j] == 0)
            {
                slots[j] = jobs[i].profit;
                totalProfit += jobs[i].profit;
                break;
            }
        }
    }

    // Output sequence
    printf("\nScheduled Slots:\n");
    for(int i = 0; i < maxDeadline; i++)
    {
        if(slots[i] == 0)
            printf("_ ");
        else
            printf("%d ", slots[i]);
    }

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

```
    return 0;
}
```

OUTPUT:

```
root@ubuntu:/home/purnisha# nano job1.c
root@ubuntu:/home/purnisha# gcc job1.c -o job1
root@ubuntu:/home/purnisha# ./job1
Enter number of jobs: 14

Enter profits of 14 jobs:
22 19 29 28 30 21 27 25 24 26 14 27 19 11

Enter deadlines of 14 jobs:
3 3 8 6 7 5 10 4 6 12 13 2 14 1

Scheduled Slots:
21 27 22 25 24 28 30 29 _ 27 _ 26 14 19

Total Profit = 292
root@ubuntu:/home/purnisha#
```

TIME COMPLEXITY:- $O(n^2)$

JUSTIFICATION:-

Sorting jobs using Bubble Sort → $O(n^2)$

Finding maximum deadline (single loop) → $O(n)$

Scheduling jobs (nested loops) → $O(n^2)$

Overall Time = $O(n^2)$

SPACE COMPLEXITY:- $O(n)$

JUSTIFICATION:-

jobs[n] → n structures (profit, deadline) → 2 integers each

→ n × 2 × 4 bytes = 8n bytes

slots[n] → n integers

→ n × 4 bytes = 4n bytes

Other variables:

n, maxDeadline, totalProfit, i, j → 5 integers

→ 5 × 4 bytes = 20 bytes

Total memory used = 8n + 4n + 20 = 12n + 20 bytes

Ignoring constants ⇒ Space Complexity = O(n)

Job sequencing (Greedy method).

Q) Let there be 14 jobs with the profit 22, 19, 29, 28, 30, 21, 27, 25, 24, 26, 14, 27, 19, 11 with deadlines - 3, 3, 8, 6, 7, 5, 10, 4, 6, 12, 13, 2, 14, 1

Sol:- No of Jobs (n) = 14

Profits corresponding to Jobs J_1 to J_{14} are P_1 to P_{14}

$(P_1$ to $P_{14}) = (22, 19, 29, 28, 30, 21, 27, 25, 24, 26, 14, 27, 19, 11)$

$(D_1$ to $D_{14}) = (3, 3, 8, 6, 7, 5, 10, 4, 6, 12, 13, 2, 14, 1)$

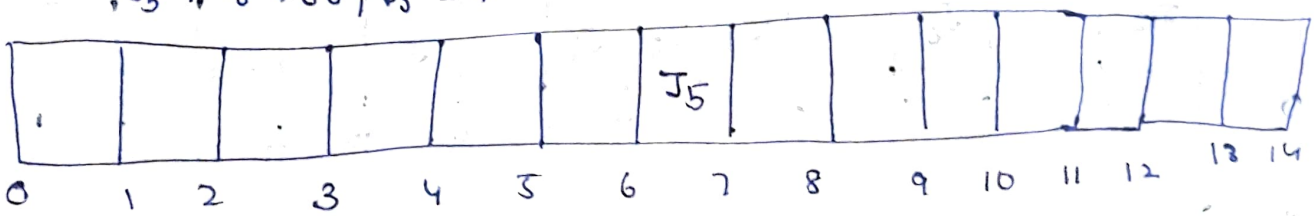
Step 1:-

Arrange the jobs in descending order based on profits and write corresponding deadlines.

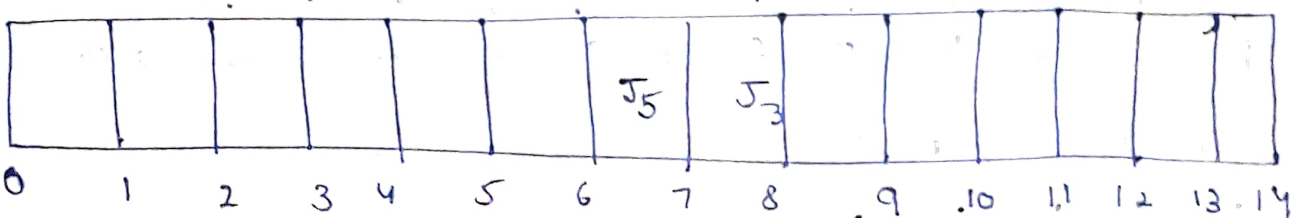
30	29	28	27	27	26	25	24	22	21	19	19	14	11
7	8	6	10	2	12	4	6	3	5	3	14	13	1
J_5	J_3	J_4	J_7	J_{12}	J_{10}	J_8	J_9	J_1	J_6	J_2	J_{13}	J_{11}	J_{14}

Step 2:- Create slots & Assign jobs

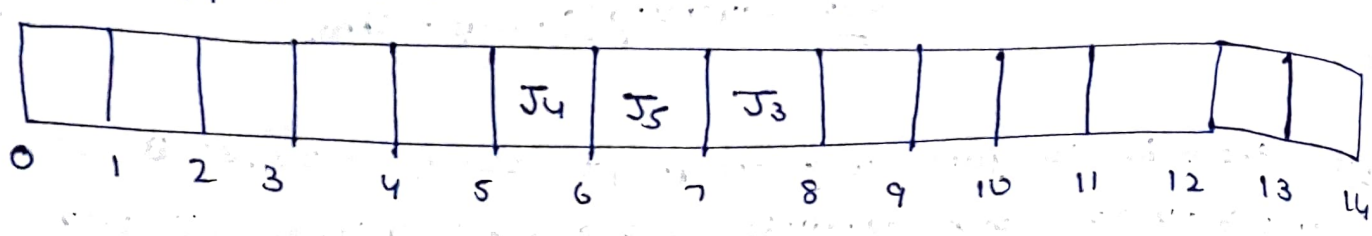
① J_5, $P_5 = 30$, $D_5 = 7$



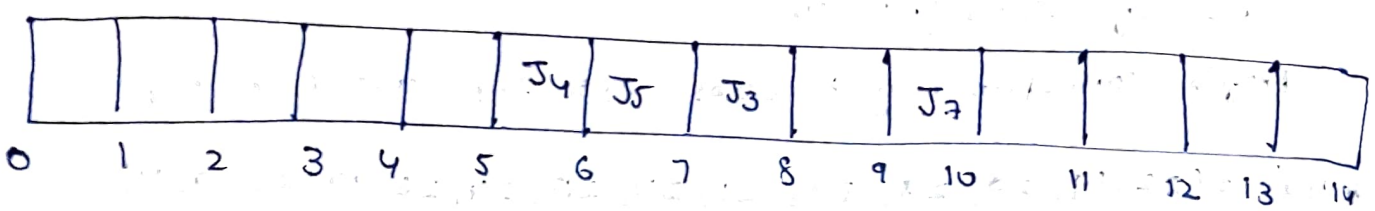
② J_3, $P_5 = 29$, $D = 8$



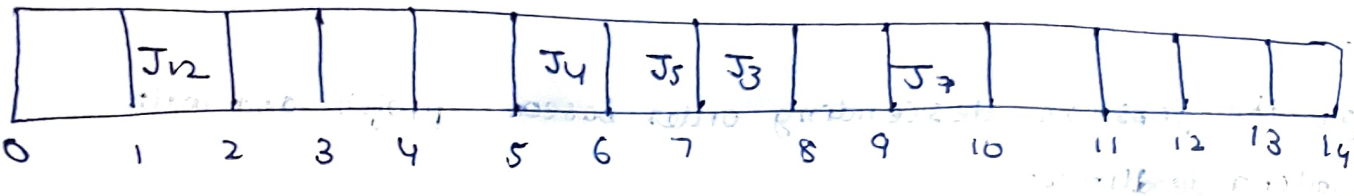
③ J_4, $P_4 = 28$, $D_4 = 6$



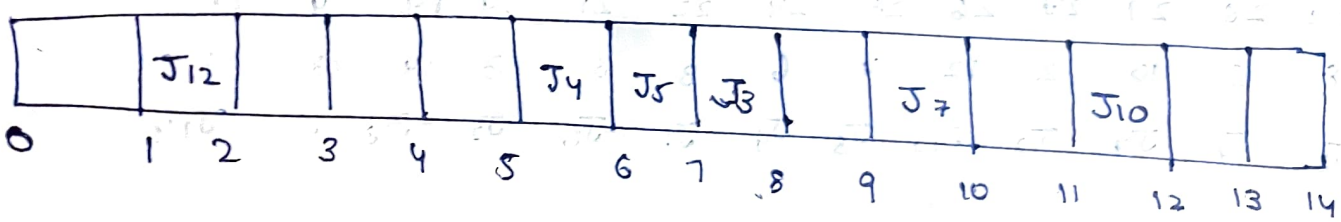
④ J_7, $P_7 = 27$, $D_7 = 10$



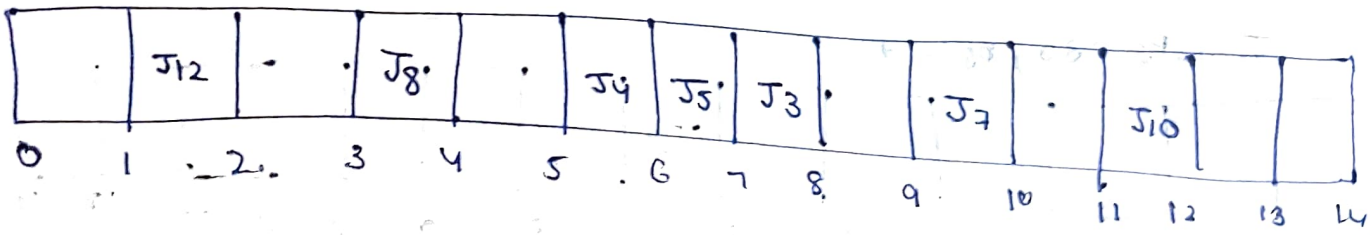
⑤ J_{12}, $P_{12} = 27$, $D_{12} = 2$



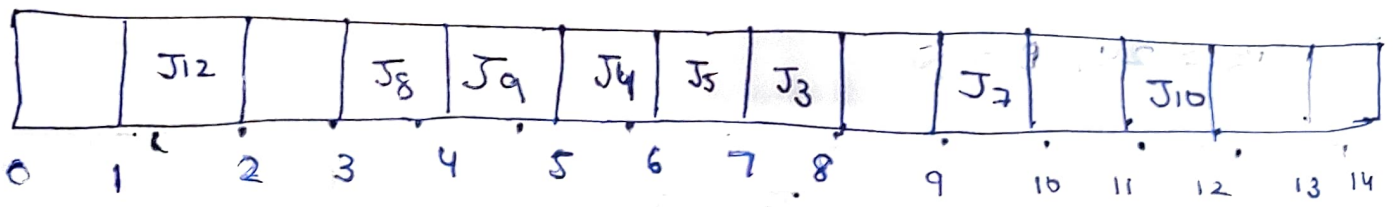
⑥ J_{10}, $P_{10} = 26$, $D_{10} = 12$



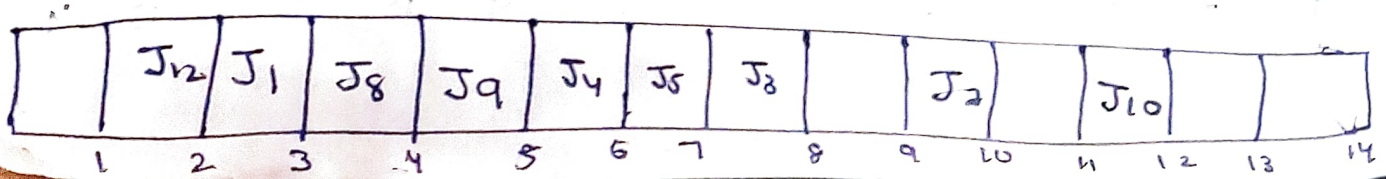
⑦ J_8, $P_8 = 25$, $P_8 = 4$.



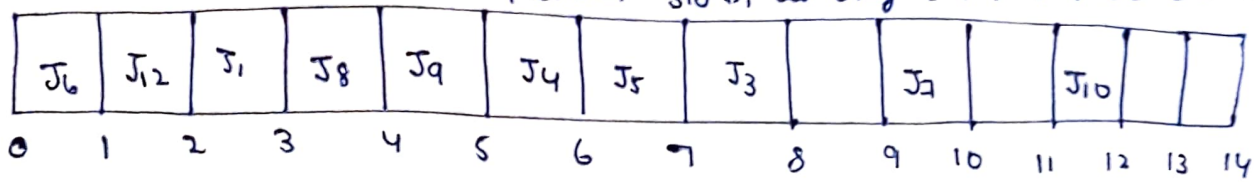
⑧ J_9, $P_9 = 24$, $D_9 = 6$



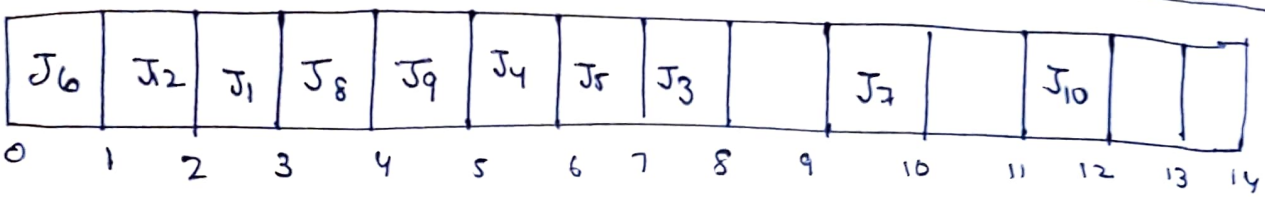
⑨ J_1, $P_1 = 22$, $P_1 = 3$



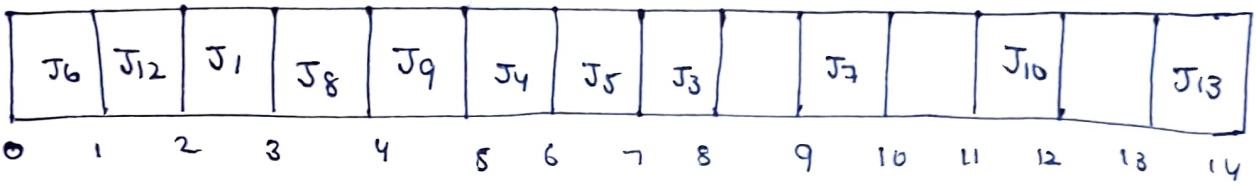
(10) J_6, $P_6 = 21$, $D_6 = 5$ [4-5] slot is already assigned, check previous slots, as only [0, 1] is free slot it with J_6



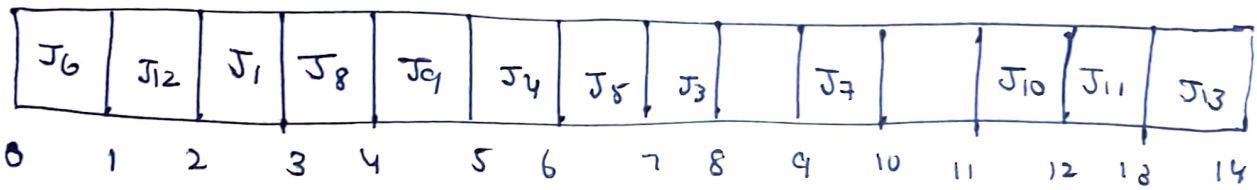
(11) J_2, $P_2 = 19$, $D_2 = 3$ All slots before deadline i.e, 3 are alloted already. So no slot for J_2 **Reject J_2**



(12) J_{13}, $P_{13} = 19$, $D_{13} = 14$

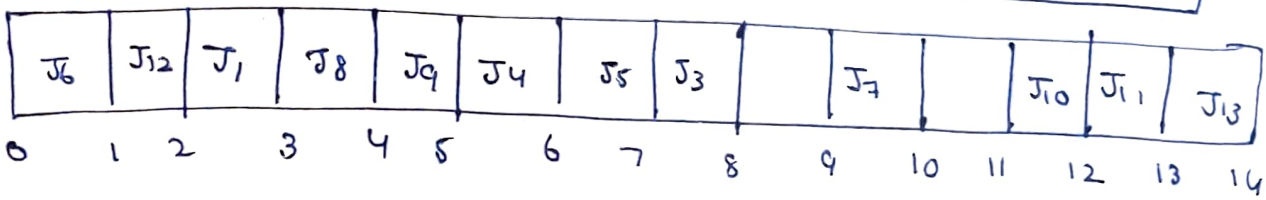


(13) J_{11}, $P_{11} = 14$, $D_{11} = 13$

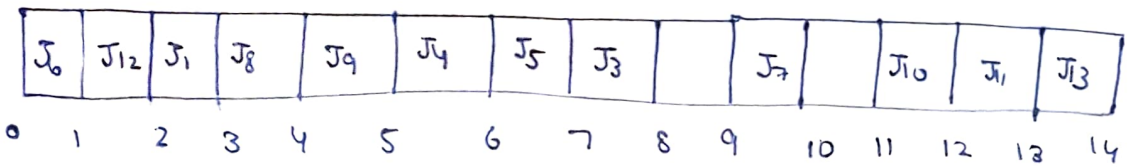


(14) J_{14}, $P_{14} = 11$, $D_{14} = 1$

As deadline is 1, there are slots left for P_{14} so **Reject J_{14}**



Final Job sequence = { J_5, J_3, J_4, J_7, J_{12}, J_{10}, J_8, J_9, J_1, J_6, J_{13}, J_{11} }



Total Profit = 21 + 27 + 22 + 25 + 24 + 28 + 30 + 29 + 27 + 26 + 14 + 19

= 292.