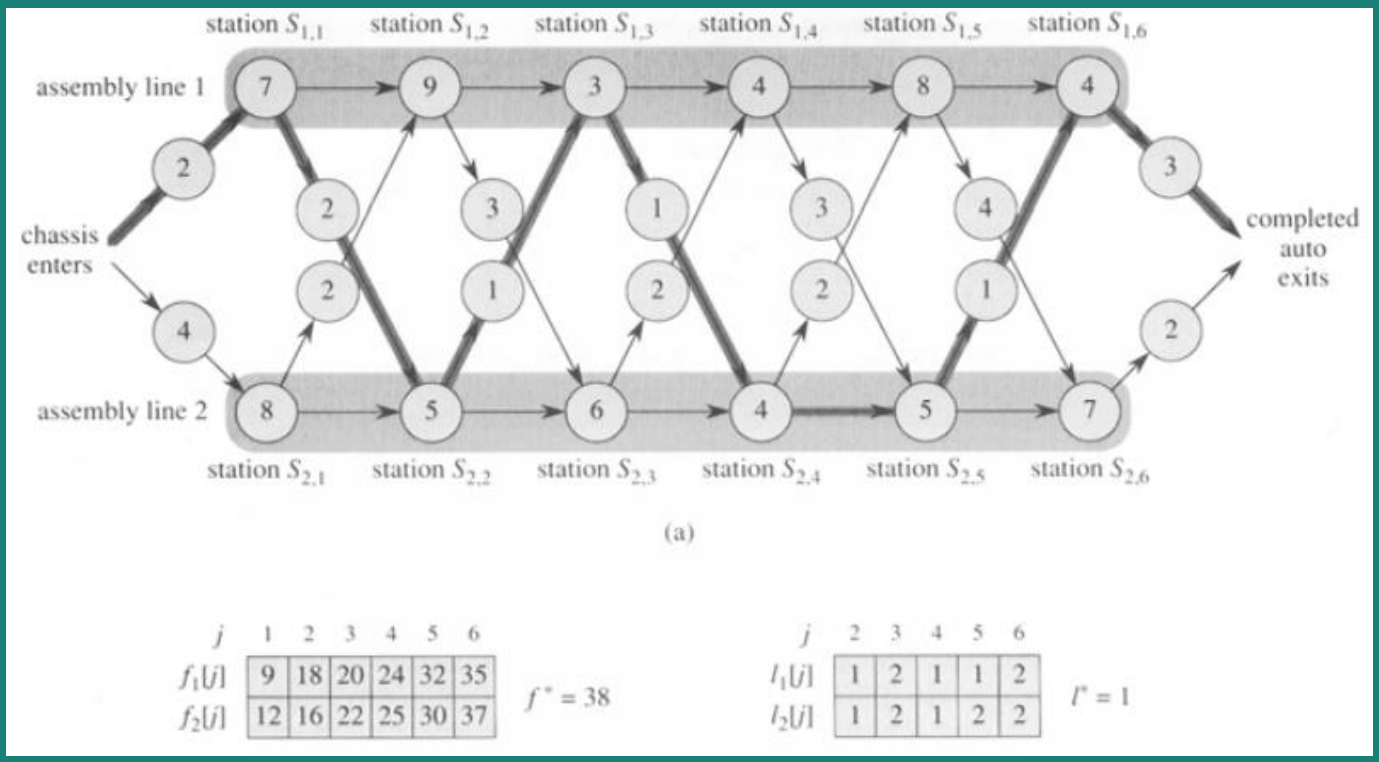


j	1	2	3	4	5	6
$f_1[j]$	9	18	20	24	32	35
$f_2[j]$	12	16	22	25	30	37

$f^* = 38$

j	2	3	4	5	6
$l_1[j]$	1	2	1	1	2
$l_2[j]$	1	2	1	2	2

$l^* = 1$

CTA PRACTICAL 7 20MCED08

Practical 7

Implement Assembly Line Scheduling problem using dynamic programming concepts.

Introduction

There are two assembly lines. Each assembly line has n stations.

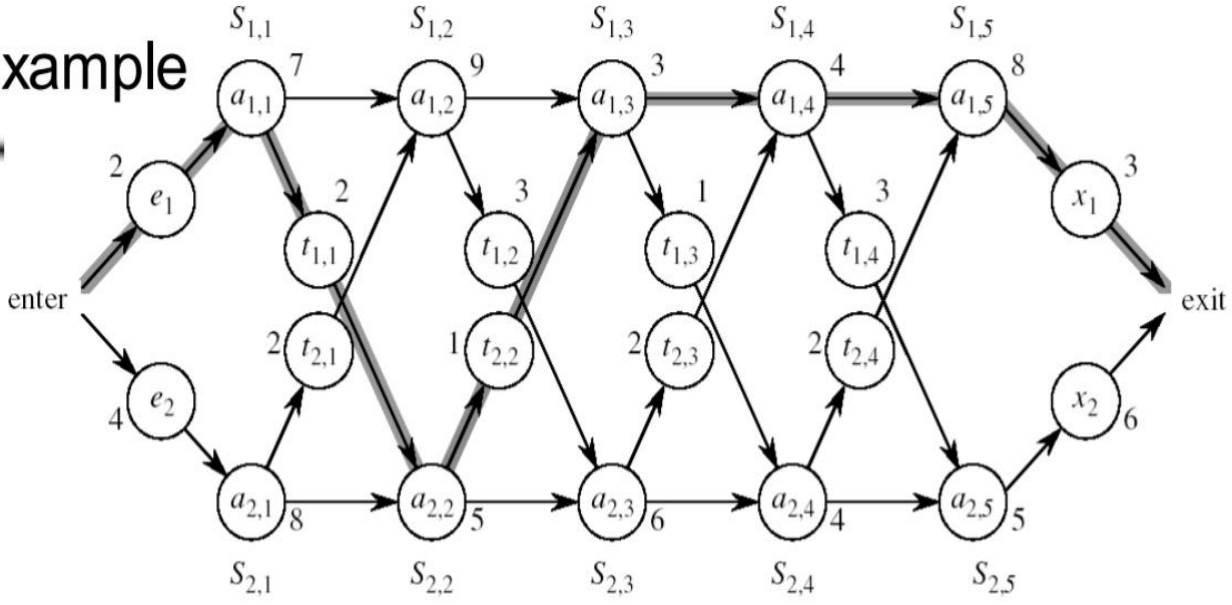
Every station has some dedicated job that needs to done. For a station number i, you can get the job done on station number i of any assembly line.

You can go to station i only when you have been through station i-1.
From a station i, you can either go to the next station on the same line or you can transfer assebly line.

Information provided –

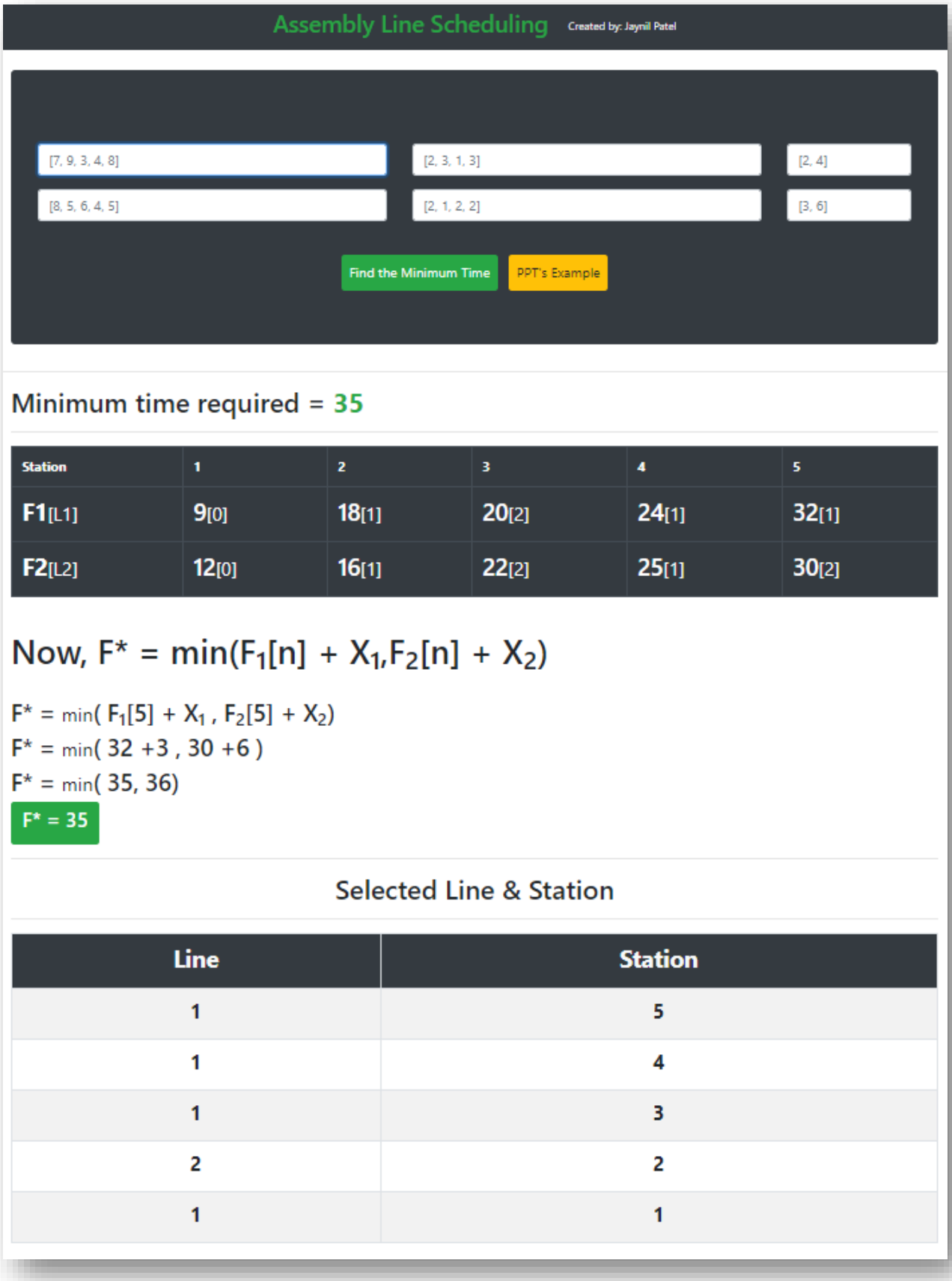
- time to enter station 1 on both assembly lines
- time to exit last station on both lines
- job processing time for every station
- The time to transfer from each station of assembly line 1 to next station of assembly line 2 is given
- The time to transfer from each station of assembly line 2 to next station of assembly line 1

Example



$$f_1[j] = \begin{cases} e_1 + a_{1,1}, & \text{if } j = 1 \\ \min(f_1[j-1] + a_{1,j}, f_2[j-1] + t_{2,j-1} + a_{1,j}) & \text{if } j \geq 2 \end{cases}$$

Assembly Line Scheduling (Simulation) (Python+Flask)



Assembly Line Scheduling (Python)

```python
def als (a1,a2,t1,t2,e,x,n):

    T1 = [[0,0] for i in range(n)]
    T2 = [[0,0] for i in range(n)]

    T1[0][0] = e[0] + a1[0]

    T2[0][0] = e[1] + a2[0]


    for i in range(1, n):
        temp1 = T1[i-1][0] + a1[i]
        temp2 = T2[i-1][0] + t2[i-1] + a1[i]
        if temp1 < temp2:
            T1[i][0] = temp1
            T1[i][1] = 1
        else:
            T1[i][0] = temp2
            T1[i][1] = 2

        temp3 = T2[i-1][0] + a2[i]
        temp4 = T1[i-1][0] + t1[i-1] + a2[i]
        if temp3 < temp4:
            T2[i][0] = temp3
            T2[i][1] = 2
        else:
            T2[i][0] = temp4
            T2[i][1] = 1
        temp5 = T1[n-1][0] + x[0]
        temp6 = T2[n-1][0] + x[1]
        if temp5 < temp6:
            T1[n-1][0] = temp5
            T1[n-1][1] = 1
            lstar = 1
        else:
            T2[n-1][0] = temp6
            T2[n-1][1] = 2
            lstar = 2


    finalT = [None,None]
    finalT[0] = T1
    finalT[1] = T2
    print(finalT[0])
    print(finalT[1])


    i = lstar
    print("\nLine : ",lstar," Station : ",n)
    for j in range(n-1,0,-1):
        i = finalT[i-1][j][1]
        print("Line : ",i," Station : ",j)


    return min(temp5,temp6)
```

```
a1 = [7,9,3,4,8]
a2 = [8,5,6,4,5]
t1 = [2,3,1,3]
t2 = [2,1,2,2]
e = [2,4]
x = [3,6]

print("\nAns = ",als(a1,a2,t1,t2, e, x,len(a1)))

# By Jaynil Patel (20MCED08)
```

Output :

```
[[9, 0], [18, 1], [20, 2], [24, 1], [35, 1]]
[[12, 0], [16, 1], [22, 2], [25, 1], [30, 2]]

Line :  1  Station :  5
Line :  1  Station :  4
Line :  1  Station :  3
Line :  2  Station :  2
Line :  1  Station :  1

Ans =  35
```

Observation:

It takes constant time to calculate initial F1 and F2 then to calculate remaning values using for loop it will take O(n) time.