4.3 MULTI-ASSEMBLY LINE SCHEDULING WITH DEPENDENCIES

Question:

An automotive company has three assembly lines (Line 1, Line 2, Line 3) to produce different car models. Each line has a series of stations, and each station takes a certain amount of time to complete its task. Additionally, there are transfer times between lines, and certain dependencies must be respected due to the sequential nature of some tasks. Your goal is to minimize the total production time by determining the optimal scheduling of tasks across these lines, considering the transfer times and dependencies.

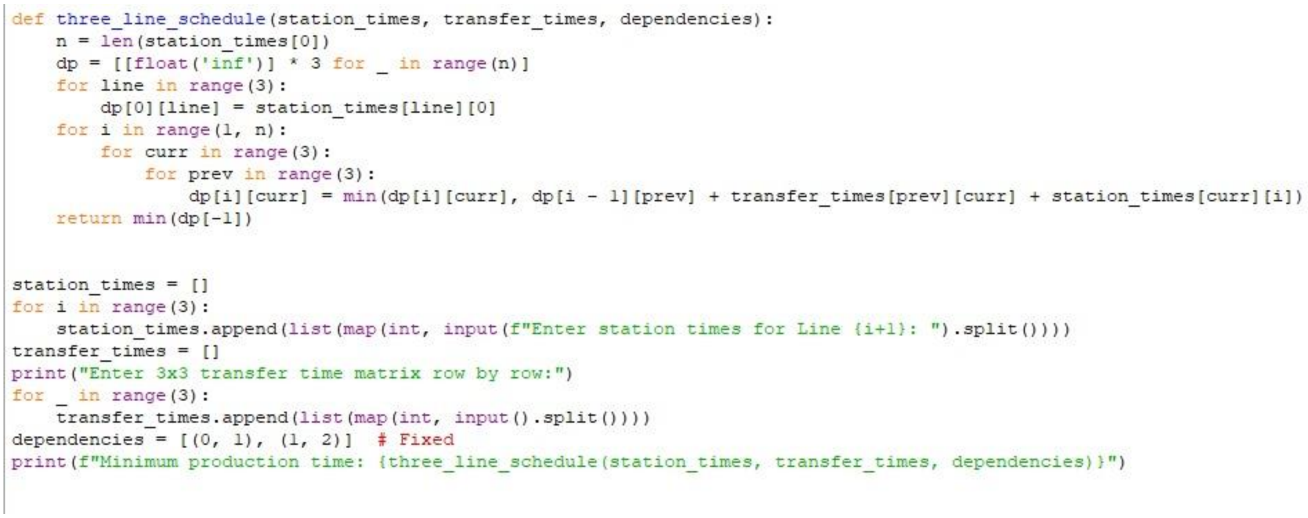
AIM

To implement a dynamic programming solution in Python that minimizes the total production time across three assembly lines, considering station times, transfer times, and task dependencies.

ALGORITHM

1. Let a[i][j] be the processing time at station *j* on line *i* (i ∈ {0,1,2}).
2. Let t[i][j][k] be the transfer time from line *i* to line *k* after station *j*.
3. Let e[i] be the entry time for line *i*, and x[i] be the exit time for line *i*.
4. Initialize T[i][j] to store the minimum time to reach station *j* on line *i*.
5. For each station *j* from 1 to n-1, compute T[i][j] as the minimum of staying on the same line or transferring from another line.
6. Return the minimum of T[i][n-1] + x[i] for all lines.

PROGRAM



```
def three_line_schedule(station_times, transfer_times, dependencies):
    n = len(station_times[0])
    dp = [[float('inf')] * 3 for _ in range(n)]
    for line in range(3):
        dp[0][line] = station_times[line][0]
    for i in range(1, n):
        for curr in range(3):
            for prev in range(3):
                dp[i][curr] = min(dp[i][curr], dp[i - 1][prev] + transfer_times[prev][curr] + station_times[curr][i])
    return min(dp[-1])


station_times = []
for i in range(3):
    station_times.append(list(map(int, input(f"Enter station times for Line {i+1}: ").split())))
transfer_times = []
print("Enter 3x3 transfer time matrix row by row:")
for _ in range(3):
    transfer_times.append(list(map(int, input().split())))
dependencies = [(0, 1), (1, 2)]  # Fixed
print(f"Minimum production time: {three_line_schedule(station_times, transfer_times, dependencies)}")
```

Input:

Enter station times for Line 1: 4 5 3 2

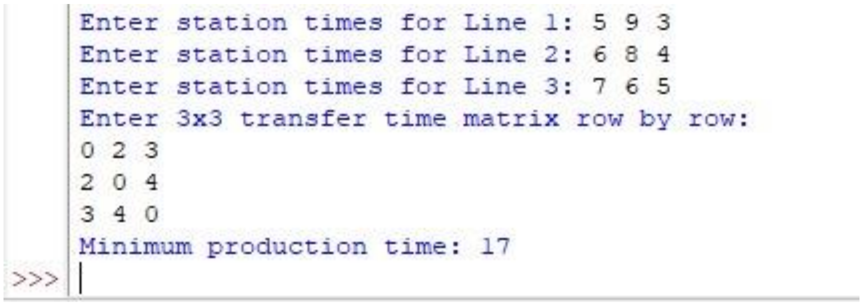
Enter station times for Line 2: 2 10 1 4

Enter station times for Line 3: 3 6 2 5

Enter 3x3 transfer time matrix row by row :

0 2 3
2 0 4
3 4 0

Output:



```
Enter station times for Line 1: 5 9 3
Enter station times for Line 2: 6 8 4
Enter station times for Line 3: 7 6 5
Enter 3x3 transfer time matrix row by row:
0 2 3
2 0 4
3 4 0
Minimum production time: 17
>>> |
```

RESULT:

Thus the program is successfully executed and the output is verified.

PERFORMANCE ANALYSIS:

- Time Complexity: $O(n \times m^2)$, where *n* is the number of stations and *m* is the number of lines
- Space Complexity: $O(n \times m)$ for storing intermediate results