TOPIC 5.4 : JOB SCHEDULING FOR MAXIMUM PROFIT

Problem Statement  
We have n jobs, where every job is scheduled to be done from startTime[i] to endTime[i], obtaining a profit of profit[i].  
You are given the startTime, endTime and profit arrays, return the maximum profit you can take such that there are no two jobs in the subset with overlapping time range.  
If you choose a job that ends at time X you will be able to start another job that starts at time X.

Example 1  
Input: startTime = [1,2,3,3], endTime = [3,4,5,6], profit = [50,10,40,70]  
Output: 120  
Explanation: The subset chosen is the first and fourth job.  
Time range [1-3] + [3-6], profit = 120 = 50 + 70.

Example 2  
Input: startTime = [1,2,3,4,6], endTime = [3,5,10,6,9], profit = [20,20,100,70,60]  
Output: 150  
Explanation: The subset chosen is the first, fourth and fifth job. Profit = 150 = 20 + 70 + 60.

Aim  
To write a program that selects non-overlapping jobs in order to maximize the total profit.

Algorithm

1. Start
2. Combine jobs as tuples of (startTime, endTime, profit)
3. Sort jobs based on endTime
4. Use dynamic programming with binary search:
   * For each job, either take it and add its profit with the best profit of a compatible previous job, or skip it
5. Return the maximum profit obtained
6. Stop

Input and Output  
A screenshot of a computer

AI-generated content may be incorrect.

Result  
The program successfully calculates the maximum profit from jobs scheduled without overlapping time.

Performance Analysis  
Time Complexity: O(n log n) due to sorting and binary search  
Space Complexity: O(n) for dynamic programming storage