Editorial - Quarantine Centers

Input data and calculate the amount of money that can be raised each flight at the same time.

Use a 2d vector to store the data of each flight in the format of {arrival_date, quarantine_finishing_date, amount_of_money}

Sort the flight data in the non descending order of the quarantine_finishing_date of each flight.

Dynamic programming is used to solve this problem.

An array is used to store the maximum amount of money that can be raised by considering up to ***i*** th flight(inclusive) in the ***i*** th index of the array

Initially amount of money from **0th** flight is stored in the index **0**

1. Each flight from **1** to ***n-1***, is considered in a loop. In each iteration,
    a. Take the amount of money from ***i*** th flight (take as **'amount'**)
    b. Find the latest prior flight with a 'quarantine_finishing_date' (take as ***j***) that doesn't conflict with the 'arrival_date' of the ***i*** th flight.
    c. If there is a flight ***j*** (There can be situation with no non-conflicting flights)* then add the value in index ***j*** of array (which is the max possible amount of money raised by considering up to ***j*** th flight) to the **'amount'**
    d. Then get the maximum amount by comparing **maximum amount of index *i-1*** and **'amount'** and assign in to the ***i*** th index of the array
    e. In the end of loop max amount of money that can be raised by considering all flights will be in the last index of the array