

# Report

## LPA - Programming problem #2

Passevite

### Algorithm description

The intent of this assignment is to, given information about a number of events  $N$ , create a feasible schedule for these events that maximizes the profit created. Being feasible means that each event finishes by its deadline, and that no two events overlap in the schedule. Many times, it isn't possible to materialize all events, and in this case we want to make a schedule in which we choose the events that are possible to schedule given the conditions and that will maximize our profit.

To figure out the solution to this problem, we started out by sorting the array of events according to their deadlines in a non decreasing order.

Then, we fill out a matrix  $profit\_matrix(i, t)$  that gives information about a feasible schedule, in which only events from 1 to  $i$  are scheduled, and all scheduled events finish by time  $t$ . The profit value of the optimal schedule is stored in  $profit\_matrix(N, d)$ .

We start out by filling the first row, from  $profit\_matrix(0, 0)$  to  $profit\_matrix(0, d)$ , with zeros (a schedule that has no scheduled events has no profit). Consider  $t$  to be the time tick by which all events are finished, and  $d_i$  to be the deadline of event  $i$ . Then, for each event  $i$  in the  $profit\_matrix$ , we will compute its possible start time, which is the difference between  $\min\{t, d_i\}$  and the duration of the event. If this value is negative, it means the event can't be scheduled, and we copy the value directly above the slot we're watching in the matrix. If the value for the starting time of the event is non negative, it is possible to schedule this event. However, we have to choose between not scheduling it (the profit for this event will be the same of the slot directly above in the  $profit\_matrix$ ) or scheduling it (the profit at the end of this event will be the sum of the event's profit and the profit possible at the time our event started).

---

Schedule a number of given events in order to maximize the profit using a DP approach

---

```
1: procedure FILL_PROFIT_MATRIX
2:   Initialize the first line of  $profit\_matrix$  with 0's
3:   for each column  $i$  in the  $profit\_matrix$  do                                 $\triangleright \mathcal{O}(n)$ 
4:     for each line  $t$  in the  $profit\_matrix$  do                                 $\triangleright \mathcal{O}(D)$ 
5:       Calculate possible starting time for this event
6:       if possible starting time is invalid ( $<0$ ) then
7:          $profit\_matrix[i][t] \leftarrow profit\_matrix[i-1][t]$                  $\triangleright$  Copy the value directly above
8:       else
9:          $profit\_matrix[i][t] \leftarrow \text{max value when choosing between scheduling or not that event}$ 
```

---

In order to validate the optimal solution computed by our approach to a given set of events, we have defined a recursive procedure that prints the scheduled events in order. The course of thought behind this procedure is that if the profit of the row above in the current time tick is the same as the value we're considering, it means we have not chosen the current event for our solution. If else, it means we have chosen the current event.

---

Printing the optimal solution to stdout

---

```

1: procedure WRITE_RESULTS_TO_STDOUT( $i, t$ )
2:   if there are no events scheduled then
3:     return
4:   if profit of the row above in the current time tick is the same then WRITE_RESULTS_TO_STDOUT( $i - 1, t$ )
5:   else
6:     Calculate the current event start time
7:     WRITE_RESULTS_TO_STDOUT( $i - 1, event\_start\_time$ )
8:     Print information of this event

```

---

## Data structures

To store each event, we use *struct event*, which saves the deadline, duration and profit of a given event. The events to schedule are all stored in an array of events, *events*. The *profit\_matrix* is the structure that will support our dynamic programming approach. We also use some support global variables, such as *total\_count\_events*, which keeps the biggest deadline of all events in a given set.

## Correctness

Let's assume that the event's list is sorted by non decreasing order of time of the event's deadlines, and consider a schedule  $A$  that contains at least one event.  $i$  is the ID of the last scheduled event, which finishes by time  $t$  (this also means that all events are finished by  $t$ , as the events are ordered by their deadlines). Then, there is also another feasible schedule  $A'$ , that schedules exactly the same events as  $A$  and that  $A'(i) = \min(t, d_i) - t_i$ , where  $d_i$  and  $t_i$  are the deadline and duration of the last event, respectively.

The value in the position  $A(i, t)$  is the maximum possible profit where only events from 1 to  $i$  are scheduled, and all the events are finished by time  $t$ .

$A(0, t)$  is always 0 (without any scheduled events the profit is zero). Now let's consider that  $t' = \min(t, d_i) - t_i$ , this represents the possible time tick to schedule the event  $i$ , so that it finishes before its deadline. If  $t' < 0$ , it means it's not possible to schedule this event. If  $t' \geq 0$  it means we can schedule this event. The profit value at this moment would be the maximum value that it's possible to obtain when choosing between scheduling event  $i$  or not scheduling it.

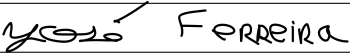
## Algorithm Analysis

As is noted in the pseudo-code for the procedure of scheduling the events, the time-complexity for this algorithm is  $\mathcal{O}(n * d)$ , but we must also take into consideration, that in this case, if the biggest deadline of a given set  $d$  is equal or bigger than  $n$ , then our approach will be quadratic  $\mathcal{O}(n^2)$ .

## References

- [1] Jeff Erickson (2018) Algorithms, 0th Edition (pre-publication draft)
- [2] Thomas H. Cormen, Charles E. Leiserson, Ronald Rivest, Clifford Stein (2001) Introduction to Algorithms, The MIT press, 2nd Edition
- [3] Rodney R. Howell (2009) Algorithms: A Top-Down Approach, Ninth Draft

## Team members

Name	Student ID	Signature
João Soares	2009113061	
José Ferreira	2014192844	 FERREIRA
Madalena Santos	2016226726	Madalena Costa Gonçalves Carlos Santos