# A Pseudocode for the second question of the project of COMP 6651

Aria Adibi

**Algorithm** Pseudocode for the Question 2 of the project

Sort the intervals based on their end time; then "rename" them as $I_1, \ldots, I_n$.

Add $I_0 = (-\infty, end_0)$ as sentinel interval such that $end_0$ is strictly smaller that start point of all intervals. *//The purpose of the sentinel is to not check whether all intervals are considered or not in the following loops. Also note that in the graph this interval is a isolated vertex, therefore it will not change the answer.*

Define $d[i][j]$ where $0 \leq i \leq n$ and $0 \leq j \leq 1$ as an `integer` array and initialize it to all 0. *//Note that integer overflow issue is ignored here; that was the reason for the modulo.*

*//Initialization*

$d[0][0] = 0;$ $\quad d[0][1] = 1$

**for** $i = 1$ to $n$

    //$d[i][0]$

    *//Partition the possibilities with respect to the latest possible interval that is in the Dominating Set*

    $j = i - 1;$ $\quad$ latestStartSoFar= $I_i.start$

    **while** $I_j.end \geq$ latestStartSoFar

        $d[i][0] = d[i][0] + d[j][1]$

        latestStartSoFar= max$\{$latestStartSoFar, $I_j.start\}$

        $j = j - 1$

    //$d[i][1]$

    *//First, consider the cases in which at least one of the neighbors of $I_i$ which is more restrictive than $I_i$ is a member of Dominating Set, i.e. $I_{neighbor}.start < I_i.start$*

    *//Note that the covered intervals by $I_i$, i.e. $I_{neighbor}.start \geq I_i.start$, add no additional restrictions, therefore it is as if they are not there and just the final answer is $\times 2$ for each, member of the Dominating Set or not.*

    $j = i - 1;$ $\quad$ numCoveredIntervals= 0

    **while** $I_j.end \geq I_i.start$

        **if** $I_j.start \geq I_i.start$

            numCoveredIntervals= numCoveredIntervals $+1$

        **else**

            $d[i][1] = d[i][1] + 2^{\text{numCoveredIntervals}} \times d[j][1]$

        $j = j - 1$

    *//Second and final case is when no restrictive neighbor of $I_i$ is in the Dominating Set. Note that now j is pointing to the first interval that is not a neighbor of $I_i$*

    $d[i][1] = d[i][1] + 2^{\text{numCoveredIntervals}} \times (d[j][0] + d[j][1])$

print $d[n][0] + d[n][1]$