Algorithm 1 'In-fitter' Algorithm

```
procedure INFITTER($box_1$, $box_2$)
    $ParameterOne$: $box_1$ Dimensions of the bigger box (Array) $[L, W, H]$
    $ParameterTwo$: $box_1$ Dimensions of the smaller box (Array) $[L, W, H]$
    $Output$: The most amount of $box_2$'s that will fit into $box_1$

    $perms \leftarrow [1,2,3;1,3,2;2,1,3;2,3,1;3,1,2;3,2,1]$        ▷ permutations of box orientation
    $amountFit \leftarrow [6]$
    for $i = 1$ to 6 do
        $boxL \leftarrow box_2[perms[i][1]]$        ▷ get current permutation configuration
        $boxW \leftarrow box_2[perms[i][2]]$
        $boxH \leftarrow box_2[perms[i][3]]$

        $amountL \leftarrow floor(box_1[1]/boxL)$
        $amountW \leftarrow floor(box_1[2]/boxW)$
        $amountH \leftarrow floor(box_1[3]/boxH)$
        $amountFit[i] \leftarrow (amountL * amountW * amountH)$
    end for
    Return $findLargest(amountFit)$
end procedure
```