

Question 1

Zeal Liang

Z5325156

First define an array A , and place the blocks of the i th stack in the i th position of A .

Then we traverse this array with i . If i is not the last index of A , then each time we move $A[i] - i - 1$ blocks from $A[i]$ to $A[i+1]$. This would end up moving the blocks in array A into $[1, 2, 3 \dots n]$.

At the same time, when i is greater than or equal to 1, we check whether the blocks in $A[i]$ is less than or equal to the blocks in $A[i-1]$ before each movement. If it is, return "NO", if not, continue the loop.

If no "NO" is returned by the end of the loop, then it is possible to make the sizes of the stacks strictly increasing.

The time complexity of this algorithm is $O(n)$ because we only perform linear time operations on each stack in array A .