COMP3121-Ass2-Q4

z5302513, Kihwan Baek

July 2021

Firstly, we can assume that if the n stacks of blocks are strictly increasing, it should be $h_1 < h_2 < ... < h_n$. And also, we can say after we sort the blocks in strictly increasing order, the number of blocks on n stacks should be $h_2 >= h_1 + 1$, $h_3 >= h_1 + 2$, ... $h_n >= h_1 + (n-1)$. Furthermore, we know that we can move the blocks from stack i to stack i + 1. So, we can conclude that after we move the blocks from h_1 to h_n to make these in strictly increasing, the sum of the numbers of blocks from h_1 to h_n should be larger than or equal to the sum of the numbers from 1 to n-1, which is $n(n-1)/2$.

Therefore, we initially make S as sum and set it zero. And then, we add the number of block on first block to S and check if S is larger than or equal to $1(1-1)/2 = 0$. If it is larger than or equal to the number, we can make the sizes of stacks strictly increasing. Otherwise, we can say that it is impossible to make the those strictly increasing. Finally, we implement this algorithm for all the stacks and it takes O(n) time. If it finishes without any stop, we can determine that such movements exists.