

9101 Assignment 2

Haojin Guo

z5216214

Q4.

4. You are given n stacks of blocks. The i^{th} stack contains $h_i > 0$ identical blocks. You are also able to move for any $i \leq n - 1$ any number of blocks from stack i to stack $i + 1$. Design an algorithm to find out in $O(n)$ time whether it is possible to make the sizes of stacks strictly increasing. (For example, 1,2,3,4 are strictly increasing but 1,2,2,3 are not). The input for your algorithm is an array A of length n such that $A[i] = h_i$. Note that you are not asked to actually move the blocks, only to determine if such movements exists or not.

Solution,

Considering that the number of blocks in the 1st position can be 0, the strictly increasing array of the stacks of blocks must at least satisfy the condition of 0,1,2,3, ...

Therefore,

the number of blocks at position $i+1$ should be greater than or equal to the value of

$$\frac{(i-1) \cdot i}{2}.$$

In this process,

1) it is need to traverse the rang (1, n) ----- (Cost = $O(n)$)

2) calculate the sum of the number of previous $i-1$ stacks (called A) and compare it with

the current value $B = \frac{(i-1) \cdot i}{2}$. (the cost is $O(1)$ each time)

If $A < B$, the n stacks of blocks cannot turn into the strictly increasing stacks. Otherwise, continue to check the next position of blocks.

The total cost for this algorithm above is $O(n)$.

Python_Django [~/Documents/Python

Python_Django > Q4.py

q2.py x Q4.py x

```
1 def check_strict_increase(blocks):
2     output = True
3     temp = blocks[0]
4     for i in range(1, len(blocks)):
5         # the sum from 0 to i-1
6         min_require = (i * (i - 1)) / 2
7
8         if temp < min_require:
9             output = False
10
11         temp += blocks[i]
12
13     return output
14
15
16 blocks = [1, 2, 1, 16]
17 result = check_strict_increase(blocks)
18
19 print(result)
20
```

Run: Q4 x

↑ /opt/anaconda3/python.app/Contents/MacOS/python /Users/guohaojin/Documents/Python_Dja
True

↓

⏏ Process finished with exit code 0