

[< Back](#)
[Facebook | Stack Stabilization | Interview](#)


Anonymous User

November 27, 2021 5:52 AM 367 VIEWS

0

There's a stack of N inflatable discs, with the i th disc from the top having an initial radius of R_i inches.



inches.

The stack is considered unstable if it includes at least one disc whose radius is larger than or equal to the radius of the disc directly below it. In other words, for the stack to be stable, each disc must have a strictly smaller radius than that of the disc directly below it. As long as the stack is unstable, you can repeatedly choose any disc of your choice and deflate it down to a strictly smaller radius. The new radius must be a positive integer number of inches. Determine the minimum number of discs which need to be deflated in order to make the stack stable, if it is already stable, return -1 instead.

Constraints

$1 \leq N \leq 50$

$1 \leq R_i \leq 1000$

$\leq 1,000,000,000$

Sample test case #1

$N = 5$

$R = [2, 5, 3, 6, 5]$

Expected Return Value = 3

Sample test case #2

$N = 3$

$R = [100, 100, 100]$

Expected Return Value = 2

Sample test case #3

$N = 4$

$R = [6, 5, 4, 3]$

Expected Return Value = -1

Need help my current code is exceeding the time limit. Please help me optimize it.

```
def getMinimumDeflatedDiscCount(N: int, R: List[int]) -> int:
    count = 0
    visited = []
    unstable = True
    while(unstable):
        for i in range(N-1, 0, -1):
            if R[i] <= R[i-1]:
                R[i-1] -= 1
                if i-1 not in visited:
                    visited.append(i-1)
```