Daily Coding Problem   Blog

Daily Coding Problem #265

Problem

This problem was asked by Atlassian.

MegaCorp wants to give bonuses to its employees based on how many lines of codes they have written. They would like to give the smallest positive amount to each worker consistent with the constraint that if a developer has written more lines of code than their neighbor, they should receive more money.

Given an array representing a line of seats of employees at MegaCorp, determine how much each one should get paid.

For example, given `[10, 40, 200, 1000, 60, 30]`, you should return `[1, 2, 3, 4, 2, 1]`.

Solution

To start, note that each employee must receive at least 1, so we can initialize our bonus array with ones.

We can proceed by making a pass from left to right over our lines of code array. If a developer has written more code than their left neighbor, we can set their bonus to be one greater than that of the neighbor. After this pass, our array above would look like `[1, 2, 3, 4, 1, 1]`.

To complete this algorithm, we would like to perform a similar operation going right to left, but we cannot simply reverse the direction and repeat the previous step. In the array above, for example, this would set the bonus of the `1000`-line developer back to 3, which

would no longer be greater than their left neighbor. We can correct for this by ensuring

that the developer's bonus is the maximum of its current value and one more than that of the right neighbor.

```python
def assign_bonuses(lines):
    bonuses = [1 for _ in range(len(lines))]

    for i in range(1, len(lines)):
        if lines[i] > lines[i - 1]:
            bonuses[i] = bonuses[i - 1] + 1

    for i in range(len(lines) - 2, -1, -1):
        if lines[i] > lines[i + 1]:
            bonuses[i] = max(bonuses[i], bonuses[i + 1] + 1)

    return bonuses
```

This algorithm consists of two passes over our input, and uses an additional array for the bonuses, so it will run in $O(N)$ time and space.