Why is this imperative style faster than this functional style?

Asked 9 years, 4 months ago Modified 9 years, 4 months ago Viewed 87 times

1

Say we have an array that we want to find his Equilibrium Index, Why is imperative style faster than functional style, and what is the logic behind this algorithm (imperative's)?

functional style:

```
def eq_indices(list)
  list.each_index.select do |i|
    list[0...i].inject(0, :+) == list[i+1..-1].inject(0, :+)
  end
end
```

imperative style:

```
def eq_indices(list)
  left, right = 0, list.inject(0, :+)
  equilibrium_indices = []

  list.each_with_index do |val, i|
    right -= val
    equilibrium_indices << i if right == left
    left += val
  end

  equilibrium_indices
end
```

ruby

Share

Improve this question

Follow

edited Aug 25, 2015 at 7:56

sawa

168k ● 49 ● 285 ● 396

asked Aug 25, 2015 at 7:36

Glubi

123 ● 5

not sure, but the comment seems related to the other form of imperative style... – tomsoft Aug 25, 2015 at 7:49

3 Answers

Sorted by: Highest score (default)

3

Because in functional style, the sum of the left and right sides is calculated from scratch for each potential equilibrium, whereas in imperative style, the sum is calculated once, and only single subtraction and addition are performed for each potential equilibrium.

Share
Improve this answer
Follow

edited Aug 25, 2015 at 7:47

answered Aug 25, 2015 at 7:45
sawa
168k ● 49 ● 285 ● 396

+1 The functional version is a Shlemiel the Painter version. – Cristian Lupascu Aug 25, 2015 at 7:47

Thank you. I get why it is faster now, do you mind explaining the logic behind the imperative style solution? I'm truly baffled by it, specifically, the reason of decreasing `right` value per each iteration, before the equalization. – Glubi Aug 25, 2015 at 7:58

@Glubi Subtracting `val` from `right` corresponds to the fact that the equilibrium point at index `i` with value `val` is not to be included in the sum. – sawa Aug 25, 2015 at 8:02

2

In this particular instance, the difference comes from the fact that the functional style solution has `O(n^2)`, while the imperative has `O(2n) = O(n)`.

Aka the functional solution makes one loop for each index and inside that loop there is another loop to determine the sums. In the imperative solution, there is one loop to assign the sum to `right` and one to find indices, but they are not nested.

Share Improve this answer Follow

answered Aug 25, 2015 at 7:47
ndnenkov
36.1k ● 9 ● 78 ● 106

1

You already have your answer, but since nobody has spelled this out clearly yet, only mentioned it implicitly, I want to make it explicit:

> Why is imperative style faster than functional style

It isn't. The two versions don't implement the same algorithm. The performance difference is due to the difference in algorithms, not the difference in styles.

Share Improve this answer Follow

answered Aug 25, 2015 at 8:25

Jörg W Mittag
369k ● 79 ● 453 ● 661