Question 4 - Recursive Sum of Elements in an Array

Jaan Tollander de Balsch

2018-09-30

Devise a divide-and-conquer (recursive) algorithm for adding up all numbers in the array $A[1...n]$. Your algorithm should divide the array A into two equal subarrays.

1) Describe your algorithm in a clear pseudocode.
2) Analyze the running time of your algorithm. Prove your answer.

The pseudocode implementation in Julia of the algorithm.

```
function sum(A)
  n = lenght(A)
  if n == 1
    # If the array A has only one element return the element.
    return A[1] # 1-based indexing
  else
    # Otherwise divide the array into equal sized subarrays and
    # then add the sums of the subarrays together.
    mid = div(n, 2)  # Integer division
    return sum(A[1:mid]) + sum(A[(mid+1):end])
  end
end
```

In the running time analysis we assume that operations `length` and `[]` (indexing and slicing) on arrays are constant time operations. We also assume that addition `+` and integer division `div` of two numbers are constant time operations. The recurrence relation therefore is of form

$$T(n) = 2T(n/2) + O(1).$$

By using the *master theorem* (1), the running time of the algorithm is

$$T(n) = O(n).$$

As we can see, the divide and conquer algorithm for the summation is no better than direct summation term by term.

References

1. Cormen TH, Leiserson CE, Rivest RL, Stein C. *Introduction to algorithms*. MIT press (2009).