

Programming Assignment 2

Joni Vrapı

10/17/2022

Statement of Integrity: I, Joni Vrapı, attempted to answer each question honestly and to the best of my abilities. I cited any, and all, help that I received in completing this assignment.

Problem (a). Both of these algorithms extend the implementation of *ImprovedRecursionTree* in order to compute the depth of the recursion tree in linear time. At each depth level of the tree, signified by an iteration of each function's main for-loop, the recursive cost, and non-recursive cost at a node is computed. The total non-recursive cost at each depth is then computed by multiplying across all nodes at that depth. A slight quirk in both algorithms, in order to handle constant time non-recursive costs, is that when they receive zero as an input for both c and d , they assume that the non-recursive cost is a constant time operation.

DIVIDE-AND-CONQUER(a, b, c, d)

```
1  if  $a < 1 \vee b \leq 1 \vee c < 0 \vee d < 0 \vee \{b, c, d\} \notin \mathbb{Q}$ 
2      throw Error
3  var a, b, c, d, n
4  for i in [1...4]
5      if  $i == 0$ 
6          recursiveCost = n
7          numNodes = 1
8          if  $c == 0 \wedge d == 0$ 
9              nonRecursiveCost = 1
10         else
11             nonRecursiveCost =  $cn^d$ 
12         printResults()
13     else
14         recursiveCost =  $\frac{n}{b^i}$ 
15         numNodes =  $a^i$ 
16         if  $c == 0 \wedge d == 0$ 
17             nonRecursiveCost = 1
18         else
19             nonRecursiveCost =  $c(recursiveCost^d)$ 
20         printResults()
21
22
```

```

CHIP-AND-BE-CONQUERED( $a, b, c, d$ )
1  if  $a < 1 \vee b \leq 1 \vee c < 0 \vee d < 0 \vee \{c, d\} \notin \mathbb{Q} \vee \{a, b\} \notin \mathbb{Z}$ 
2      throw Error
3  var  $a, b, c, d, n$ 
4  for  $i$  in  $[1..4]$ 
5      if  $i == 0$ 
6          recursiveCost =  $n$ 
7          numNodes = 1
8          if  $c == 0 \wedge d == 0$ 
9              nonRecursiveCost = 1
10         else
11             nonRecursiveCost =  $cn^d$ 
12         printResults()
13     else
14         recursiveCost =  $n - b * i$ 
15         numNodes =  $a^i$ 
16         if  $c == 0 \wedge d == 0$ 
17             nonRecursiveCost = 1
18         else
19             nonRecursiveCost =  $c(n - b * i)$ 
20         printResults()
21
22

```

Problem (b). The worst case asymptotic running time for both of these recursion tree generator functions is $O(n)$. This is due to the fact that there is a single loop in each implementation, iterating exactly 4 times. Each mathematical calculation within the loop can be executed in $O(1)$ time. $3 * O(1) * O(n) = O(n)$.

Problem (c). The asymptotic running time of both algorithms is exactly the same, and remains so regardless of how you vary either the inputs or even the depth (number of iterations of the loop).

Analysis. After running the *tests.py* file as described in the README.md file, you will see that the produced output in terms of iterations stays exactly at 4. If we were to also vary the depth of the tree that is generated (against your instructions), we would see that this algorithm remains linear, even though I severely varied the Rationals, as well as other inputs into both functions. This was expected as per the asymptotic analysis performed above.