|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Kingdom of Saudi Arabia**  **Ministry of Education**  **University of Jeddah**  **College of Science and Computer Engineering** |  | **المملكة العربية السعودية**  **وزارة التعليم**  **جامعة جدّة**  **كلية علوم وهندسة الحاسب** | |  |  |

**CCCS 314 – Design and Analysis of Algorithms**

**LAB 5**

**Topics:**

1. **Divide and Conquer**

**Total Marks: 2**

**Student Name: Bassam Alghamdi**

**Student ID: 2141362**

**Marks:**

|  |  |  |  |
| --- | --- | --- | --- |
| Exercises | 1 | 2 | Total |
| Allocated | 1 | 1 | 2 |
| Obtained |  |  |  |
| **CLO, PLO** | 1.1, K1 | 2.1, S1 |  |

**CLO** **Marks:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | CLO1.1, K1 | CLO2.1, S1 | Total |
| Allocated | 1 | 1 | 2 |
| Obtained |  |  |  |

**Exercise 1:** **(Divide and Conquer: Binary tree traversal)**

Write pseudocode for one of the classic traversal algorithms (preorder, inorder, and postorder) for binary trees. Assuming that your algorithm is recursive, find the number of recursive calls made.

**Answer** :

function inorderTraversal(node):

if node is null:

return

inorderTraversal(node.left) // Recursive call on left subtree

visit(node) // Process current node

inorderTraversal(node.right) // Recursive call on right subtree

preorder(node):

if node is null:

return

# Process the current node

visit(node)

# Recursively traverse the left subtree

preorder(node.left)

# Recursively traverse the right subtree

preorder(node.right)

postorder(node):

if node is null:

return

# Recursively traverse the left subtree

postorder(node.left)

# Recursively traverse the right subtree

postorder(node.right)

# Process the current node

visit(node)

number of recursive calls made: depend on the tree height and how many nodes so it would be O(n) because they visit them all.

**Exercise 2:** **(Divide and Conquer: Big Integers Multiplications)**

1. **Compute 1201\*2430 by applying the divide-and-conquer algorithm outlined in the lecture.**

**Answer** :

Step 1: Break down the given integers into smaller parts.

1201 = 12 \* 100 + 1

2430 = 24 \* 100 + 30

Step 2: Recursively compute the products of these smaller parts.

A = 12 \* 24

B = 12 \* 30

C = 1 \* 24

Step 3: Combine the results of the recursive computations.

The final product, P, is given by:

P = (A \* 100^2) + (B \* 100) + C

Let's calculate each part:

A = 12 \* 24 = 288

B = 12 \* 30 = 360

C = 1 \* 24 = 24

Finally, combining the results:

P = (A \* 100^2) + (B \* 100) + C

= (288 \* 10000) + (360 \* 100) + 24

= 2880000 + 36000 + 24

= 2918430

Therefore, 1201 \* 2430 = 2918430.