An example of an inductive proof of algorithm correctness

1. An algorithm in pseudo-code
   This is a general scheme for recursive functions based on post-order traversal.
   ```
   int F(TNode* T_ptr)
   {
       if (the node exists)
       {
           L: Process the left sub-tree
           R: Process the right sub-tree
           V: Combine the information from
               the left sub-tree, the right sub-tree,
               and the current node and pass it
               to the level above
       }
   }
   ``
   TNode – a tree type node
   TNode* - a pointer of TNode type
   int – integer type

2. Tree traversal (example)
   Task: Add the numbers in data fields
   ![Tree Diagram](image)

3. The actual code, which adds numbers inside data fields.
   Variable T_ptr stores the address of a tree type node
   NULL is the value, which indicates, that there is no corresponding node
   T_ptr->data allows us to access the data field
   T_ptr->left_link allows us to access the address of the left child
   T_ptr->right_link allows us to access the address of the right child
int F(TNode* T_ptr)
{
    if(T_ptr!=NULL)
    {
        int left, right;
        left=F(T_ptr->left_link);
        right=F(T_ptr->right_link);
        return left+right+T_ptr->data;
    }
    else
    return 0;
}

4. An inductive proof of correctness of the algorithm

$P[k]$ - Recursive function $F$ computes the sum of data fields of any binary tree consisting of $k$ nodes.

$P[1]$ - Direct inspection shows, that if the tree consists of just one node, then the function returns the value of the data field. Both $T_{ptr}→left\_link$ and $T_{ptr}→right\_link$ are NULL and therefore values of variables $left$ and $right$ are 0. Function $F$ returns $left+right+T_{ptr}→data$, which in this case is $T_{ptr}→data$. We conclude that $P[1]$ holds.

$(\forall k \mid 1 \leq k \leq n : P[k]) \Rightarrow P[n+1]$ - Let us take any binary tree $T$ consisting of $n+1$ nodes. We need to show, that $F$ computes the sum of data fields of $T$. We know that $F$ computes the sum of data fields of any tree consisting of fewer than $n+1$ nodes. This is our inductive hypothesis. We observe that the left sub-tree $L$ and the right sub-tree $R$ of the root of $T$ both consist of fewer than $n+1$ nodes. Therefore the inductive hypothesis applies to them. This in turn implies, that in the first recursive call, where $T_{ptr}$ is the address of the root, variable $left$ contains the sum of data fields of $L$ and variable $right$ contains the sum of data fields of $R$. Therefore $left+right+T_{ptr}→data$ is the sum of data fields of $T$. We conclude that $P[n+1]$ holds.

```
T
  /\   /
 /   \ /
L      R
```

\[ \cdots \]