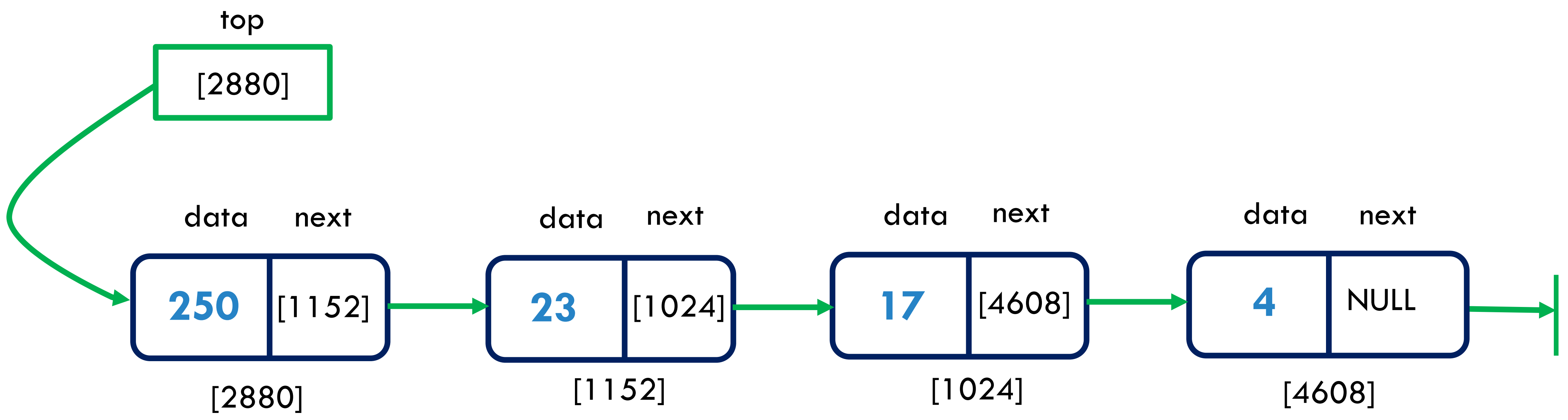
COMP 2611, Data Structures

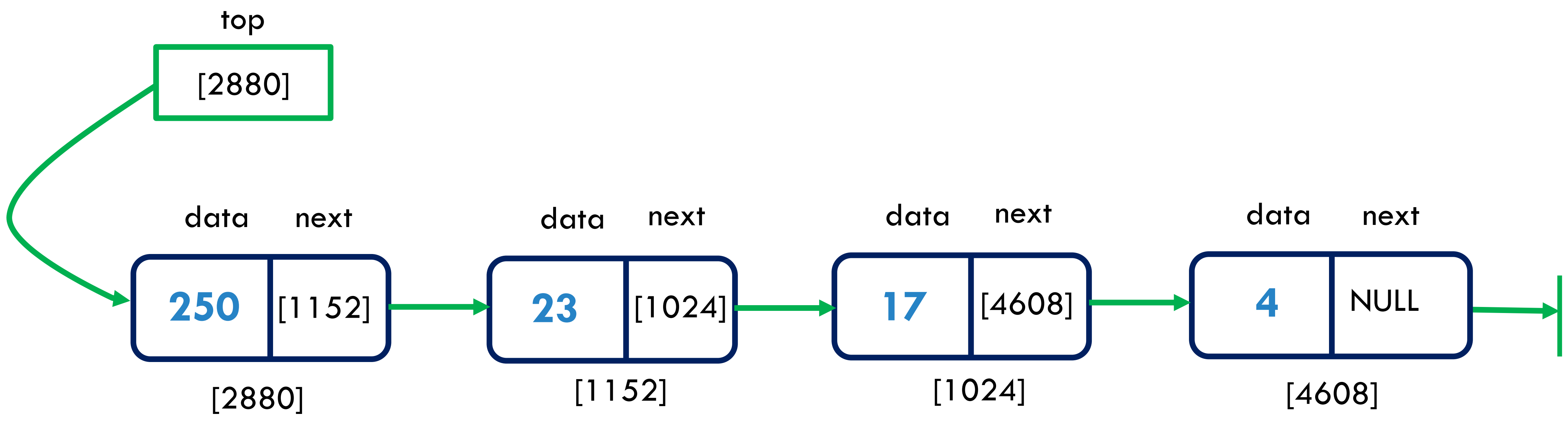
LECTURE 5: RECURSION WITH ARRAYS AND BINARY TREES

PRINTLISTREVERSE

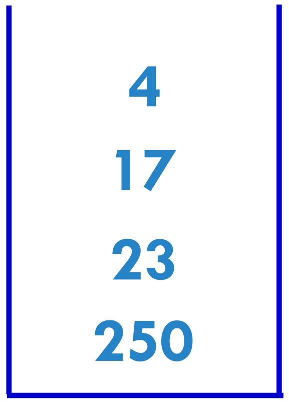


How to display elements in reverse order WITHOUT using recursion?

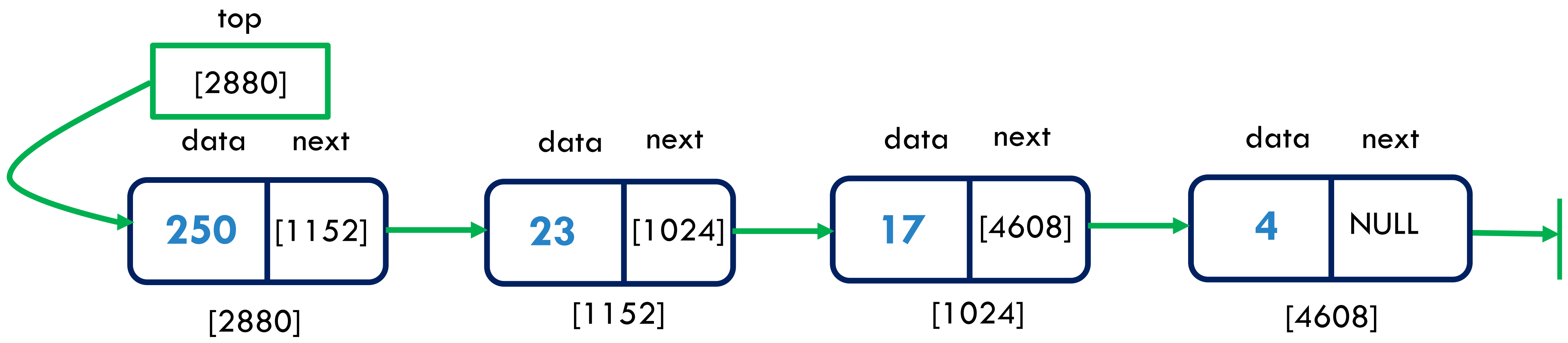
PRINTLISTREVERSE

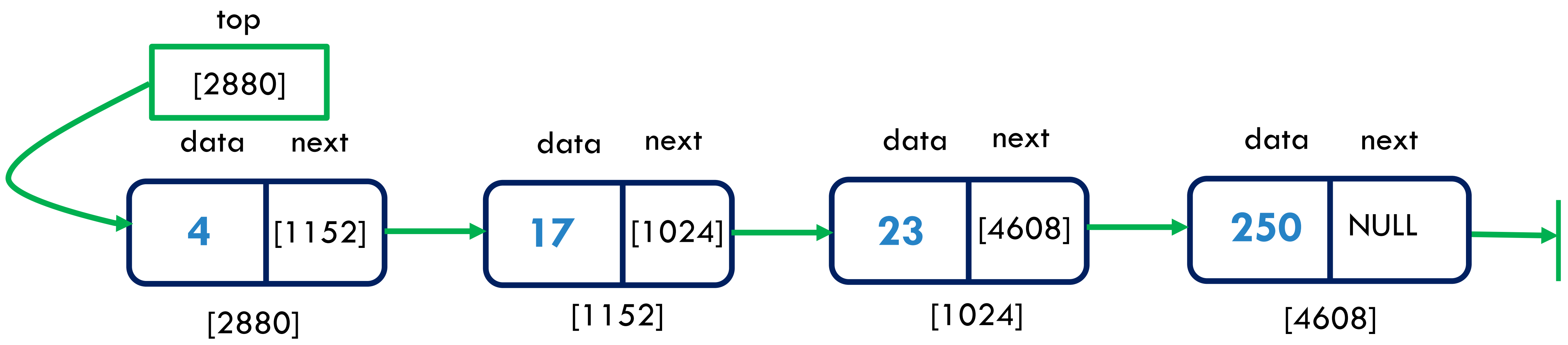


- Traverse the elements from the top
- Store the elements in a stack
- Pop the elements from the stack and print

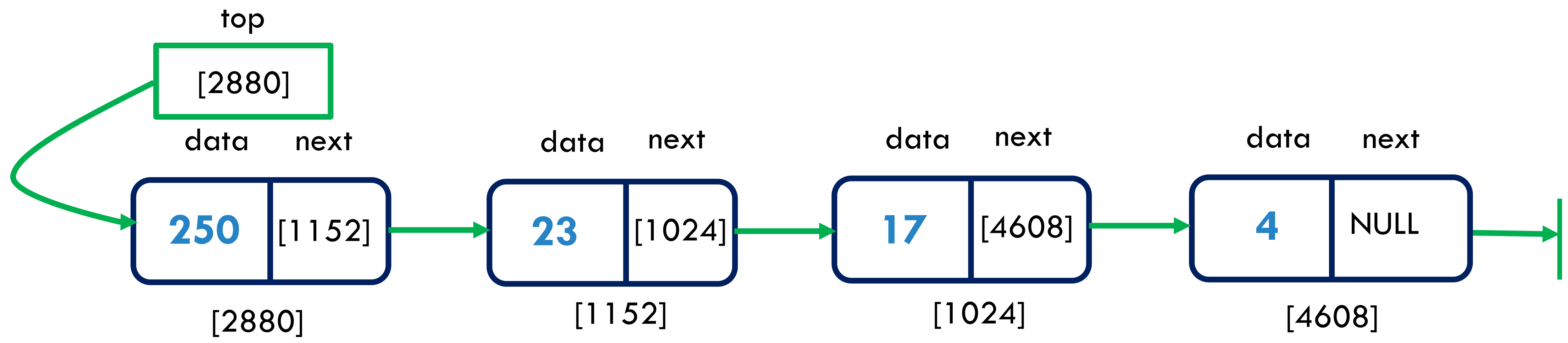


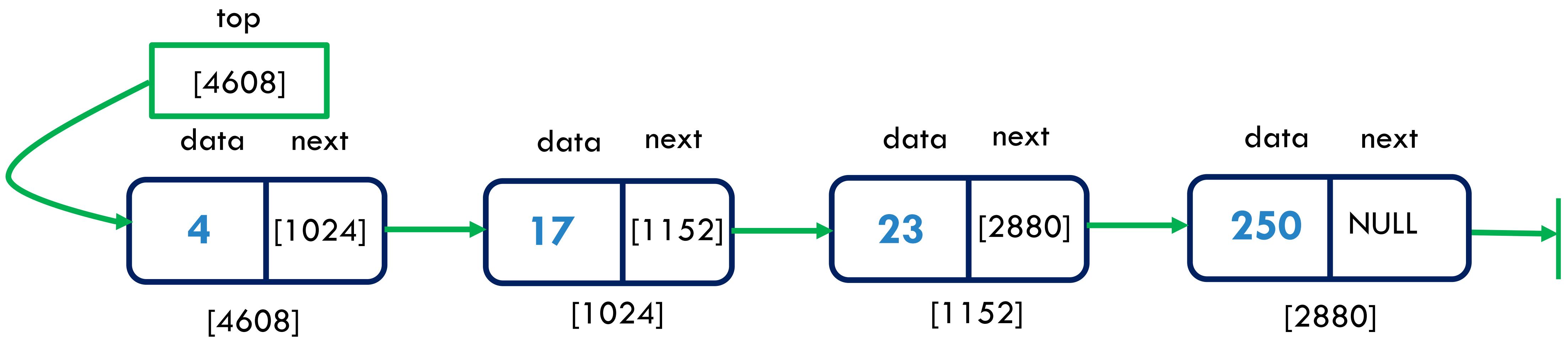
HOW TO REVERSE THE ELEMENTS IN THE LIST?





HOW TO REVERSE THE ELEMENTS IN THE LIST?

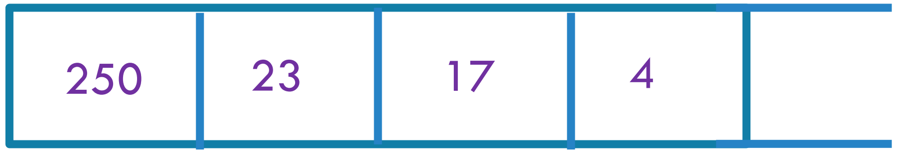




RECURSION WITH ARRAYS

```
void printArrayRec (int a[], int i, int n);
```

- *printArrayRec* displays all the elements of the array *a*, where *n* is the number of elements in *a*.
- Suppose that the array *a* has the following elements:

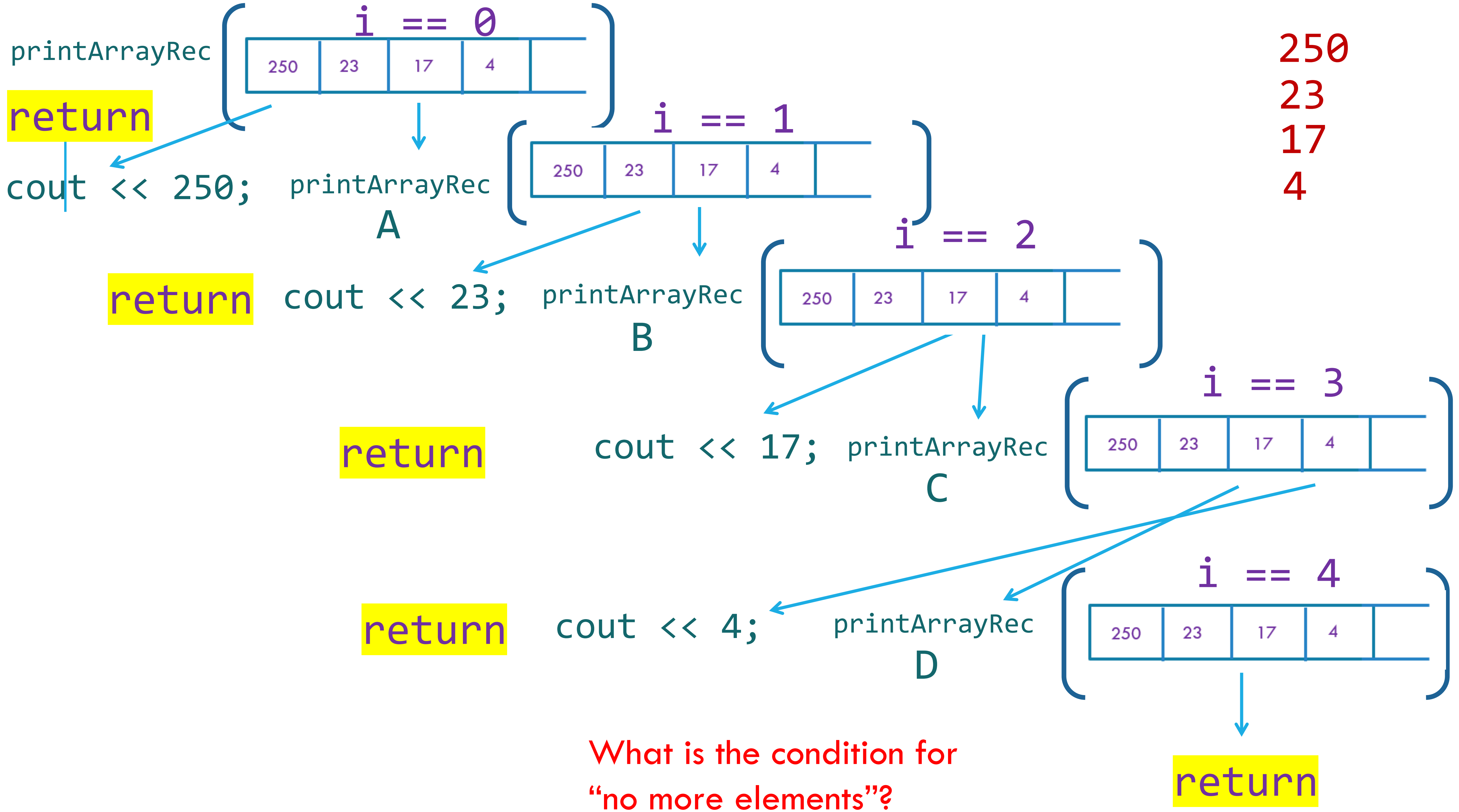


- *printArrayRec* will be called as follows:

```
printArrayRec (a, 0, 4);
```

RECURSION TREE FOR PRINTARRAYREC

- printArrayRec displays all the elements of an array on the monitor using recursion.



What is the condition for
"no more elements"?

CODE FOR PRINTARRAYREC

```
void printArray (int a[], int i, int n) {

  if (i >= n)
    return;

  cout << a[i] << endl;

  printArray (a, i+1, n);

}
```

CODE FOR PRINTARRAYREC

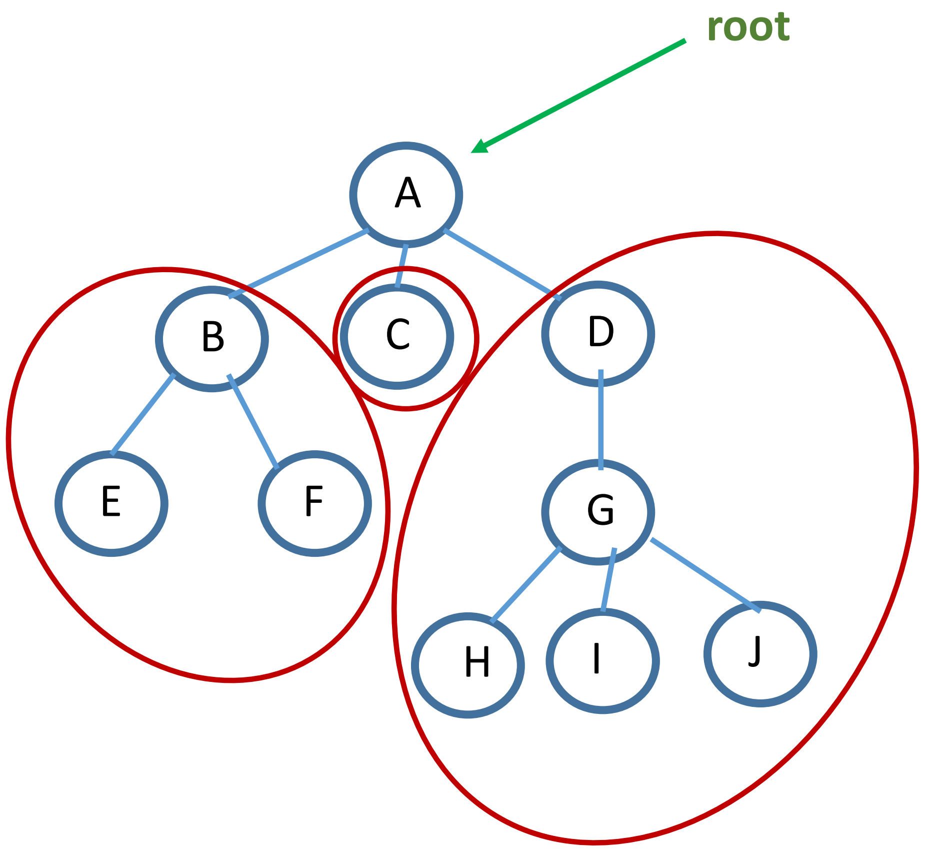
```
void printArray (int a[], int i, int n) {

  if (i < 0 || i >= n)
    return;

  cout << a[i] << endl;

  printArray (a, i+1, n);

}
```

What is a Tree?

- A woody perennial plant, typically having a single stem or trunk growing to a considerable height and bearing lateral branches at some distance from the ground:

What is a Tree?

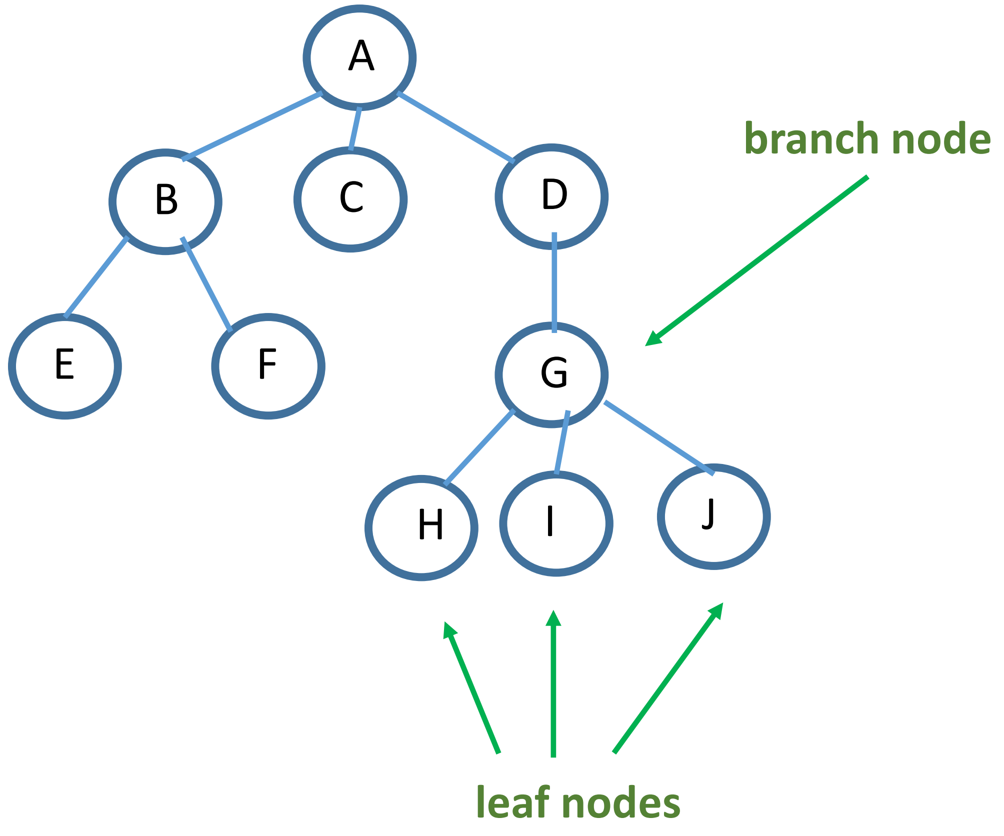
- A tree is a finite set of nodes such that:
  - There is one specially designated node called the *root* of the tree.
  - The remaining nodes are partitioned into m ≥ disjoints sets T_1, T_2, …, T_m, and each of these sets is a tree.



- The root of the given tree is A.
  - There are three subtrees rooted at A.
  - The *degree* of a node is the number of subtrees of the node.

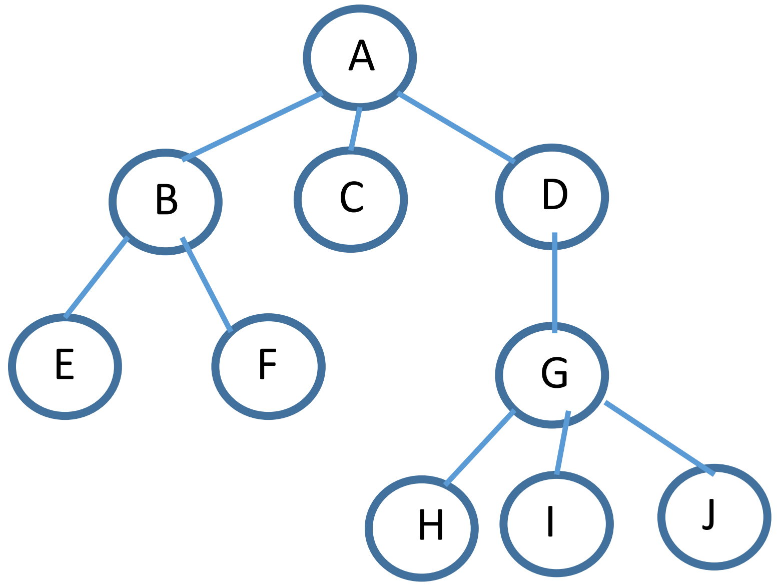
Tree Terminology

- The terms *parent*, *child*, and *sibling* are used to refer to the nodes of a tree.
- A node may have several children but only one parent (except for the root). The root is the only node that does not have a parent.
- *Sibling* nodes are child nodes of the same parent (e.g., *B*, *C*, *D*).
- A *terminal* node (or *leaf* is a node of degree 0). A *branch* node is a nonterminal node.



Tree Terminology

- The *moment* of a tree is the number of nodes in the tree.
- The *weight* of a tree is the number of leaves in the tree.
- The *level* (or *depth*) of a node is the number of branches that must be traversed on the path to the node from the root. The root has level 0.
- The *height* of a tree is the longest path from the root node to any leaf node in the tree.


A
B
C
D
E
F
G
H
I
J