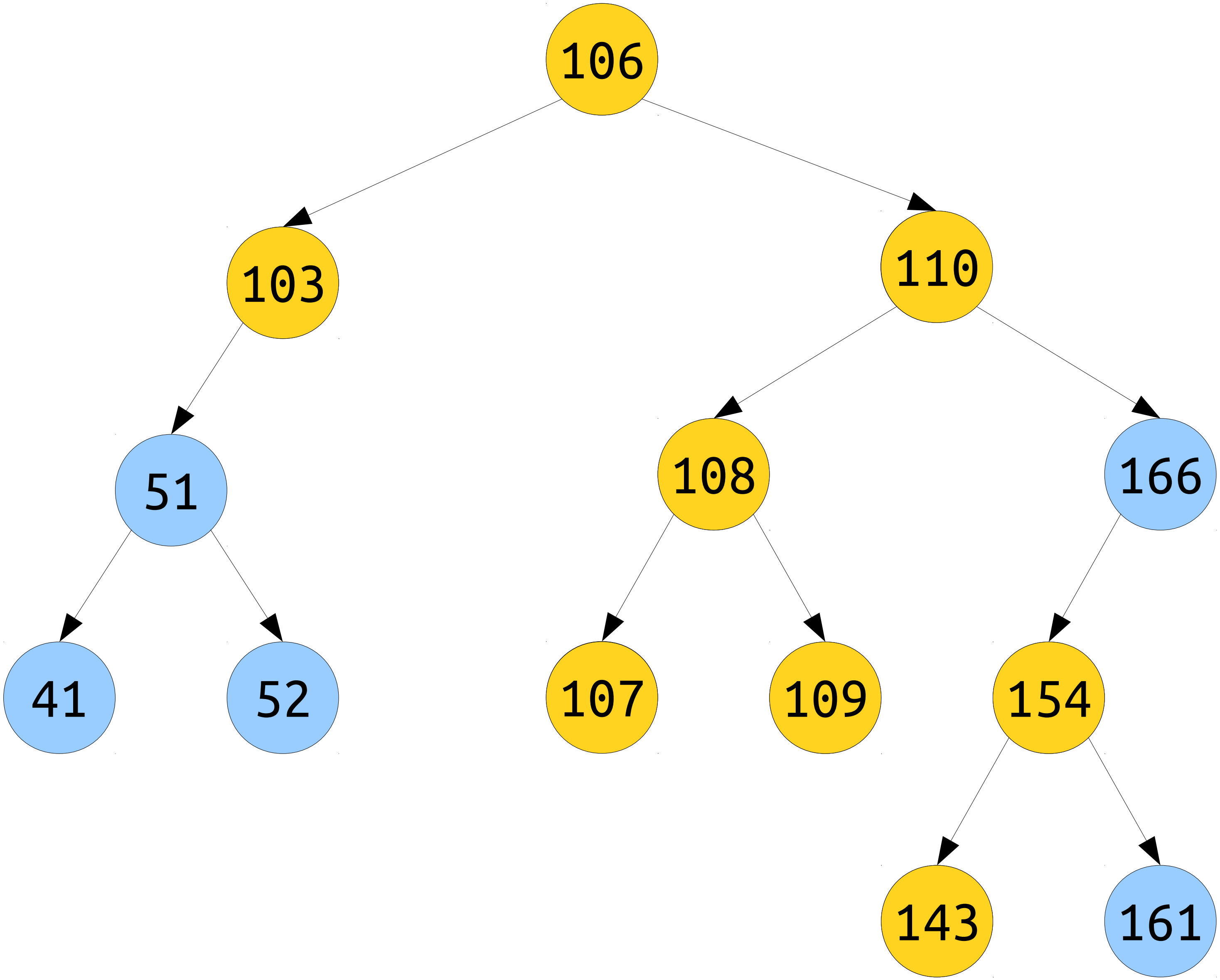
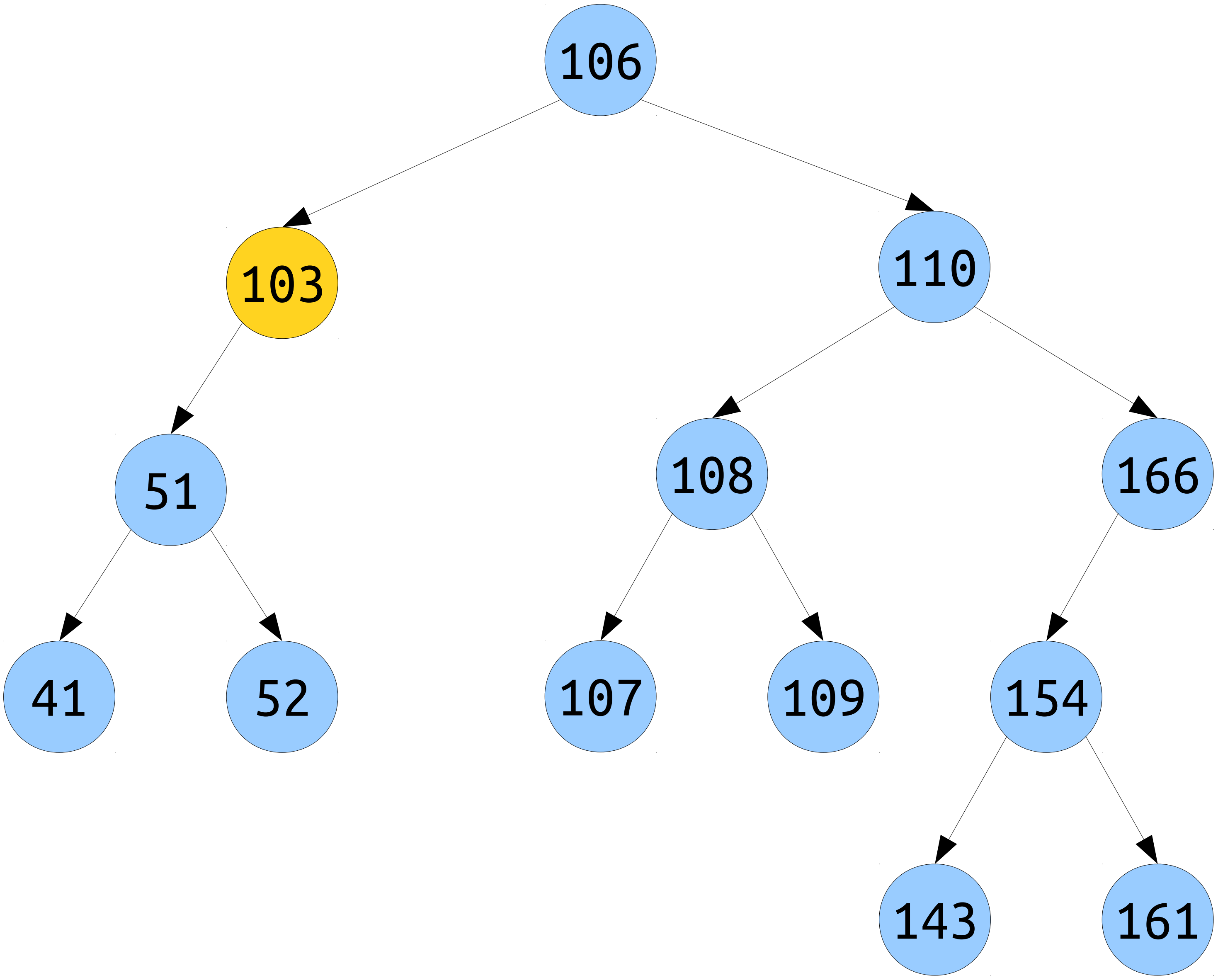
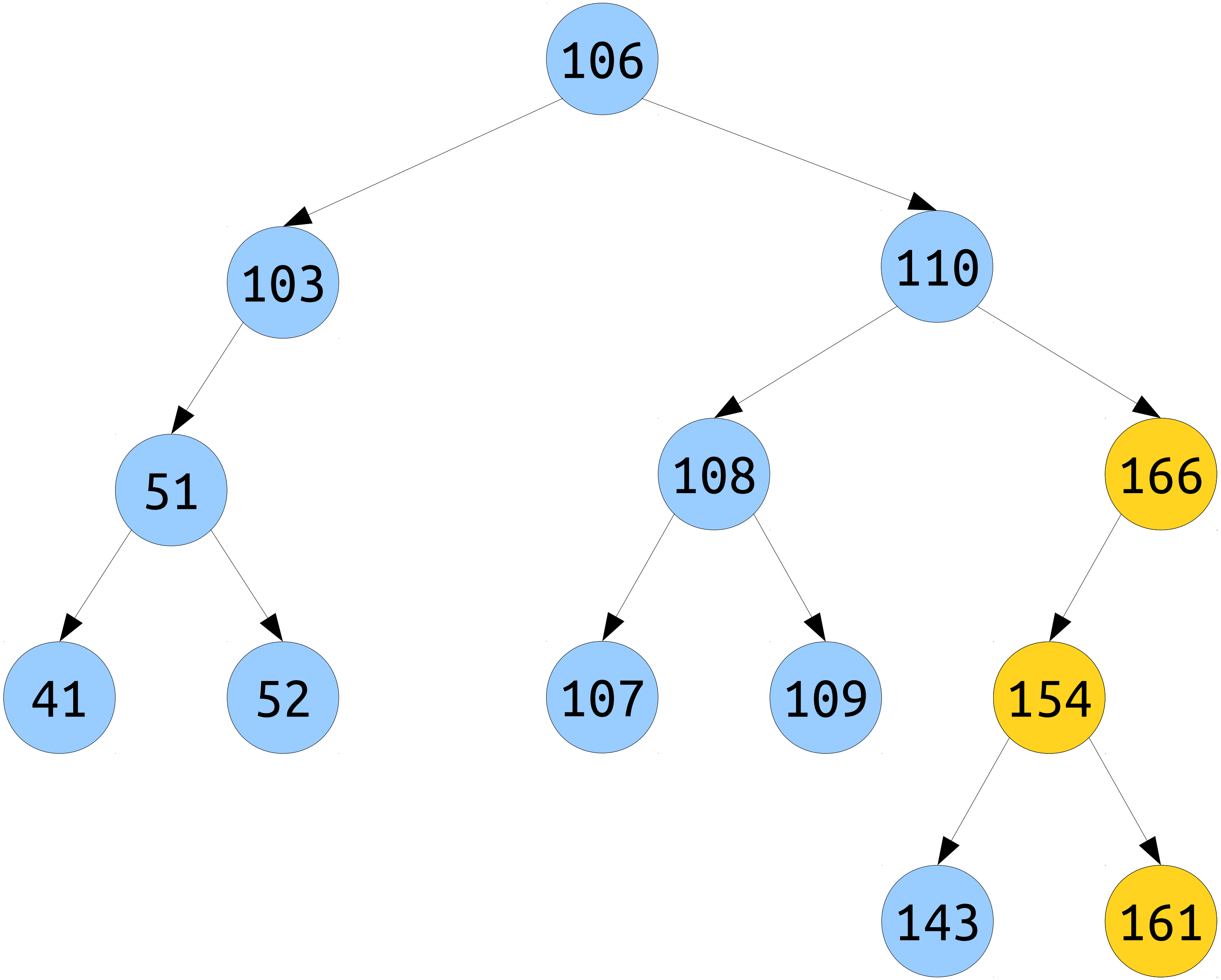
Range Searches



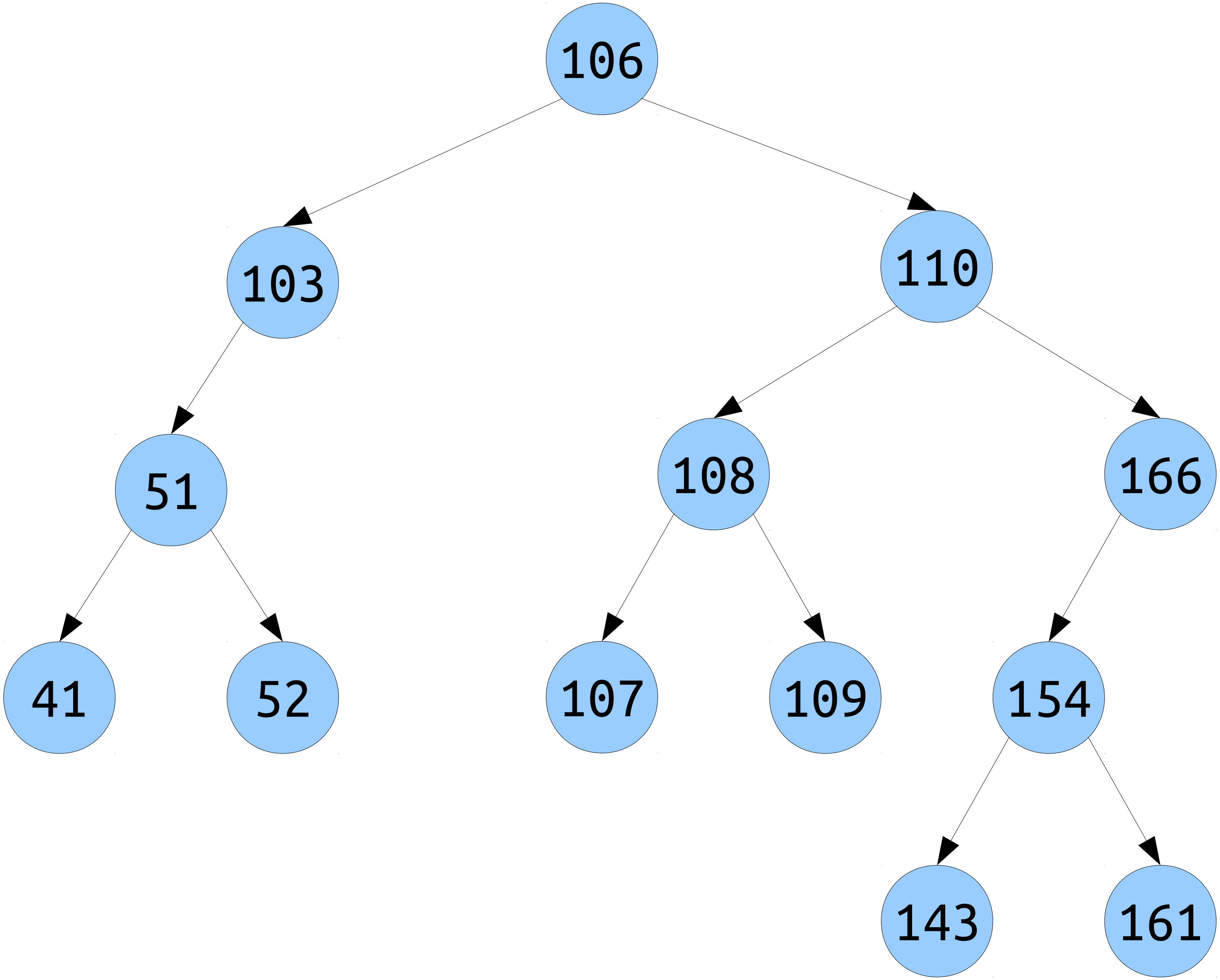
Find all elements in this tree in the range **[103, 154]**.



Find all elements in this tree in the range **[99, 105]**.



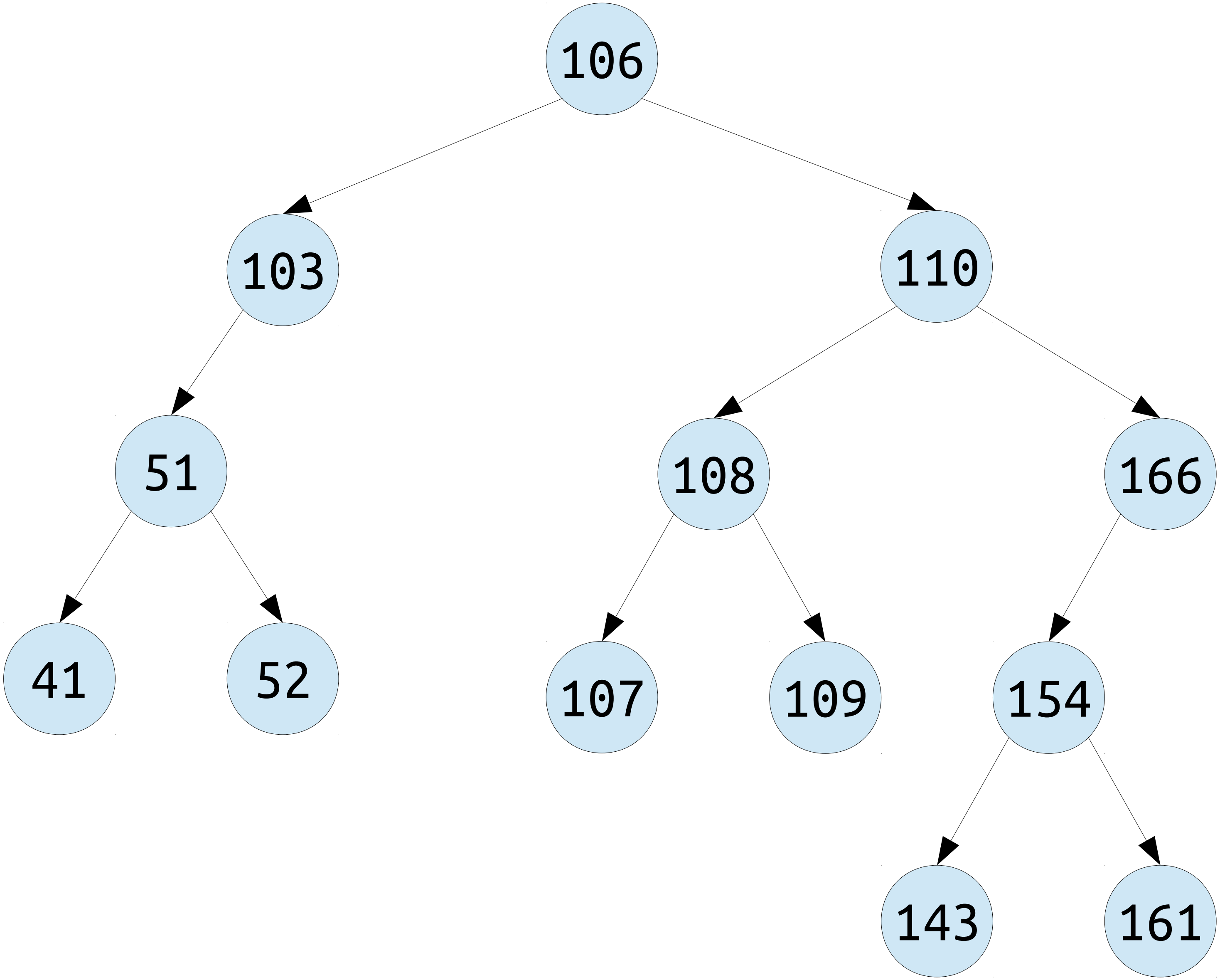
Find all elements in this tree in the range **[150, 170]**.



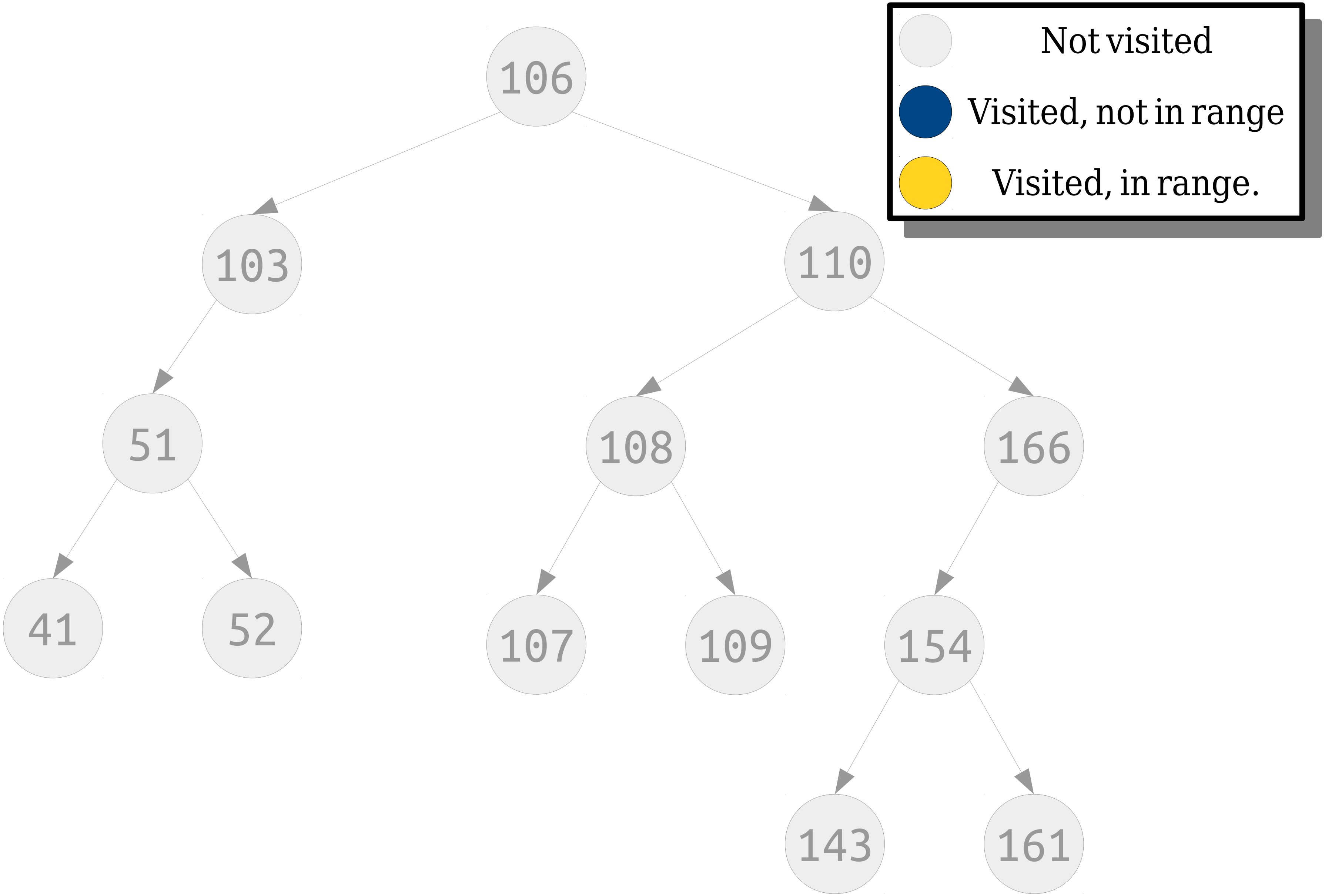
Find all elements in this tree in the range **[137, 138]**.

Range Searches

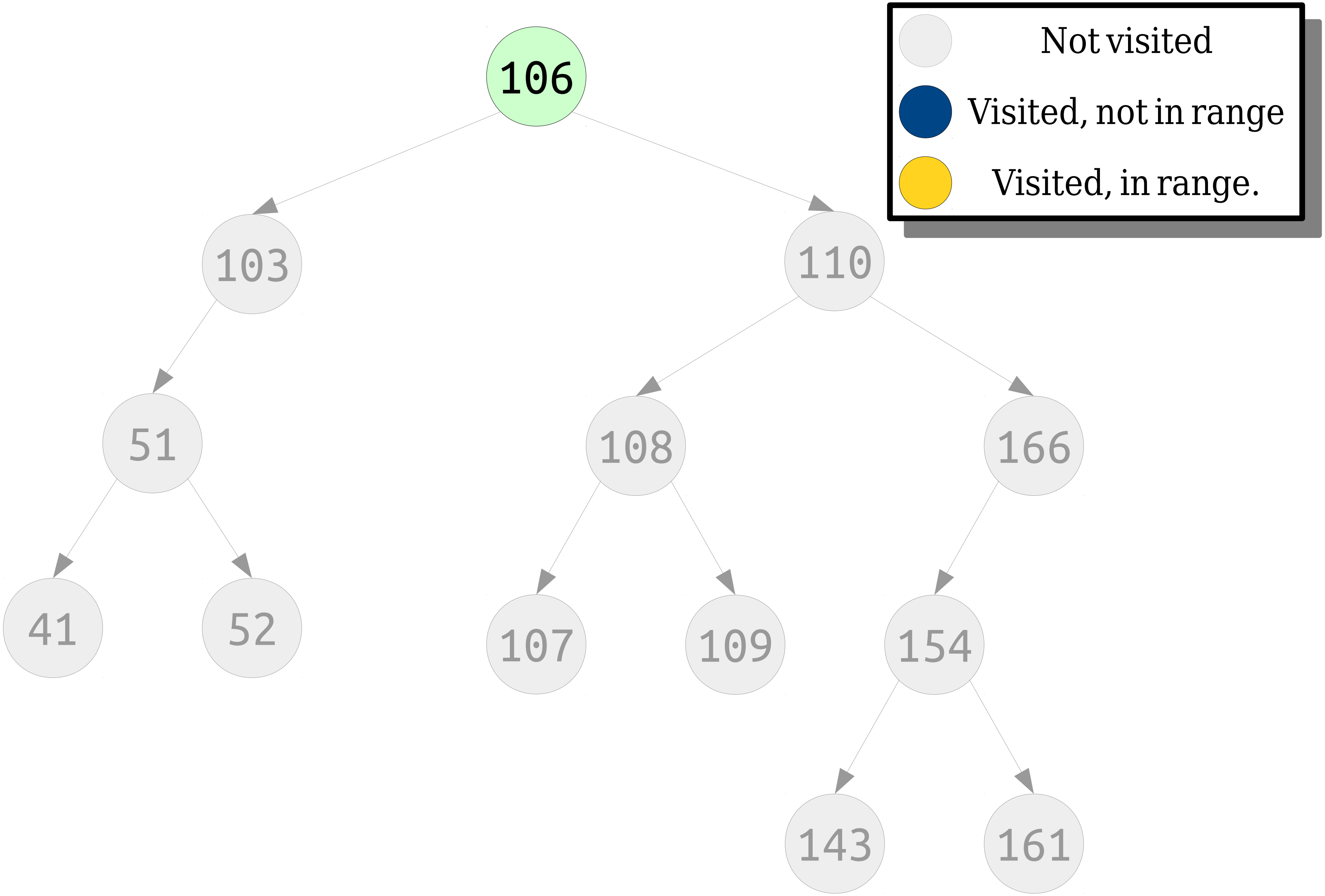
- We can use BSTs to do ***range searches***, in which we find all values in the BST within some range.
- For example:
  - If the values in the BST are dates, we can find all events that occurred within some time window.
  - If the values in the BST are number of diagnostic scans ordered, we can find all doctors who order a disproportionate number of scans.



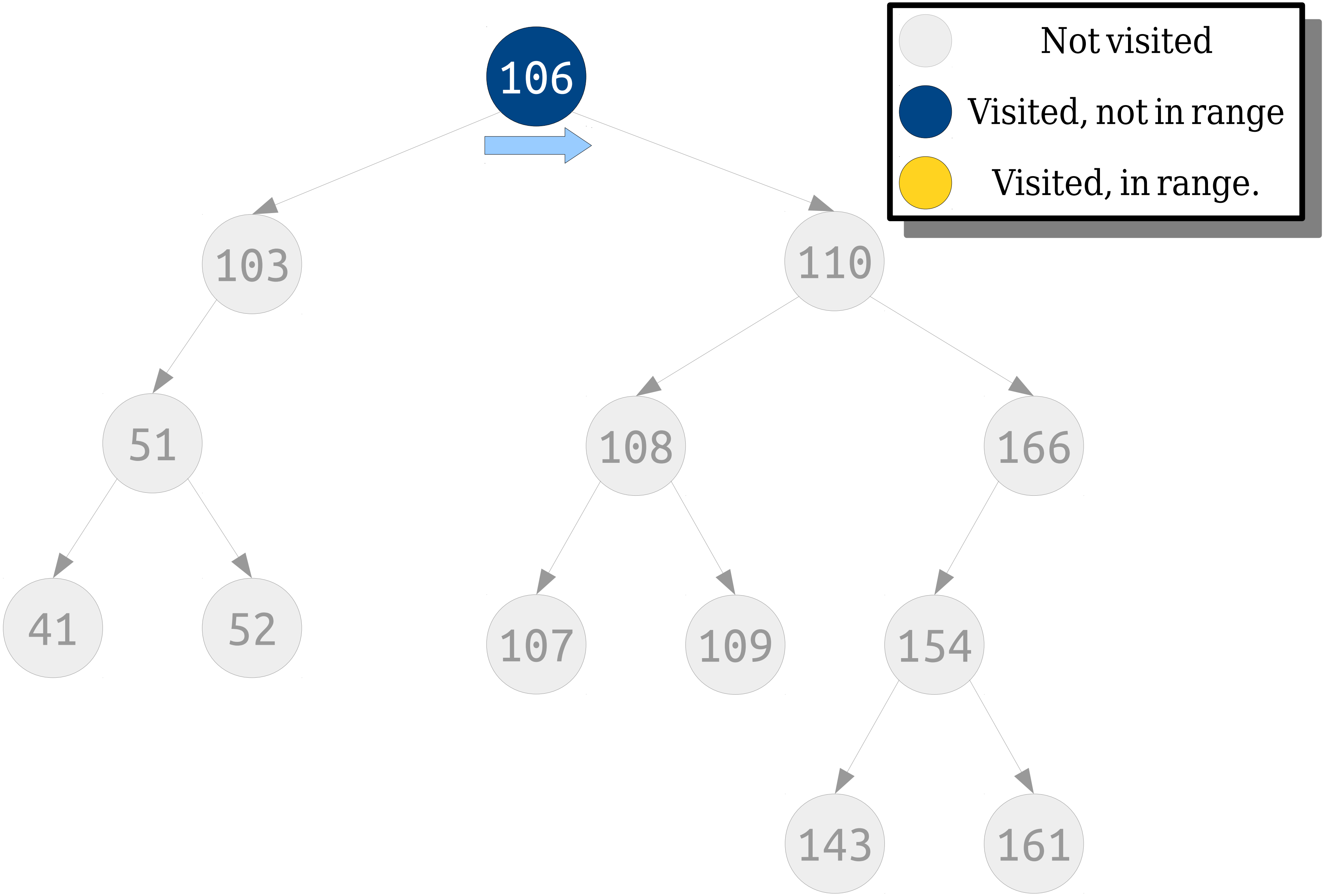
Find all elements in this tree in the range **[109, 163]**.



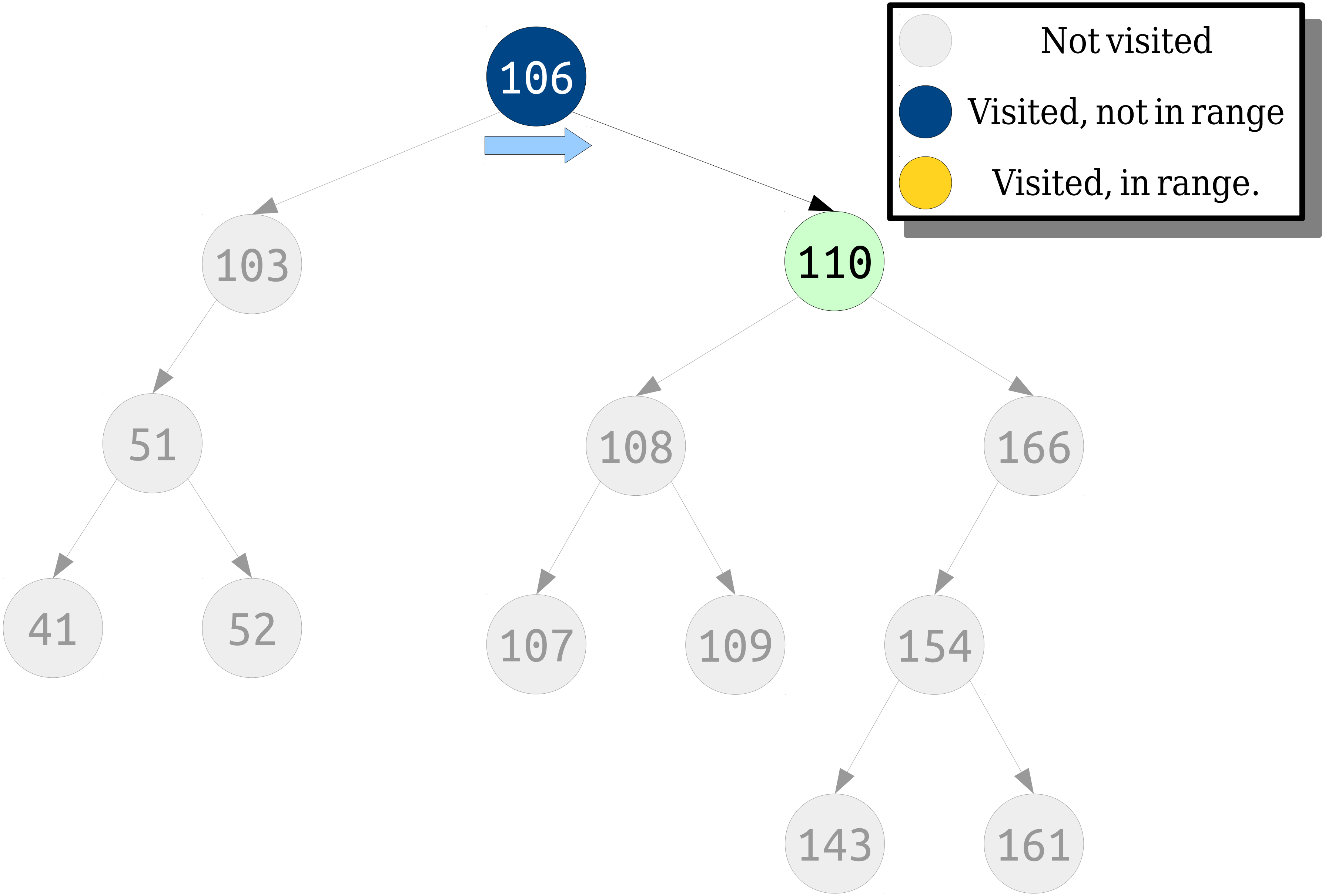
Find all elements in this tree in the range **[109, 163]**.



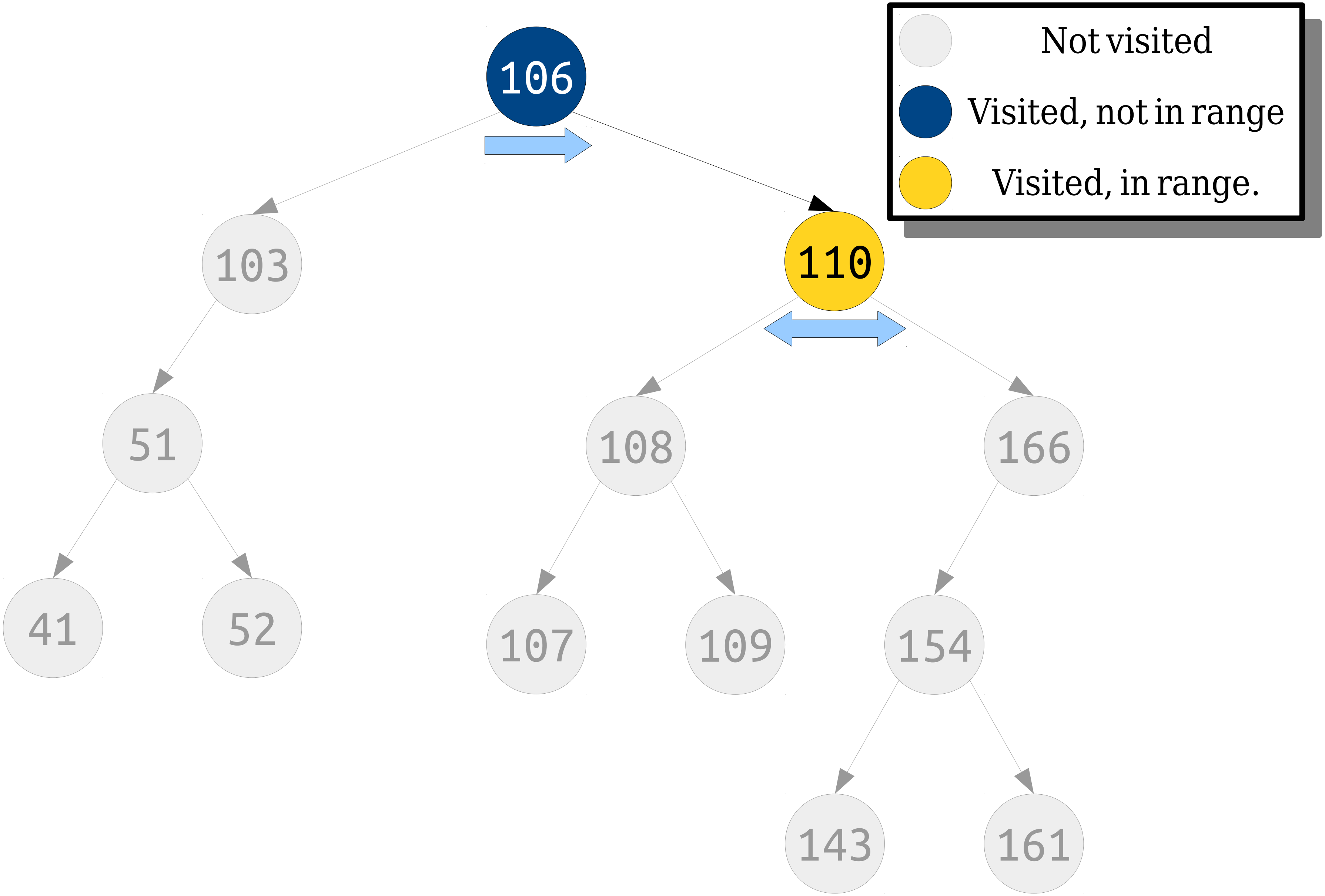
Find all elements in this tree in the range **[109, 163]**.



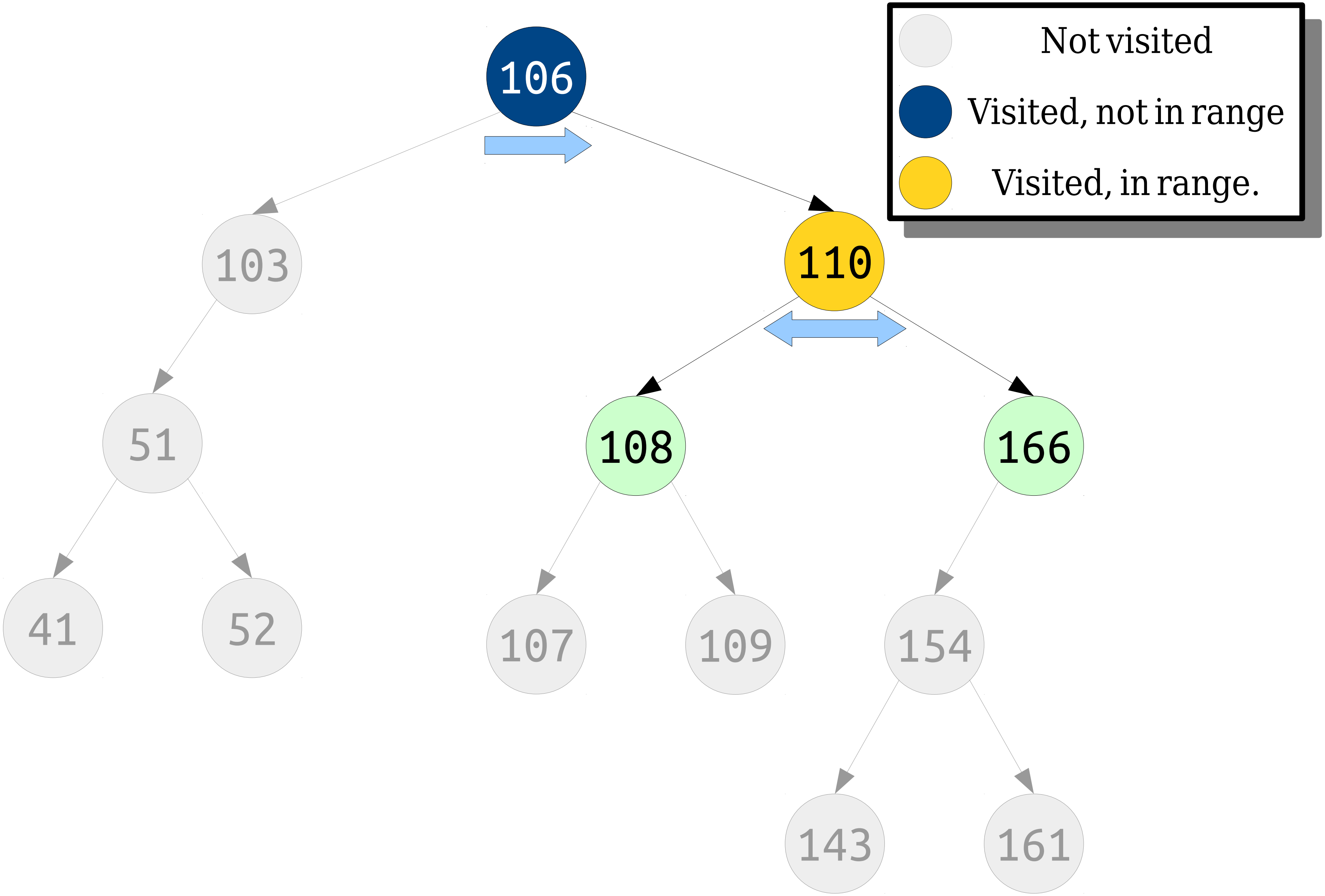
Find all elements in this tree in the range **[109, 163]**.



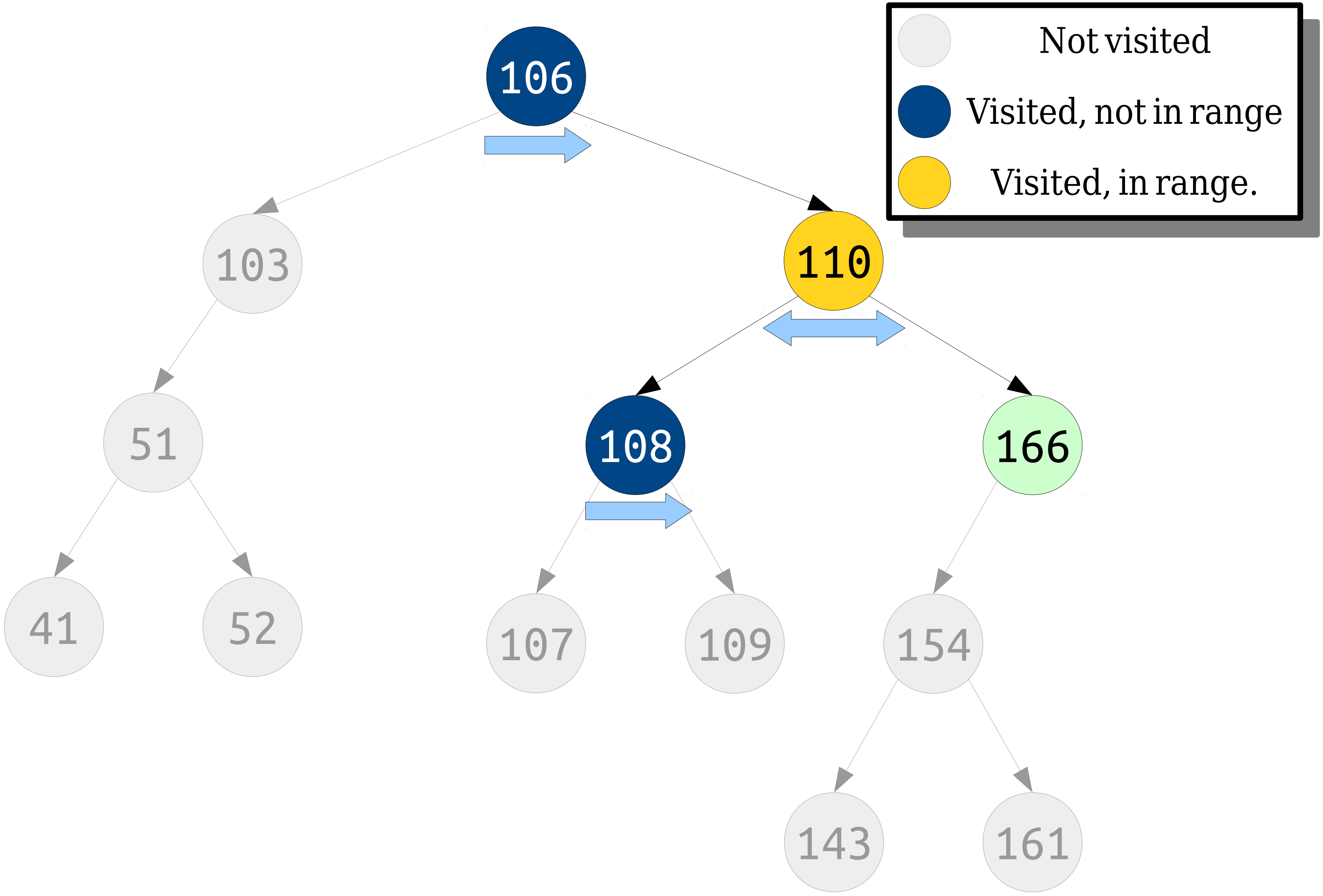
Find all elements in this tree in the range **[109, 163]**.



Find all elements in this tree in the range **[109, 163]**.



Find all elements in this tree in the range **[109, 163]**.

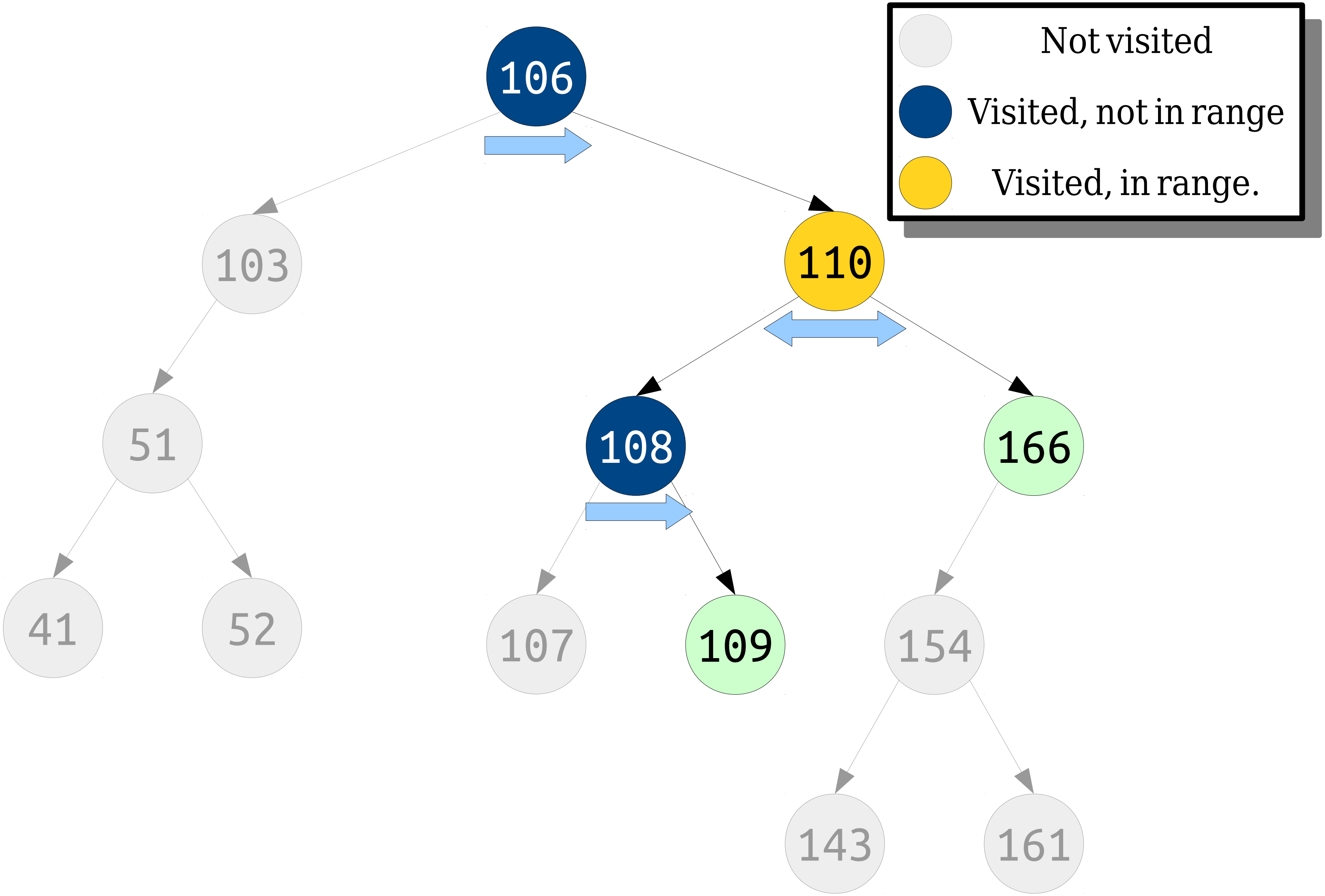


○ Not visited

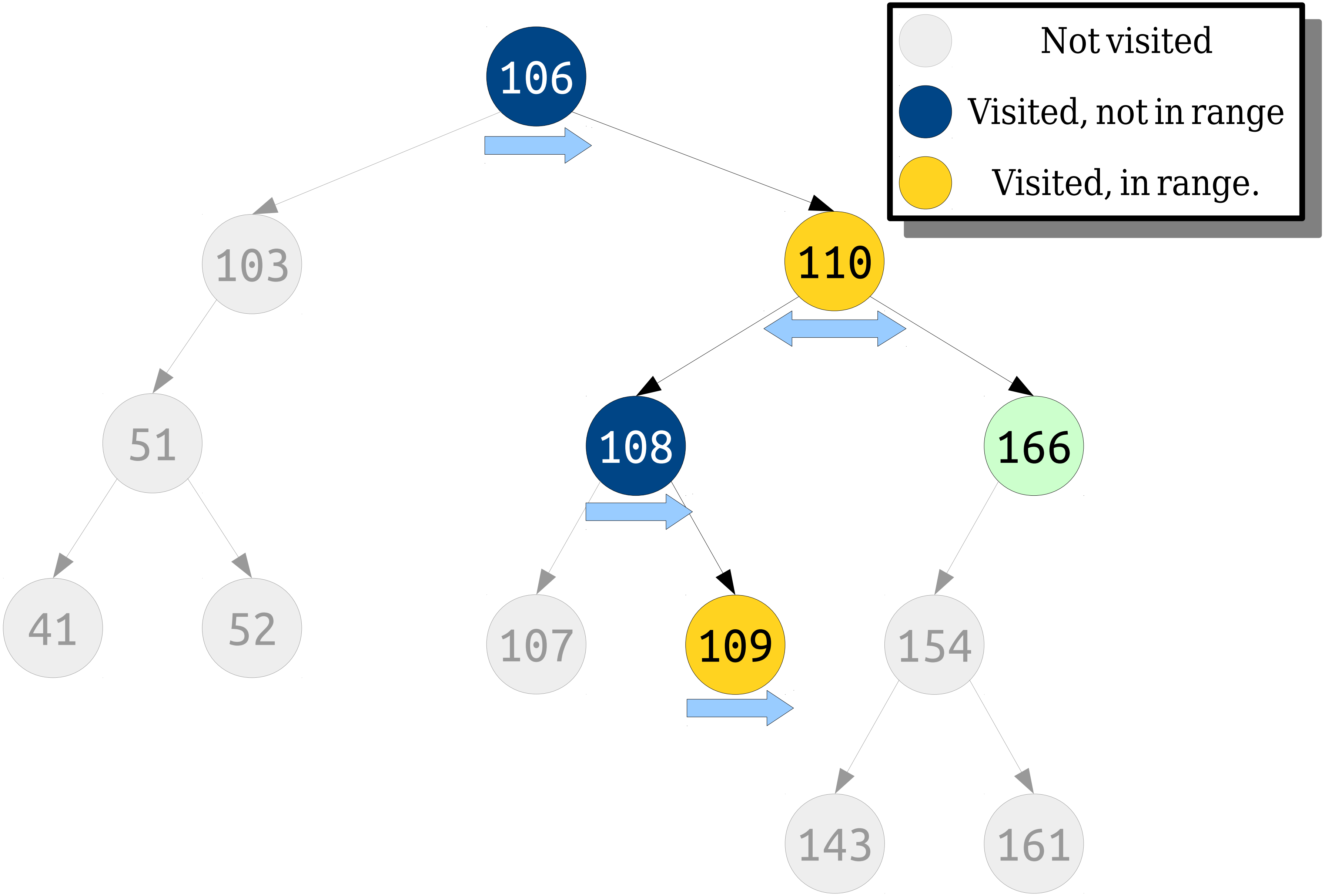
🔵 Visited, not in range

🟡 Visited, in range.

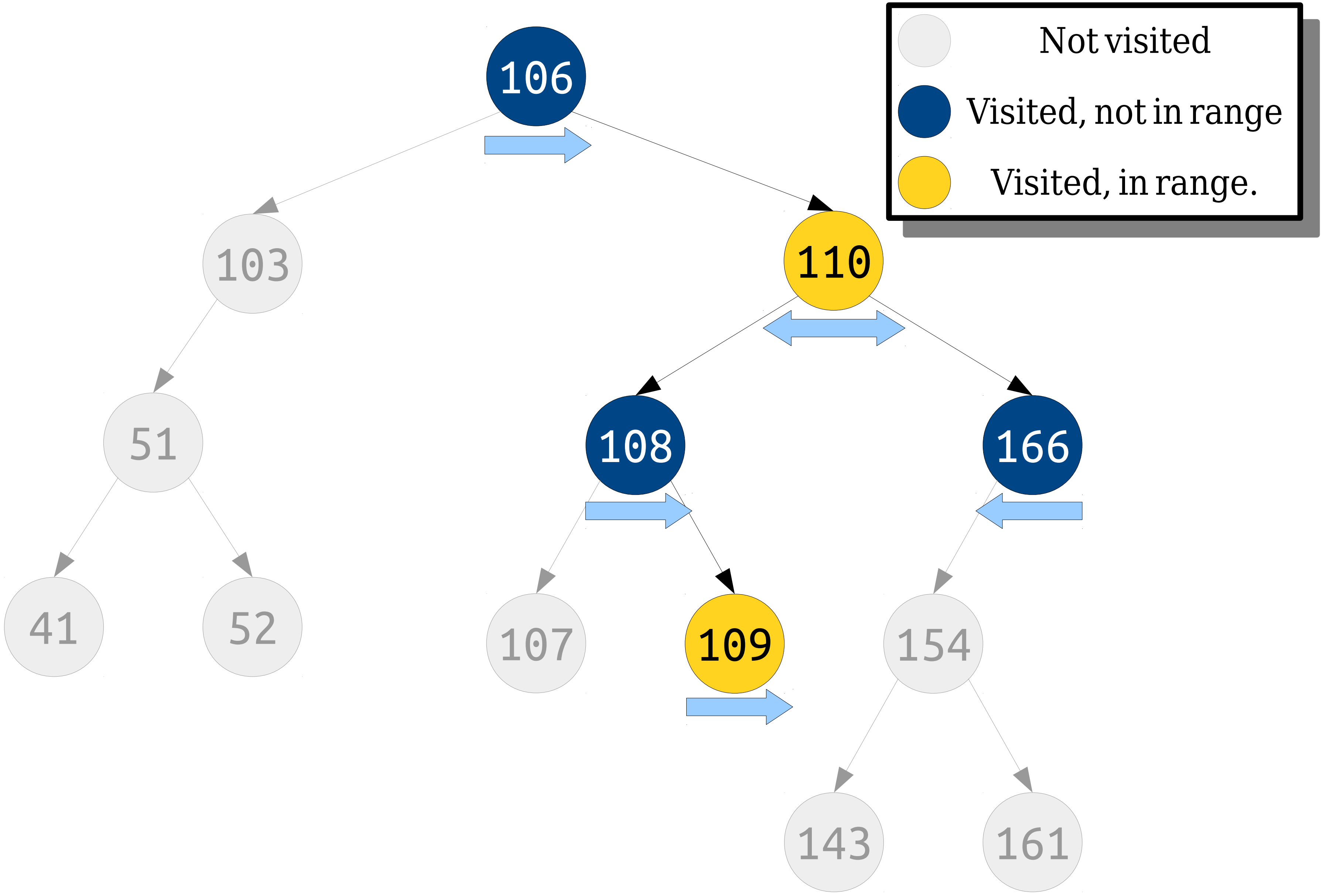
Find all elements in this tree in the range **[109, 163]**.



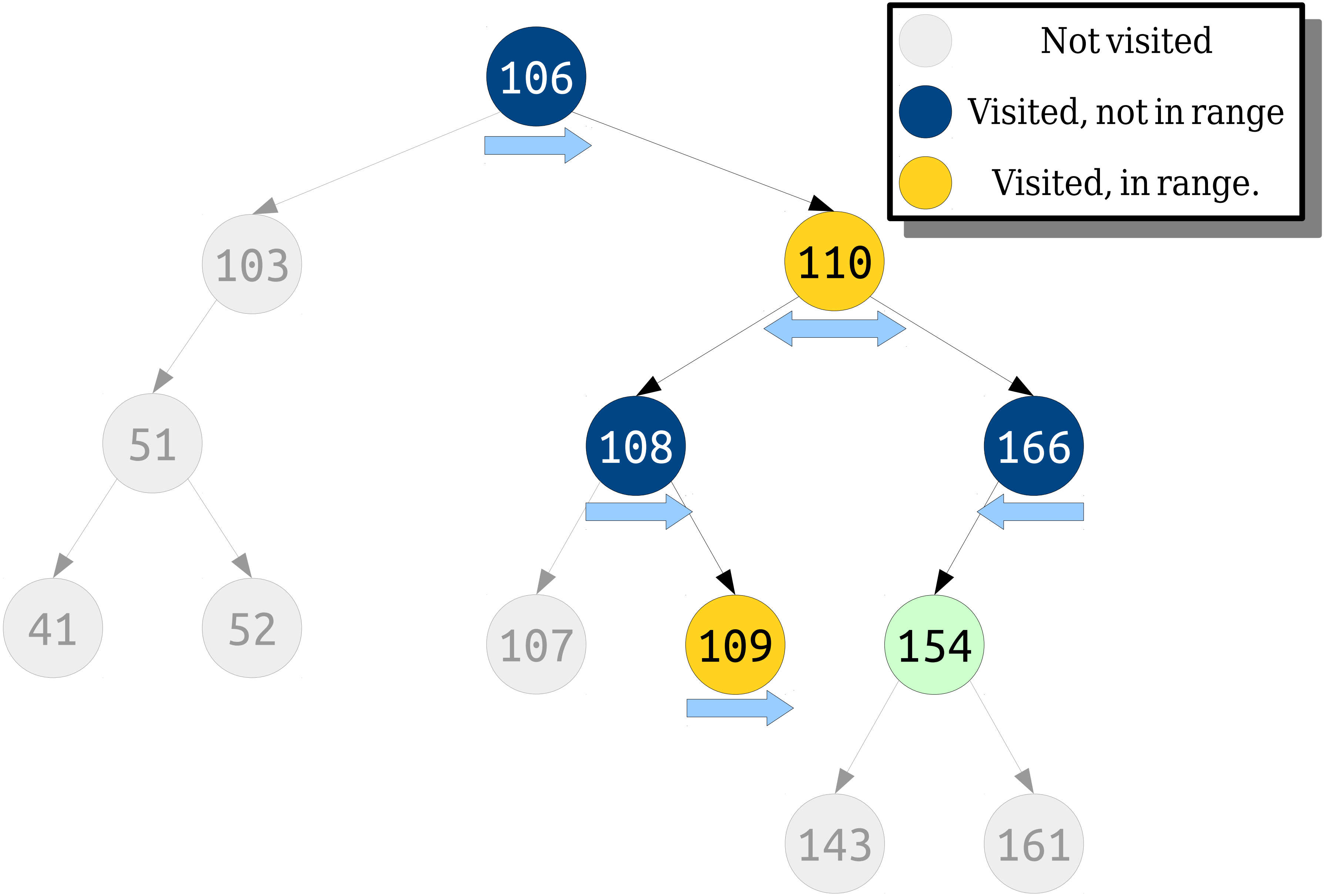
Find all elements in this tree in the range **[109, 163]**.



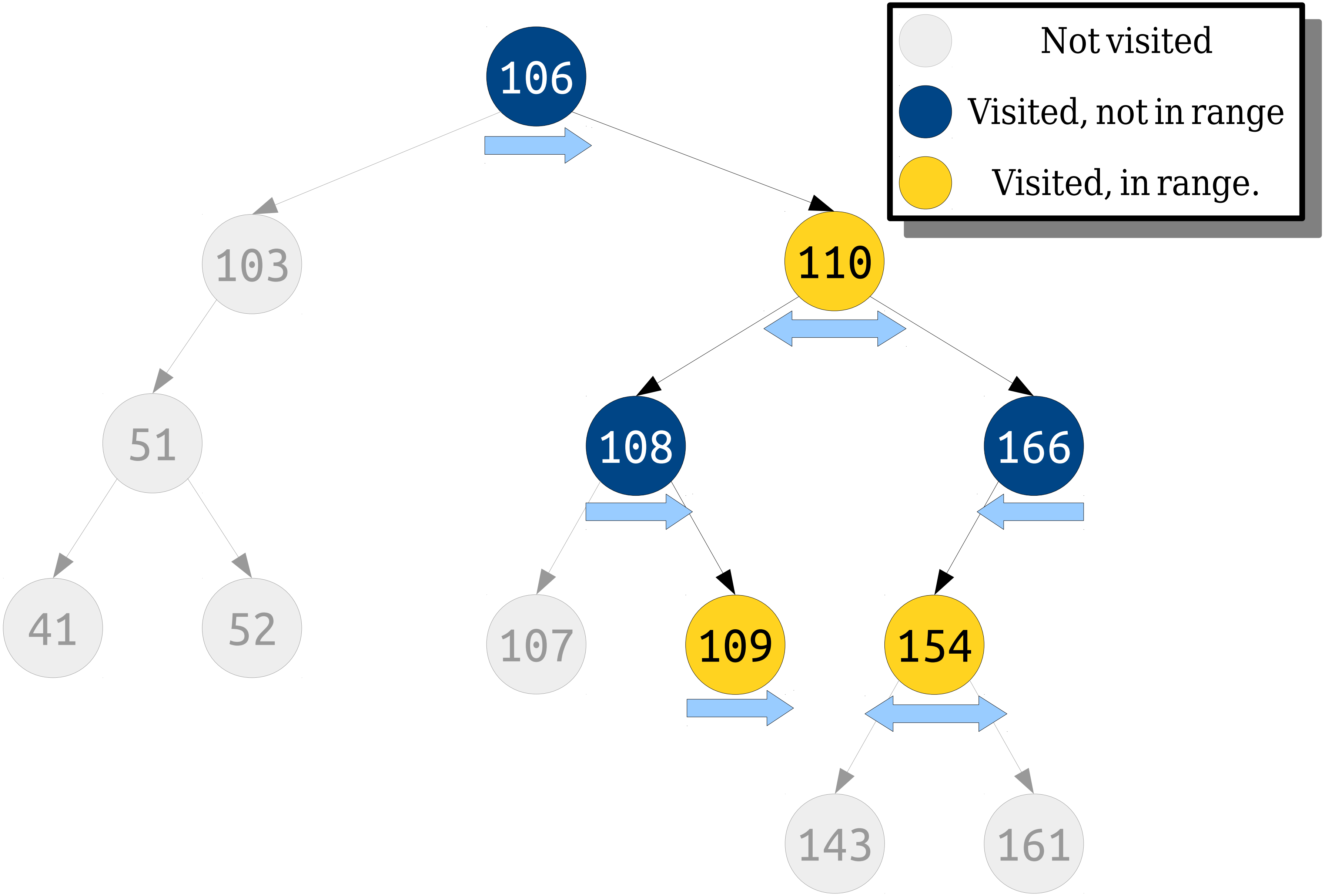
Find all elements in this tree in the range **[109, 163]**.



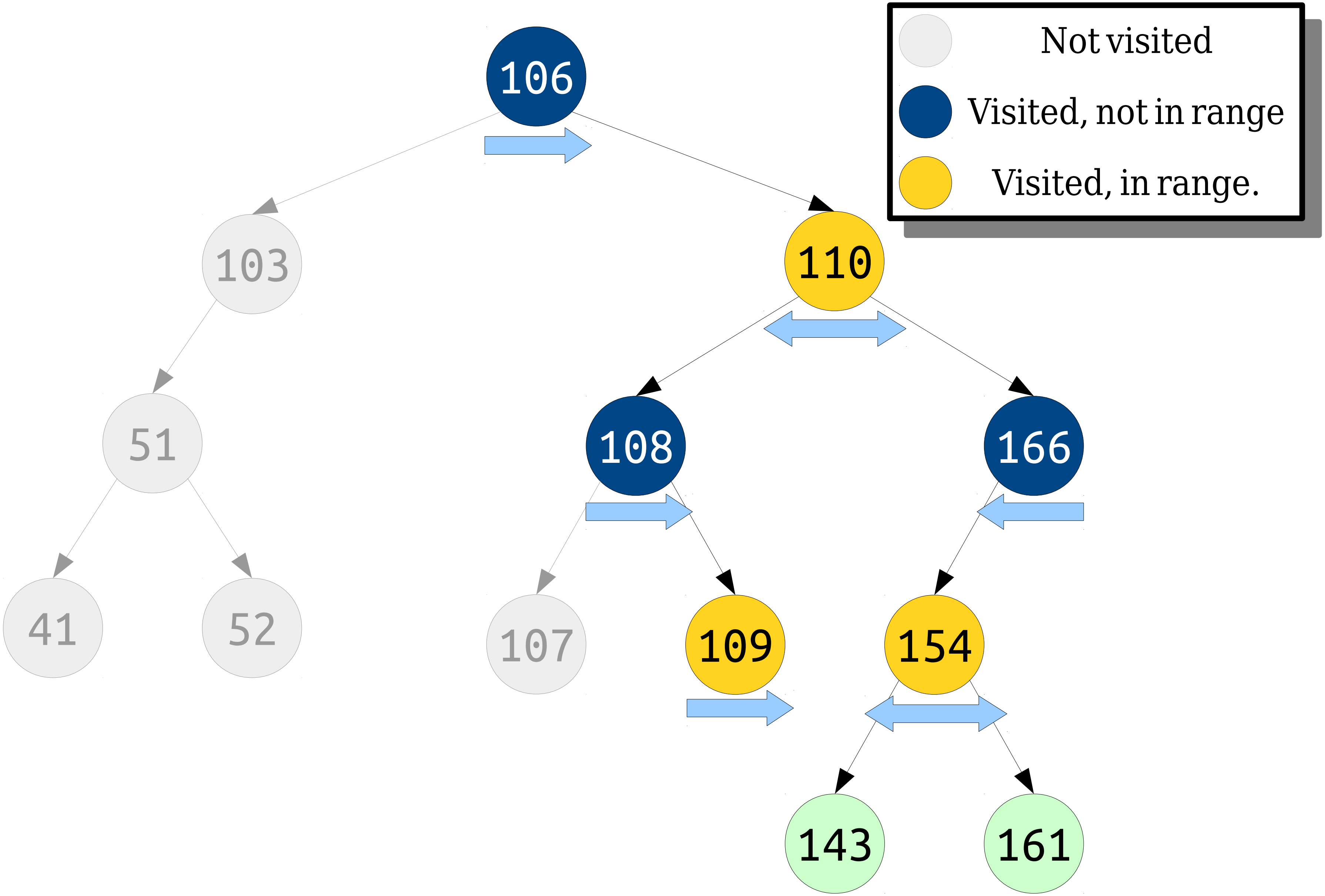
Find all elements in this tree in the range **[109, 163]**.



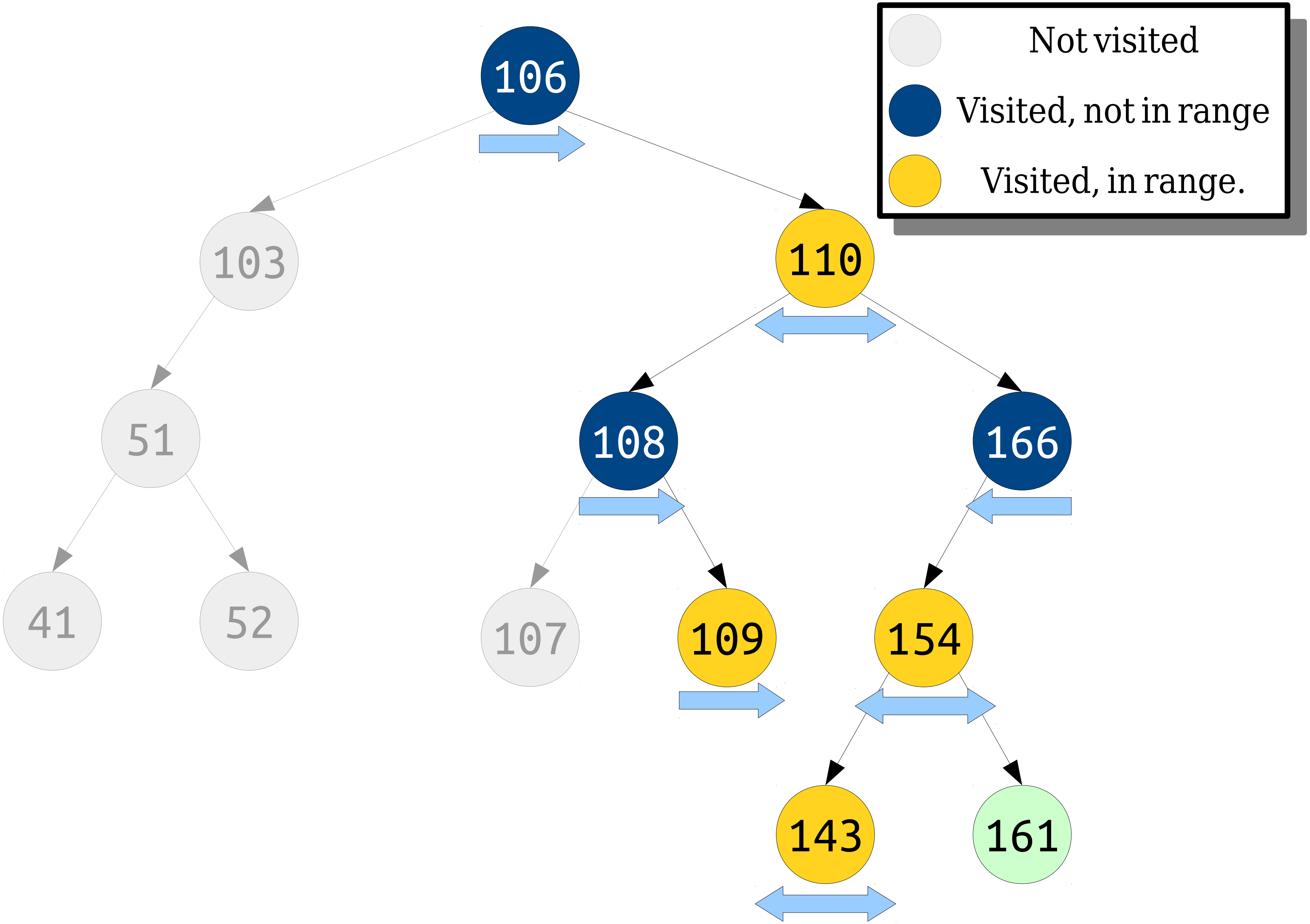
Find all elements in this tree in the range **[109, 163]**.



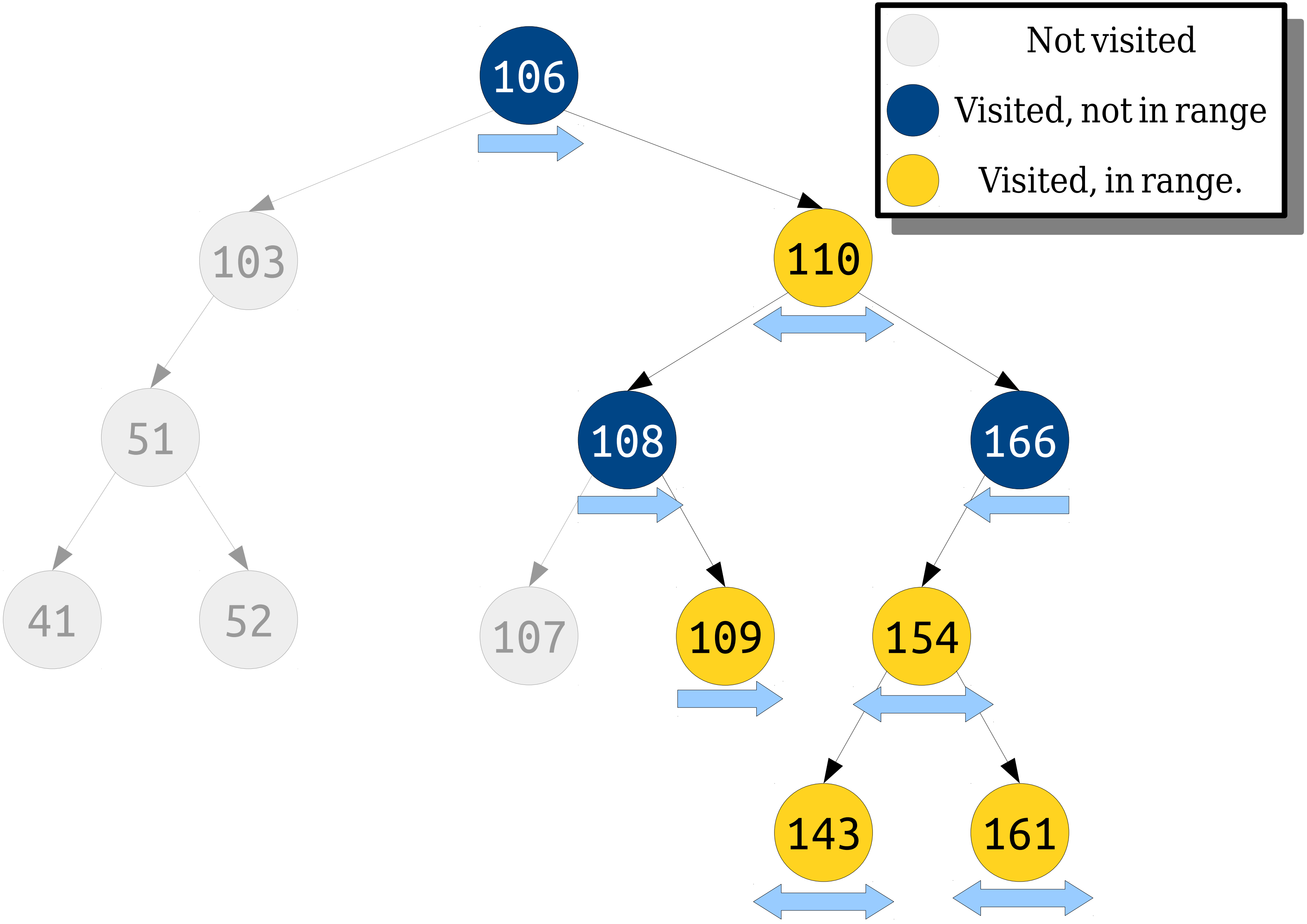
Find all elements in this tree in the range **[109, 163]**.



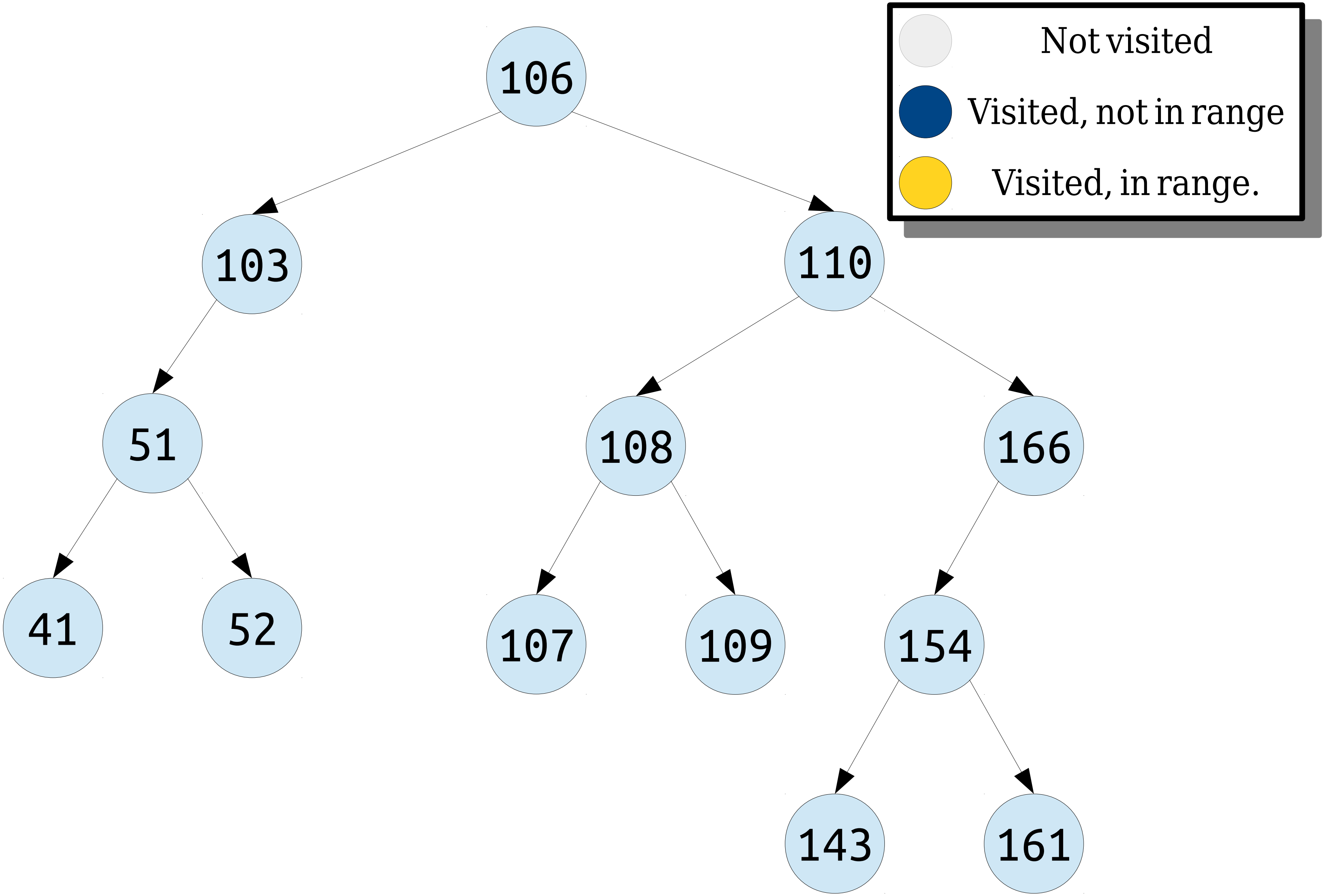
Find all elements in this tree in the range **[109, 163]**.



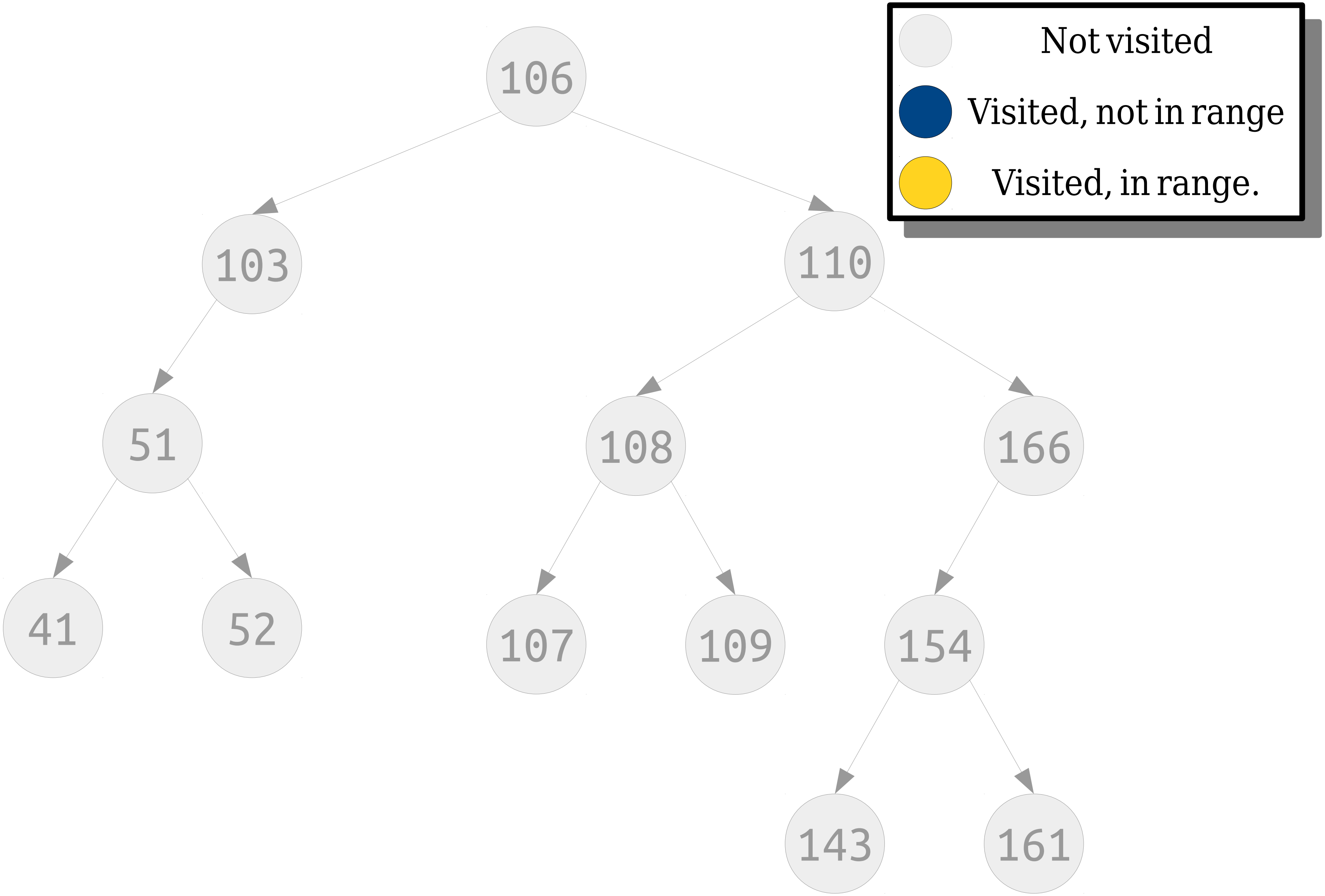
Find all elements in this tree in the range **[109, 163]**.



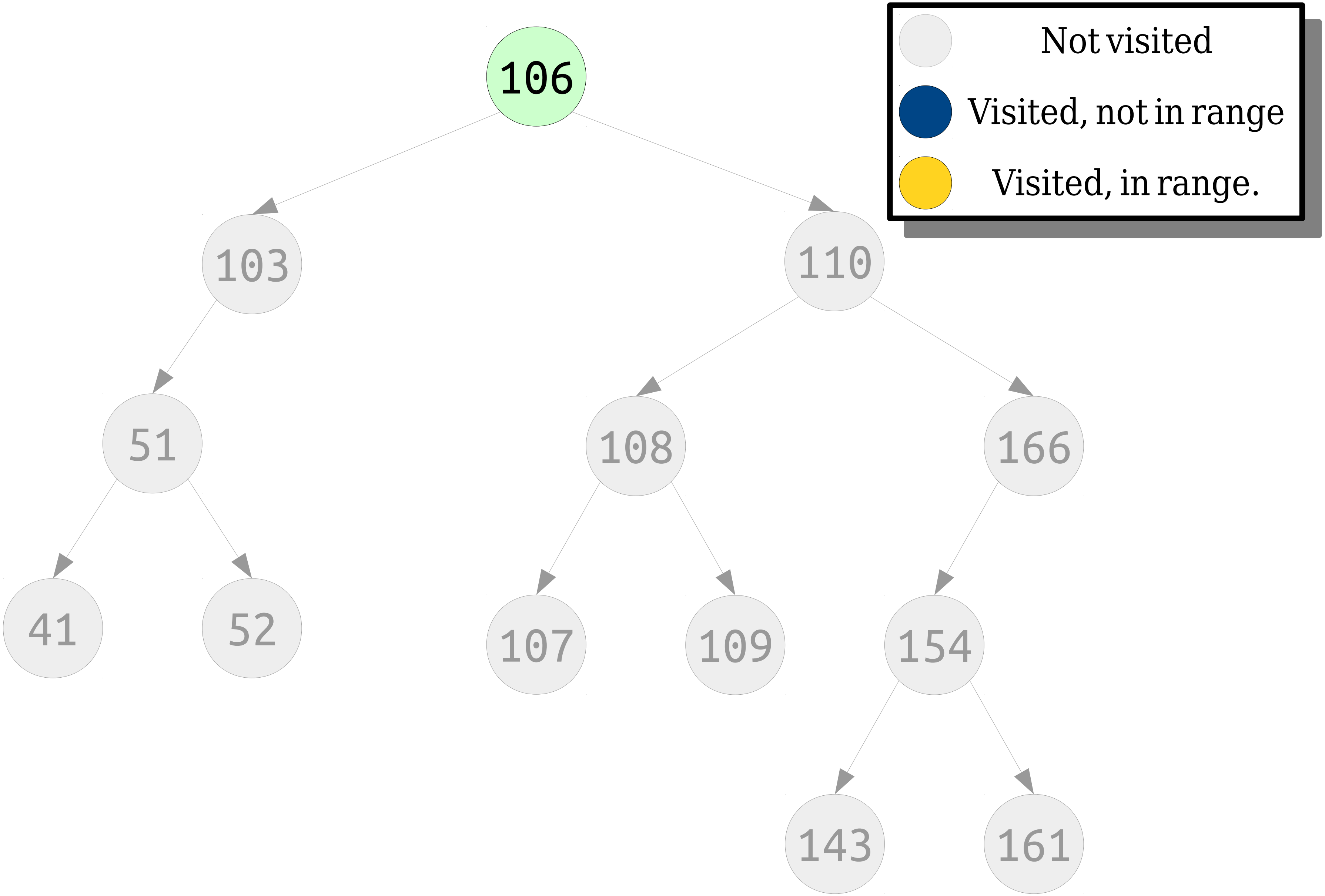
Find all elements in this tree in the range **[109, 163]**.



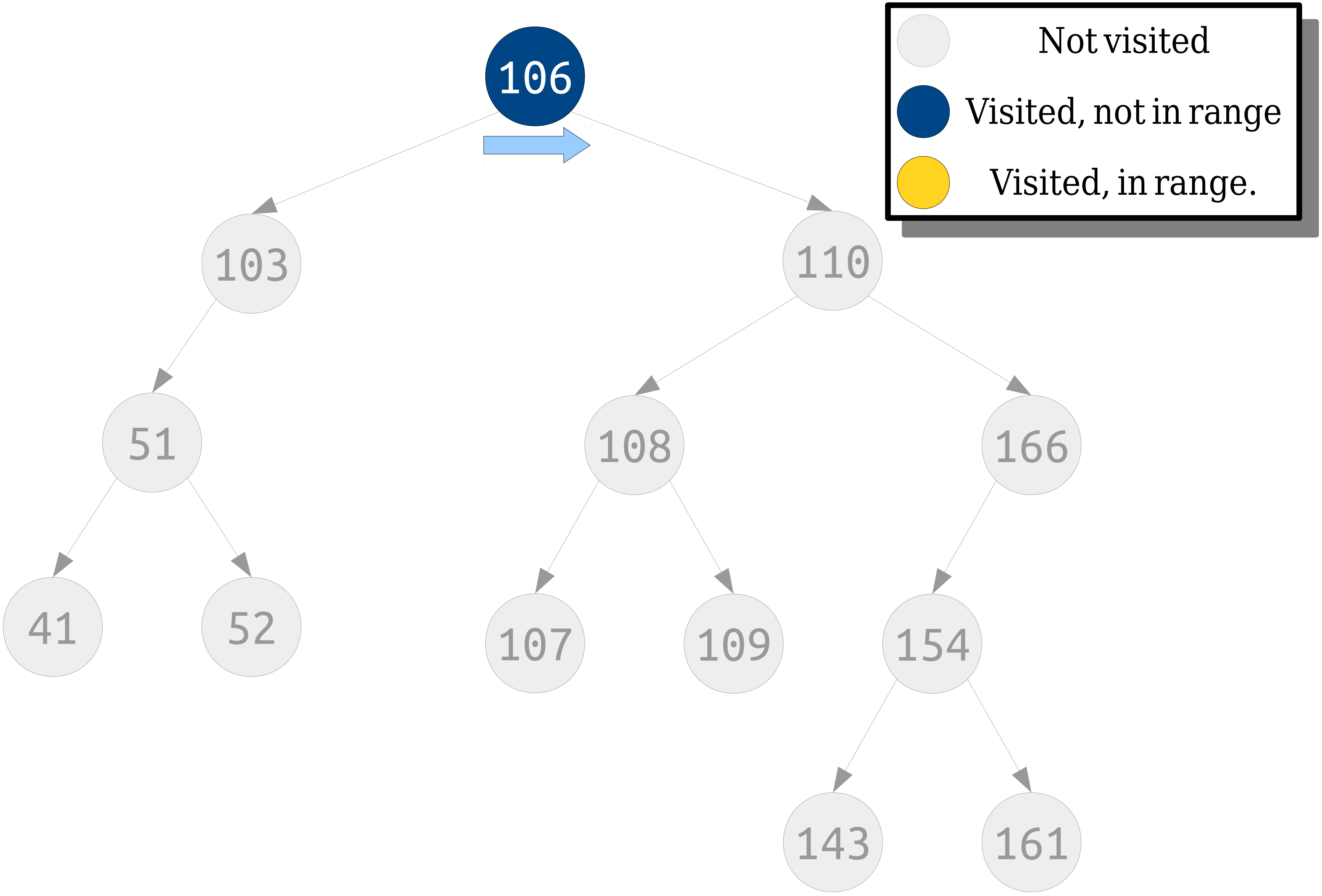
Find all elements in this tree in the range **[124, 155]**.



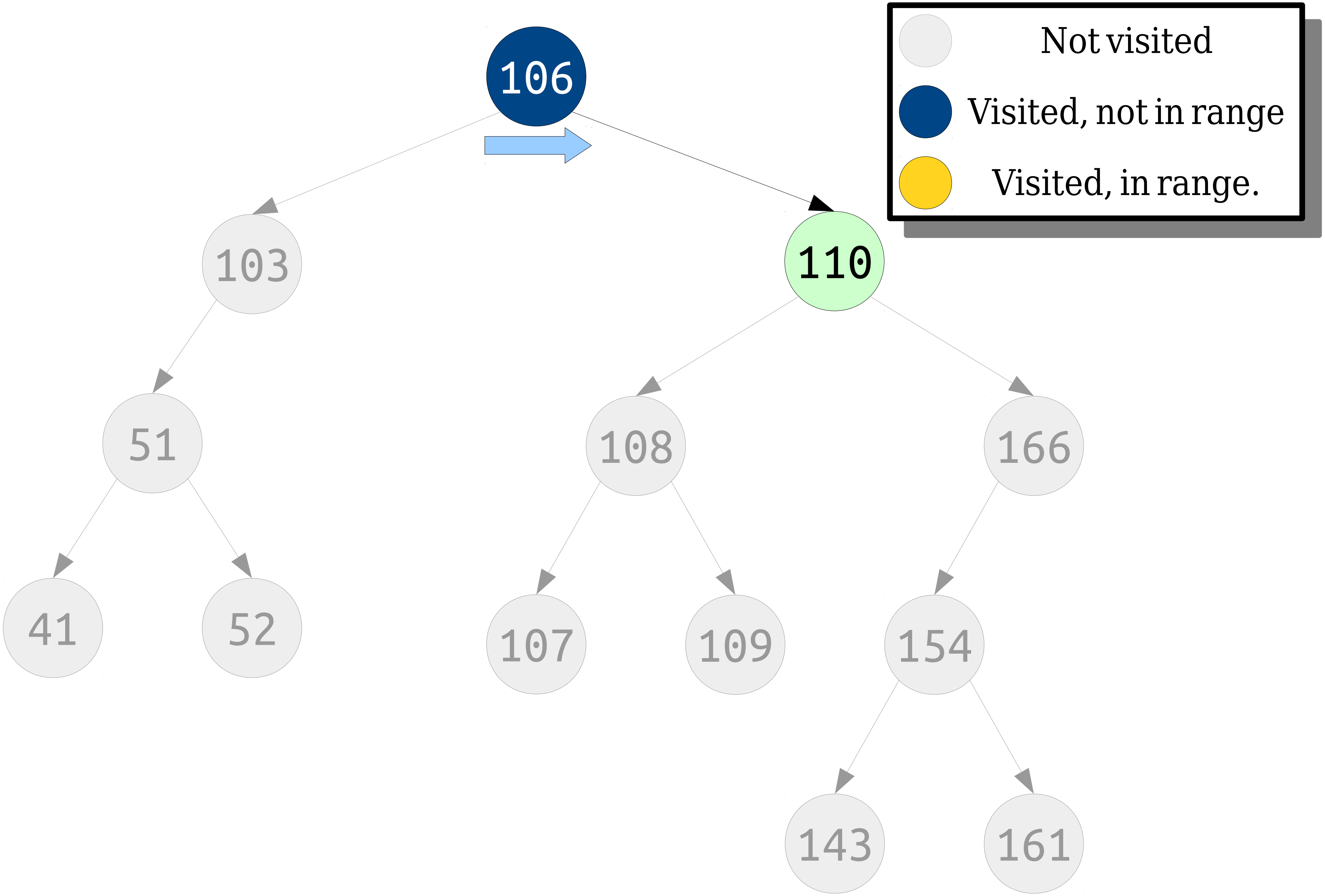
Find all elements in this tree in the range **[124, 155]**.



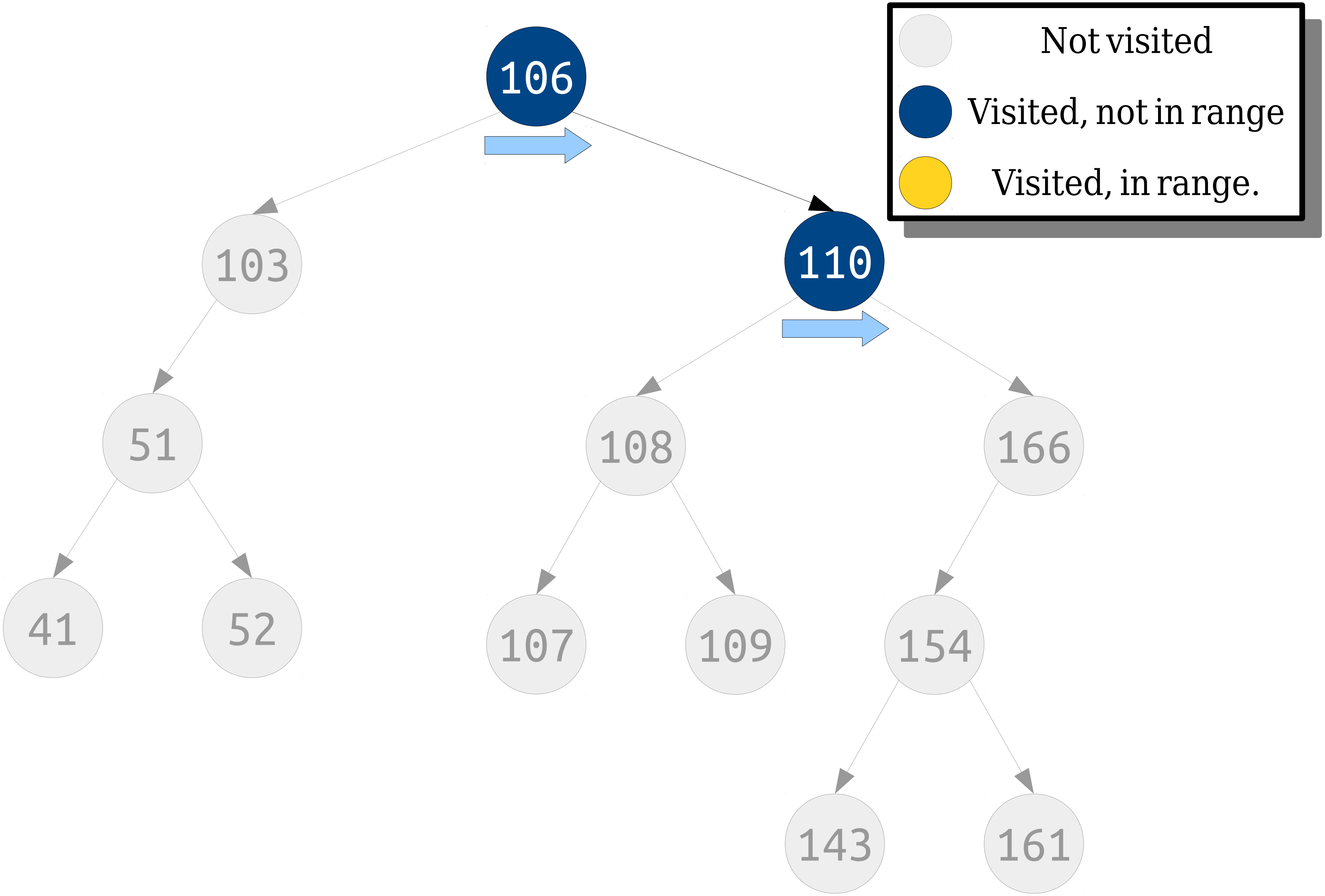
Find all elements in this tree in the range **[124, 155]**.



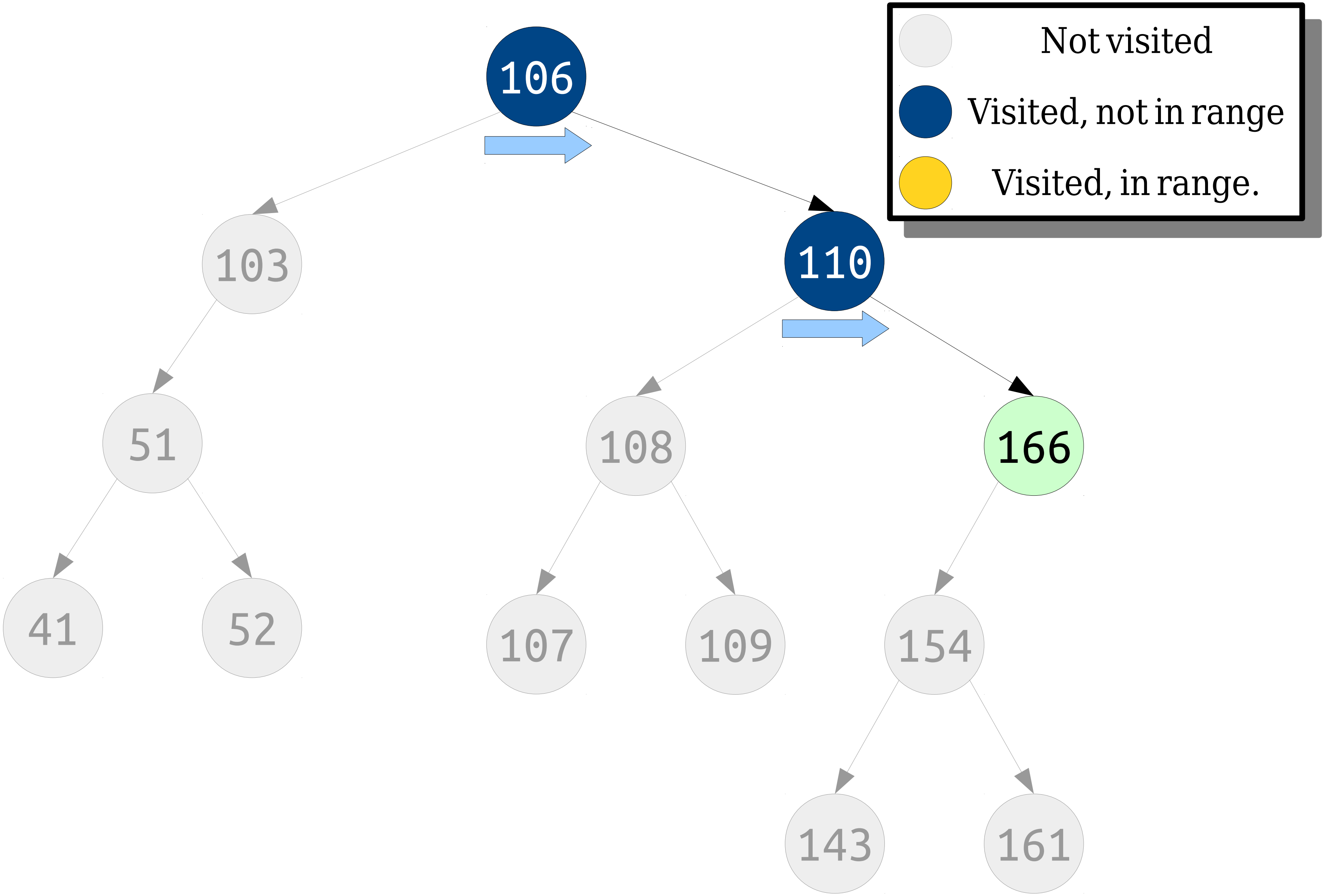
Find all elements in this tree in the range **[124, 155]**.



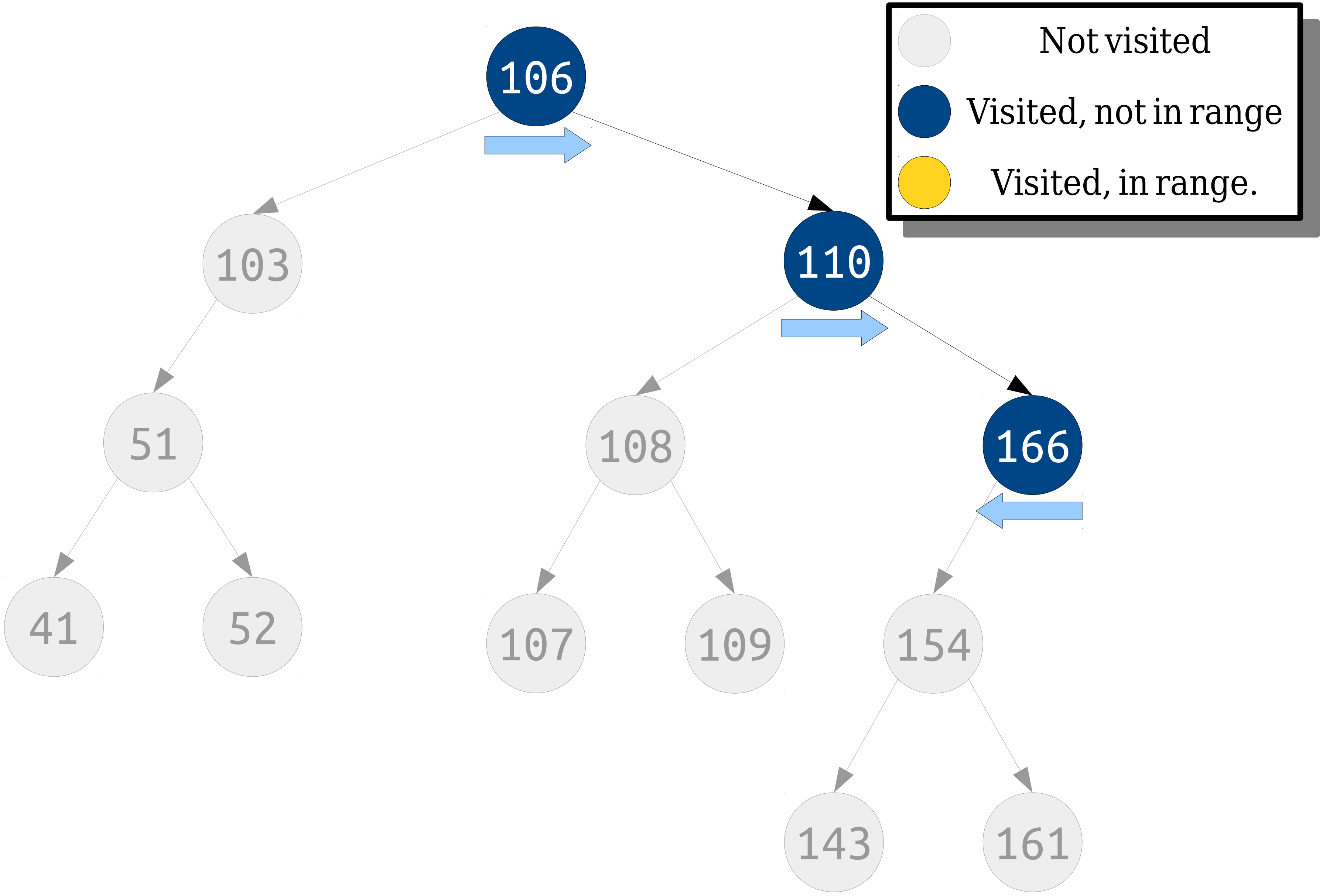
Find all elements in this tree in the range **[124, 155]**.



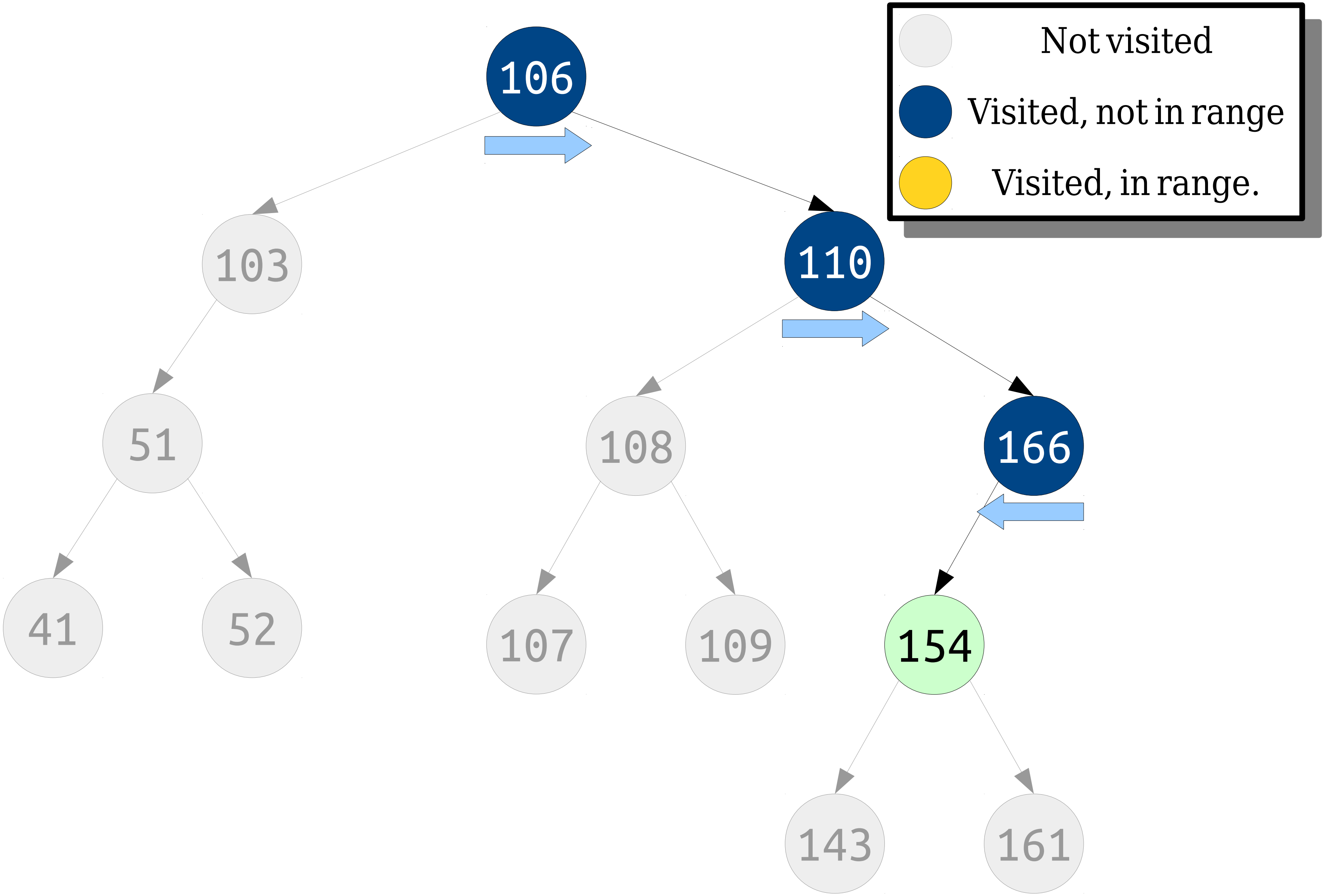
Find all elements in this tree in the range **[124, 155]**.



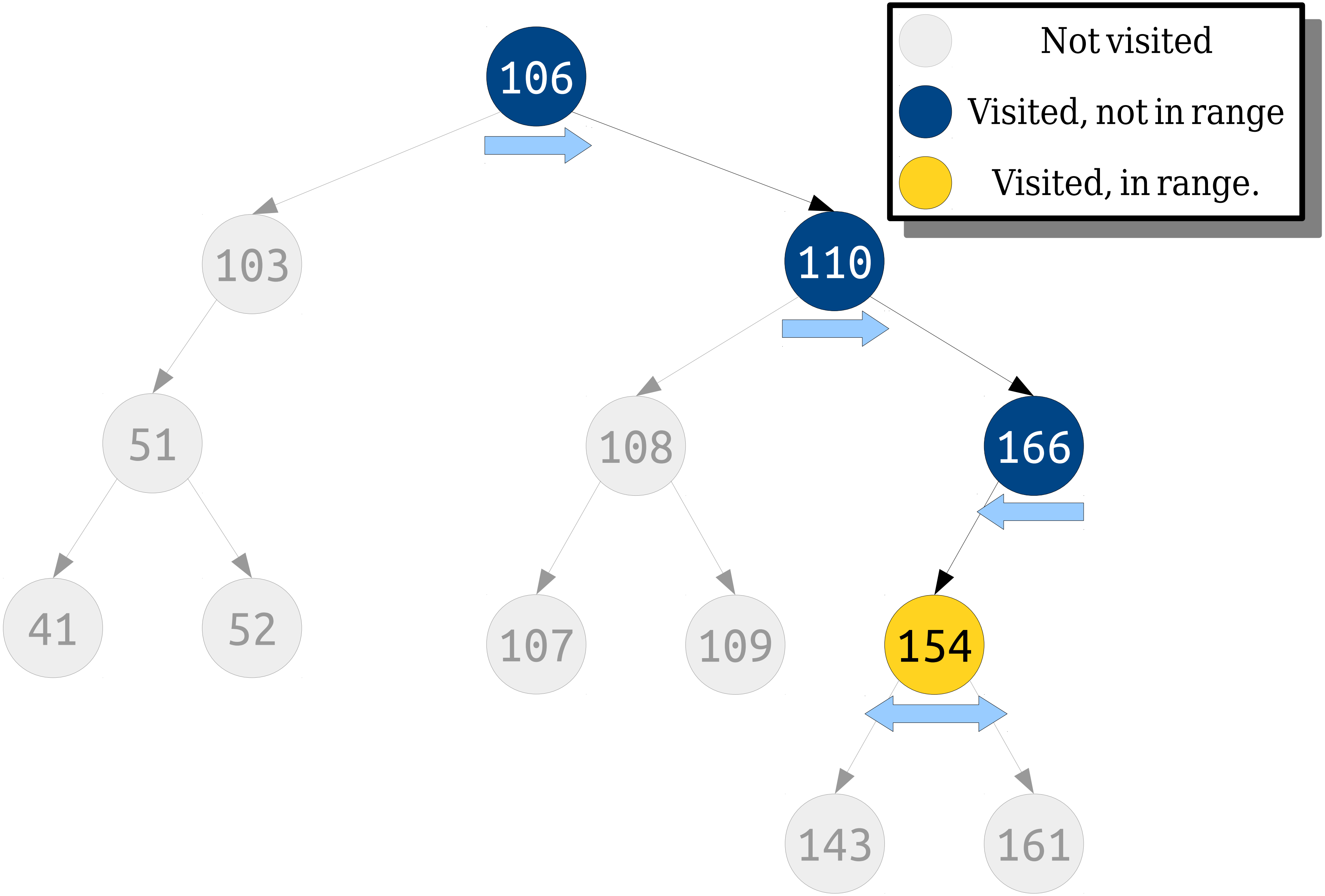
Find all elements in this tree in the range **[124, 155]**.



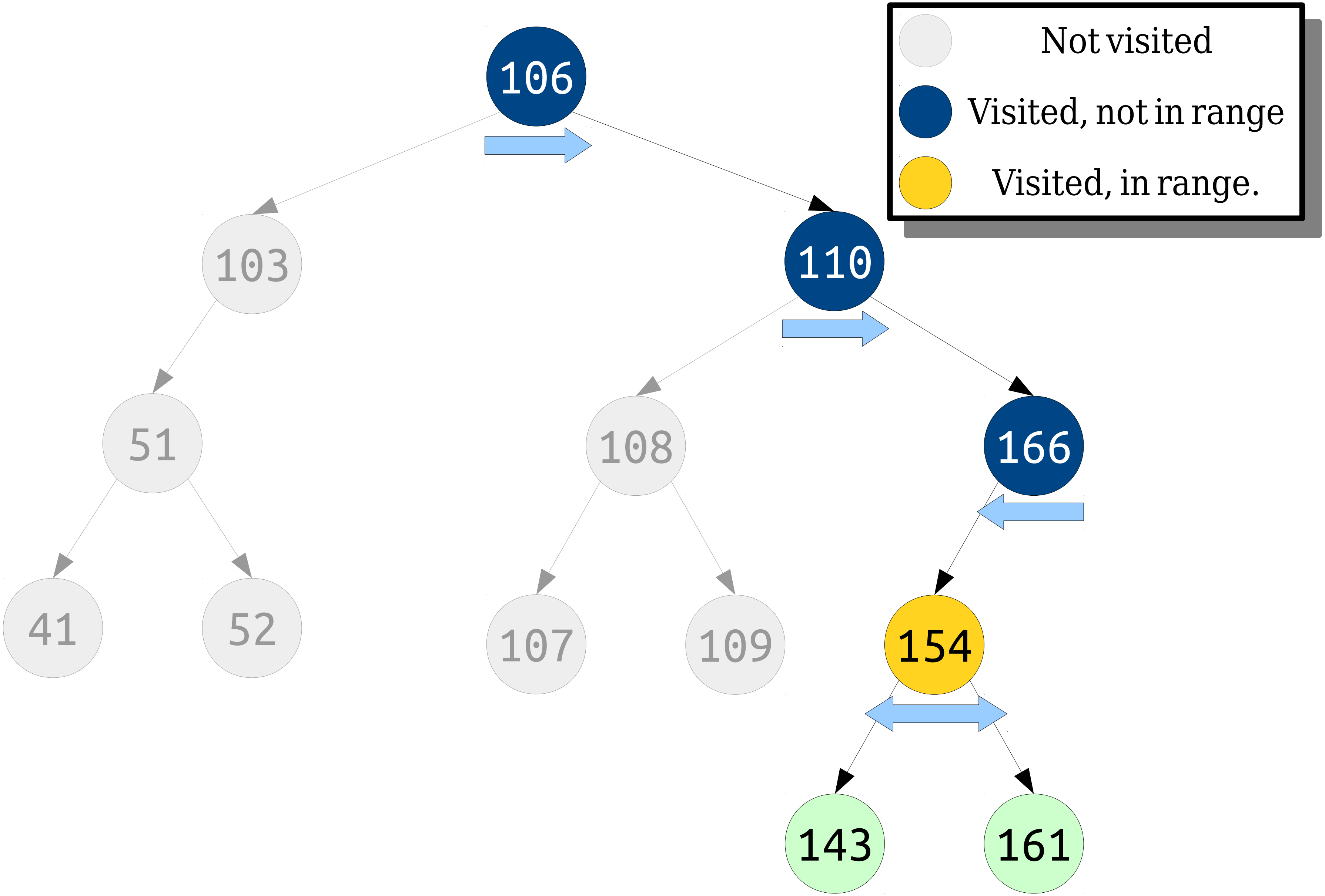
Find all elements in this tree in the range **[124, 155]**.



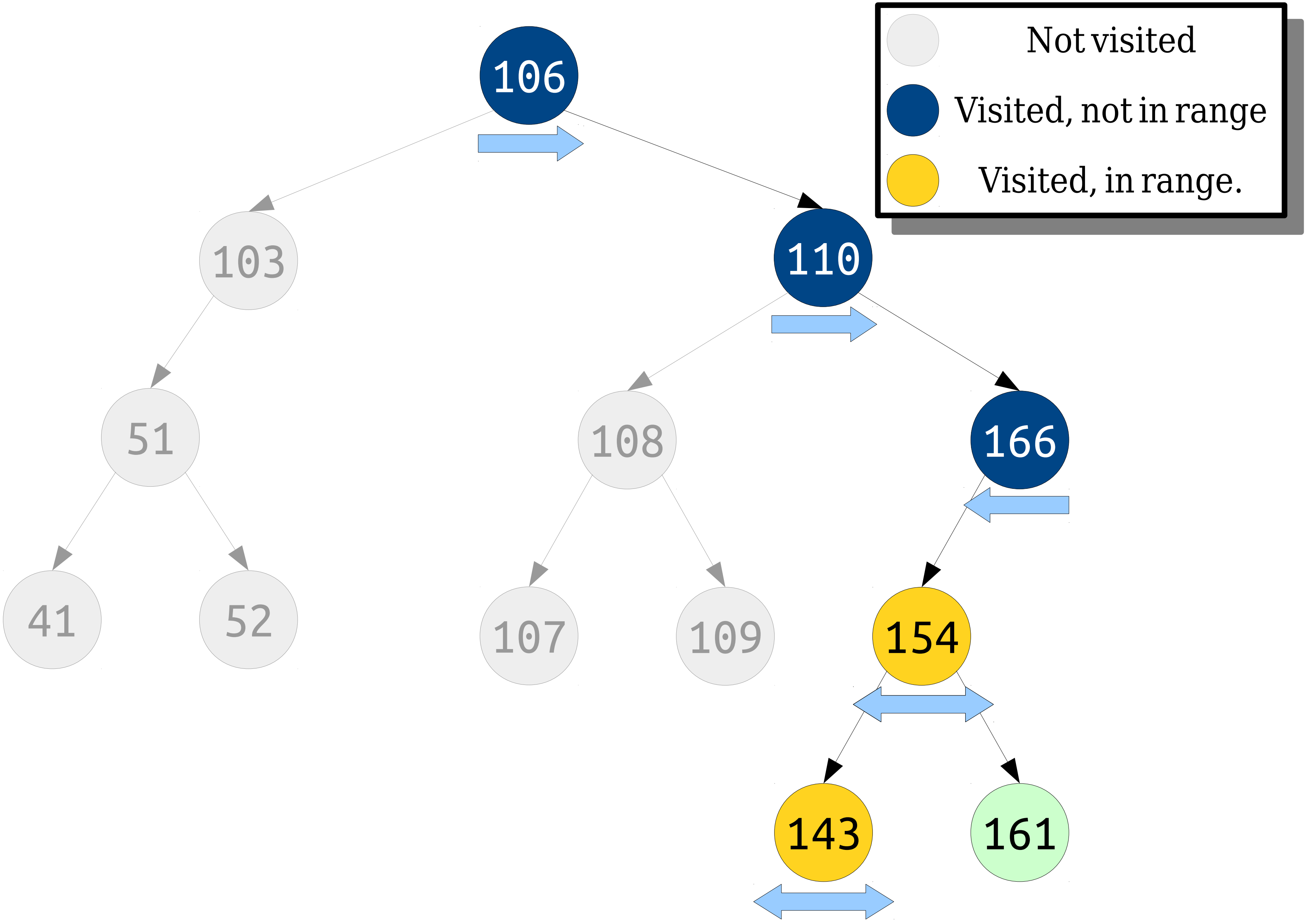
Find all elements in this tree in the range **[124, 155]**.



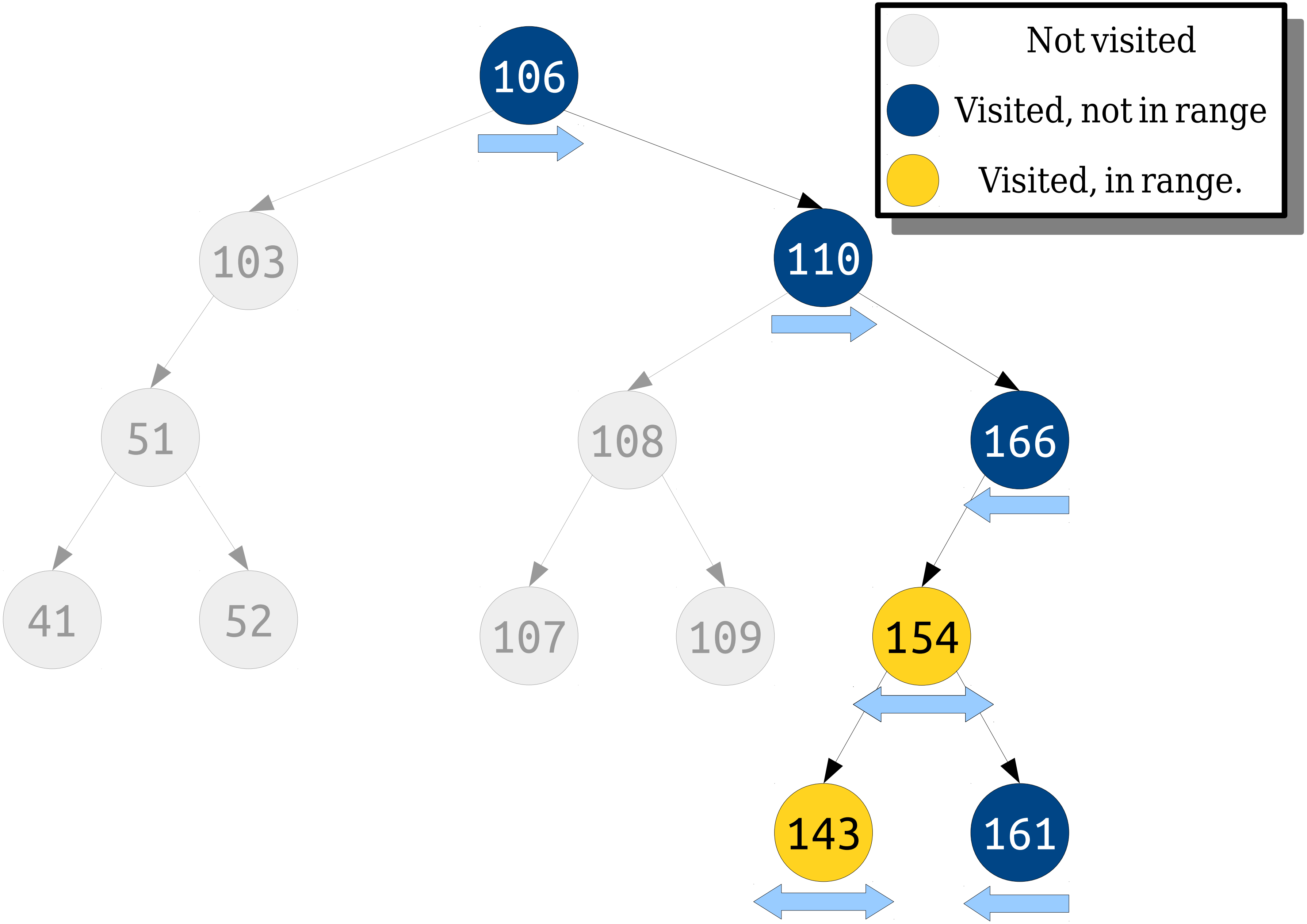
Find all elements in this tree in the range **[124, 155]**.



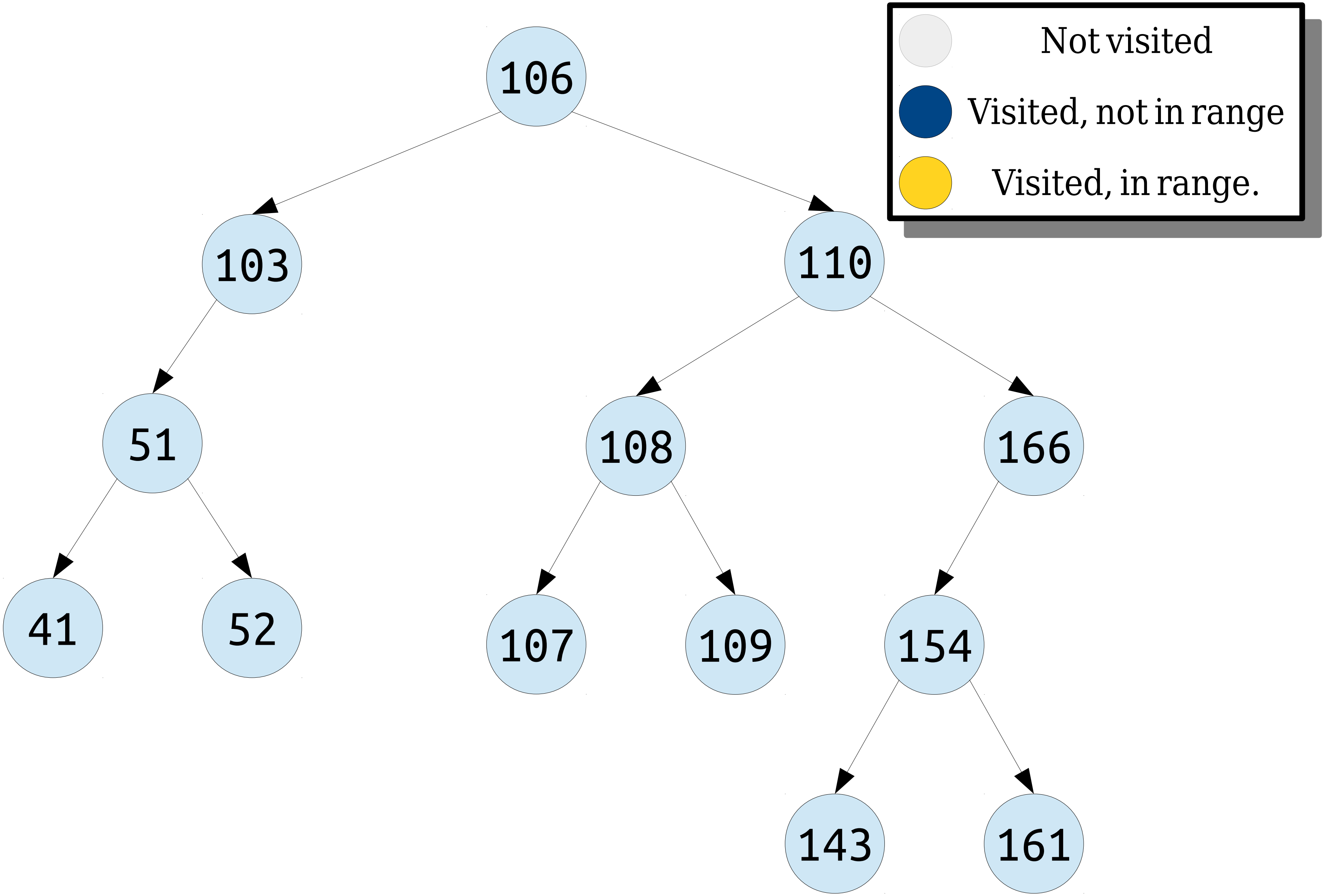
Find all elements in this tree in the range **[124, 155]**.



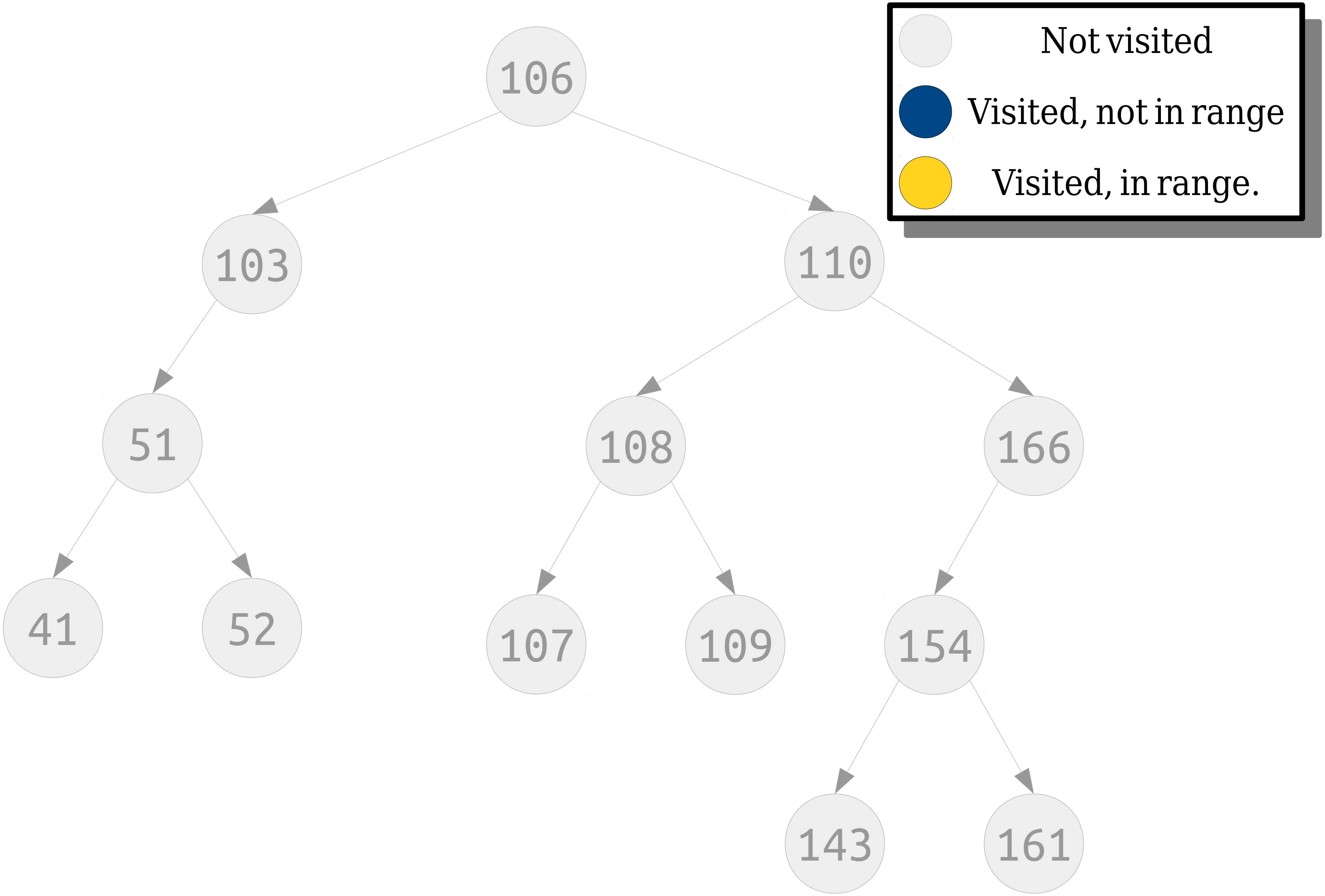
Find all elements in this tree in the range **[124, 155]**.



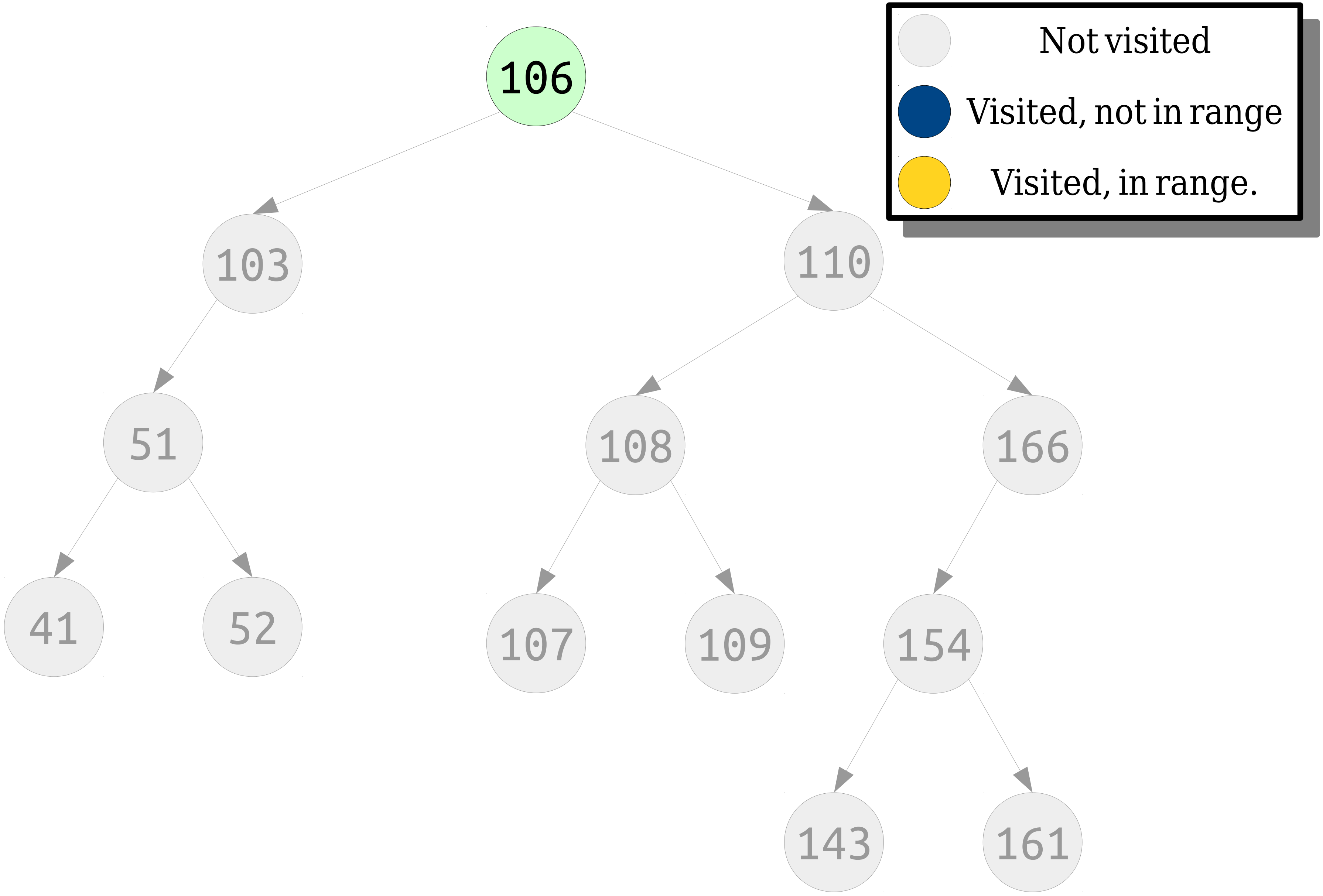
Find all elements in this tree in the range **[124, 155]**.



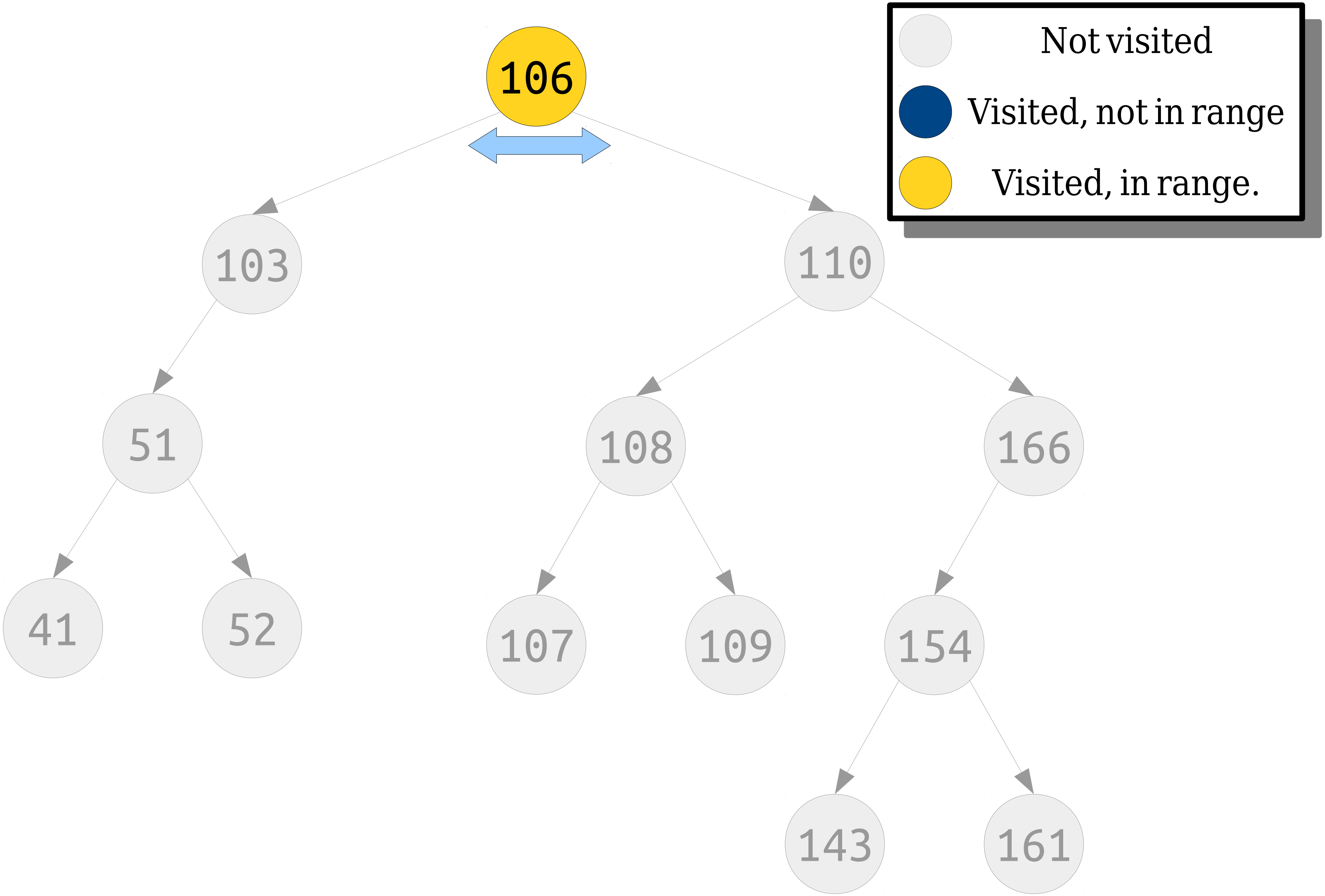
Find all elements in this tree in the range **[42, 165]**.



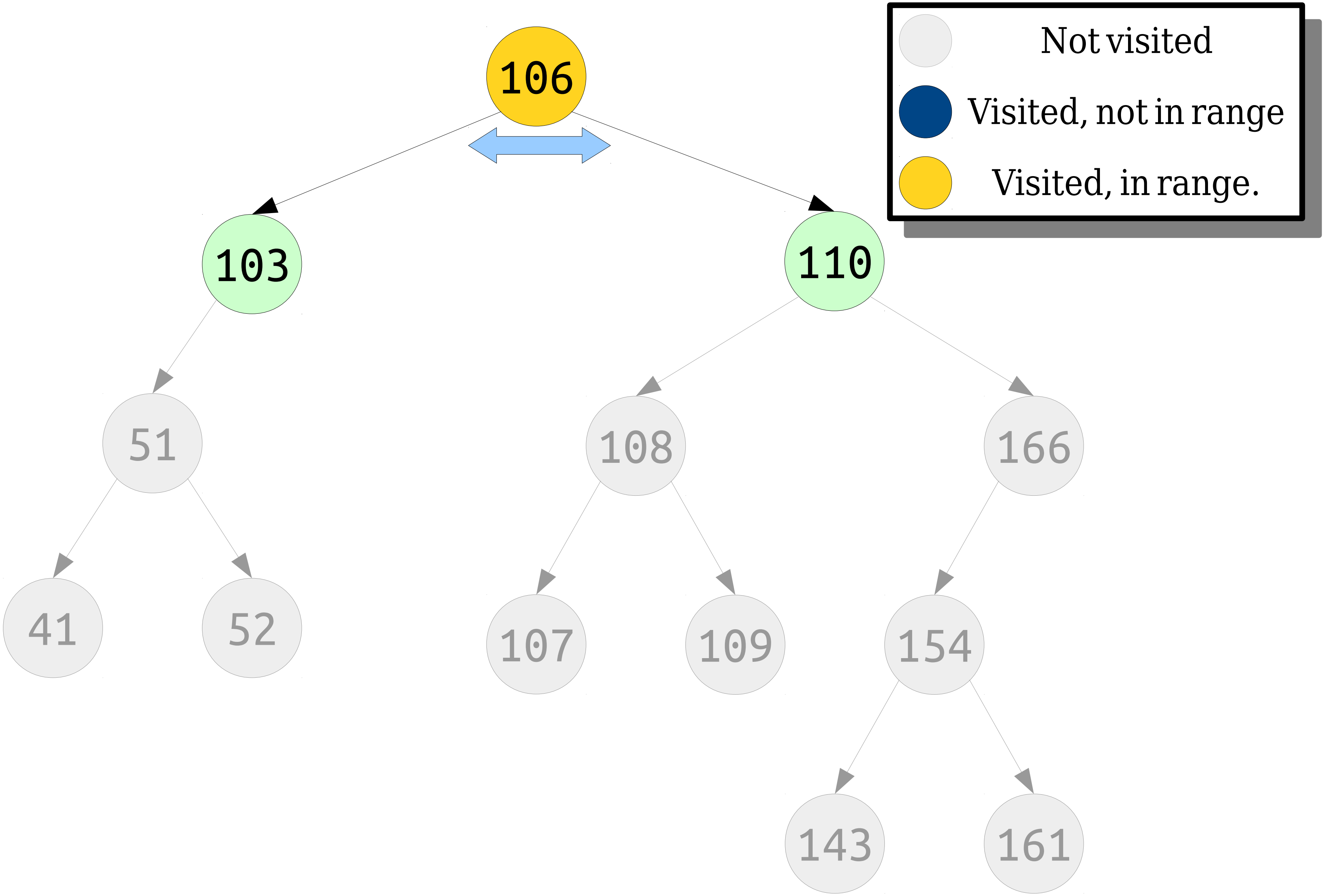
Find all elements in this tree in the range **[42, 165]**.



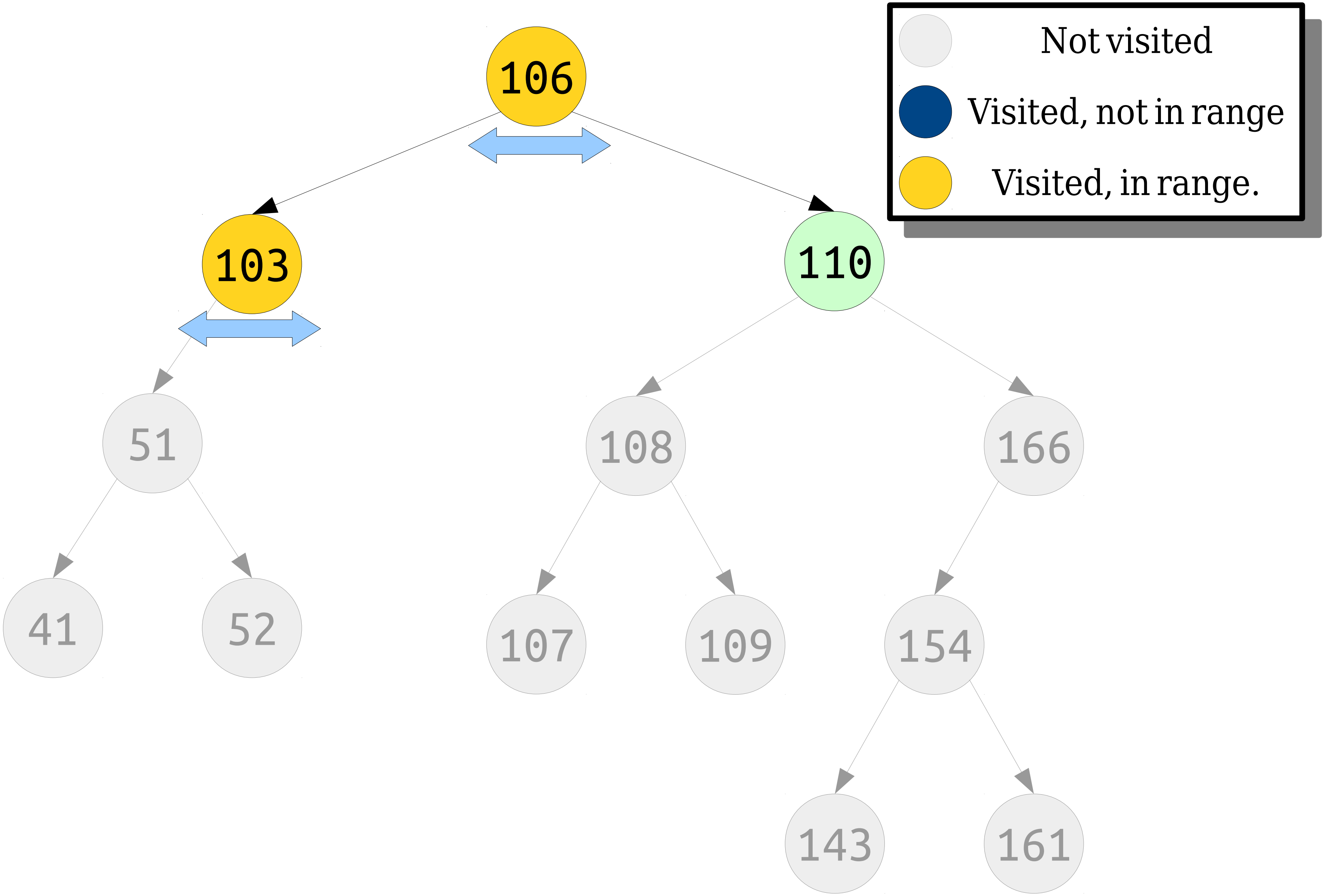
Find all elements in this tree in the range **[42, 165]**.



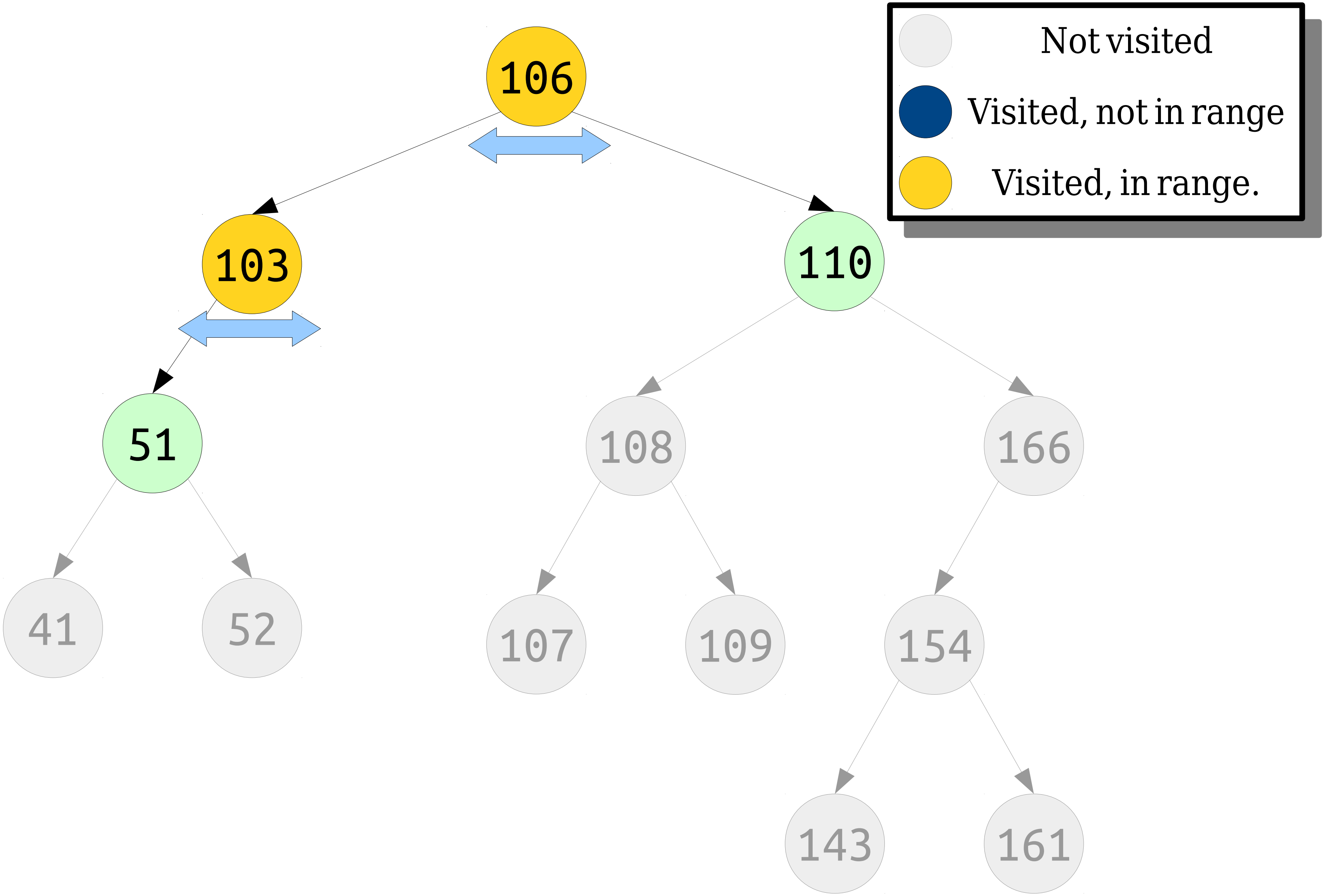
Find all elements in this tree in the range **[42, 165]**.



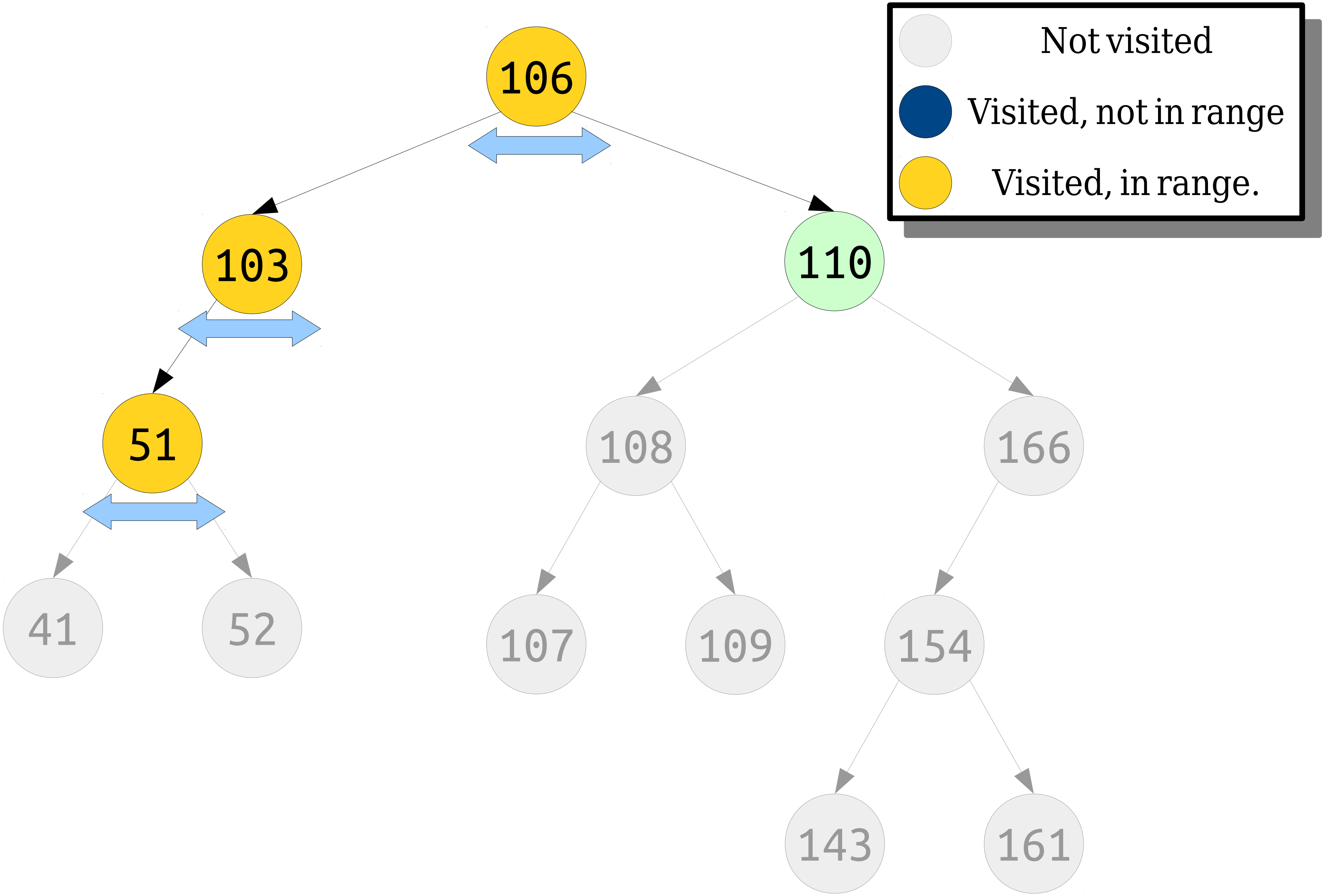
Find all elements in this tree in the range **[42, 165]**.



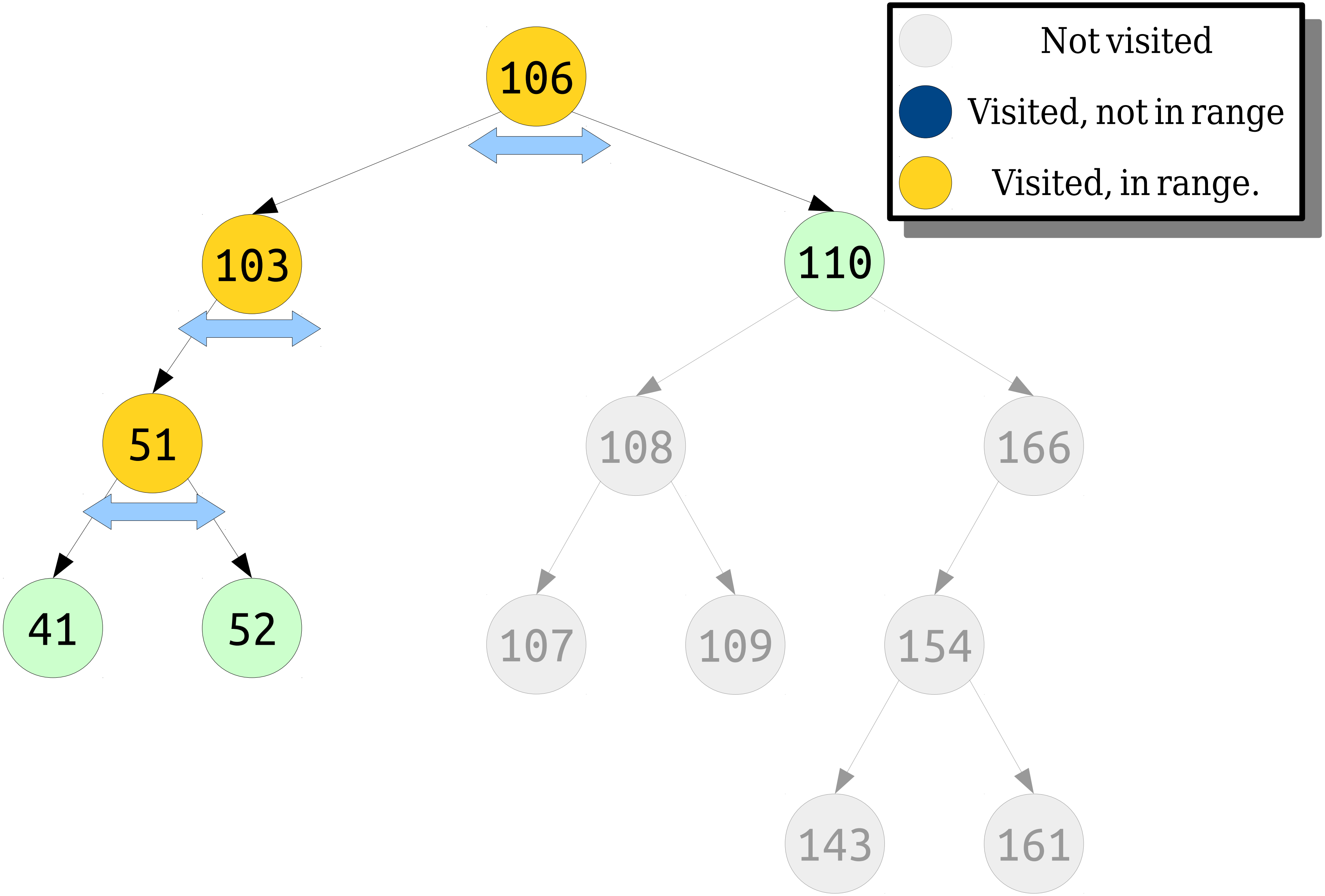
Find all elements in this tree in the range **[42, 165]**.



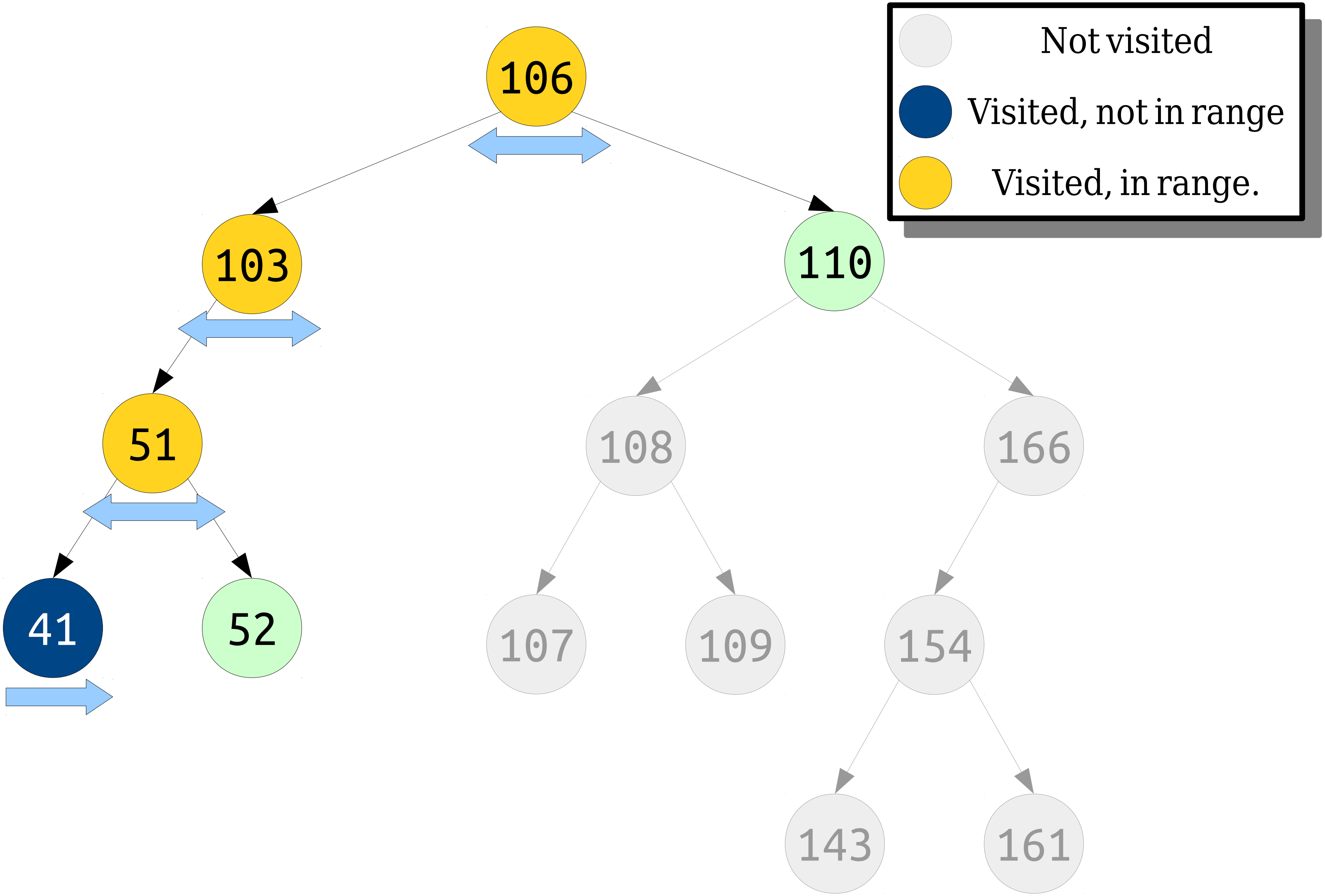
Find all elements in this tree in the range **[42, 165]**.



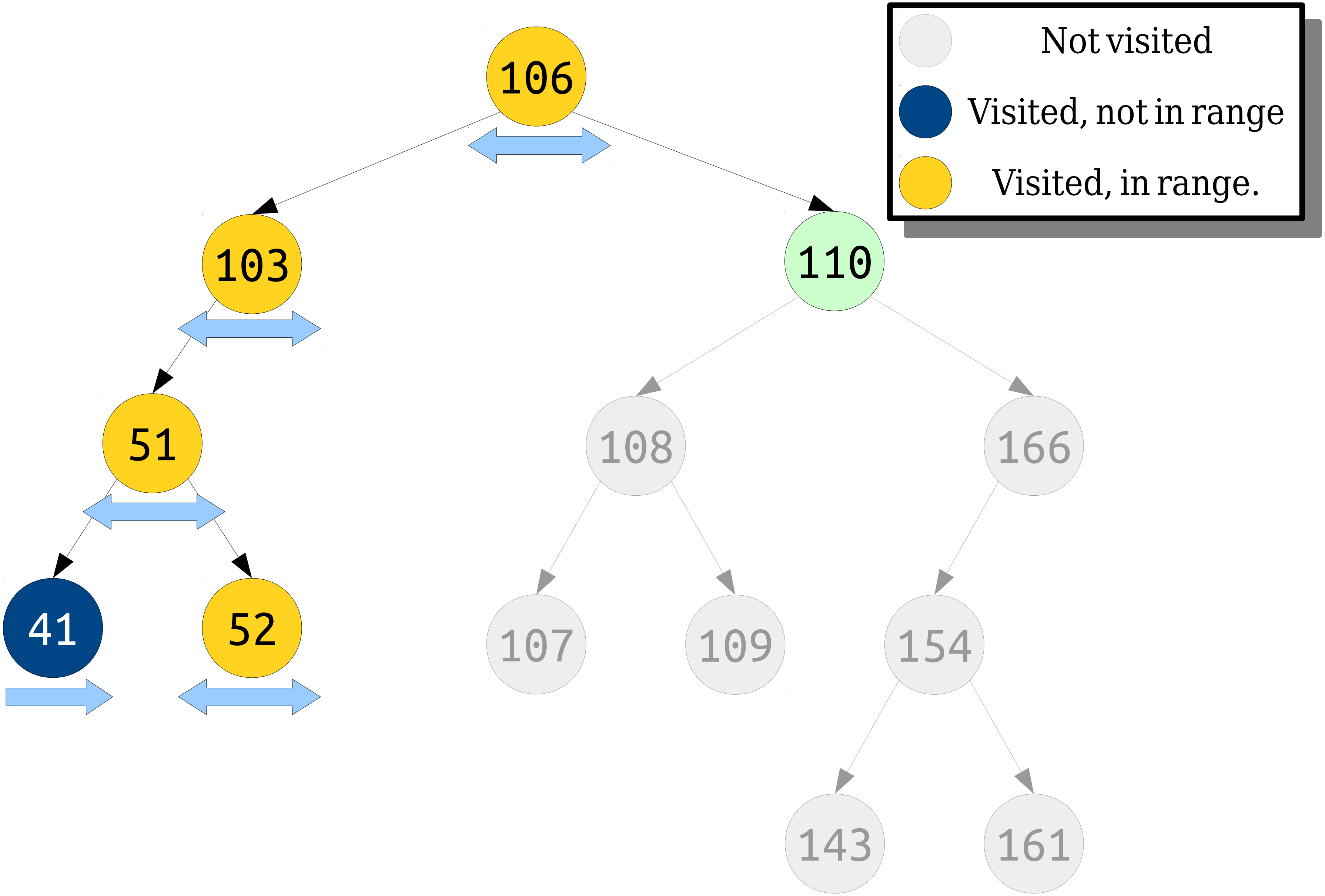
Find all elements in this tree in the range **[42, 165]**.



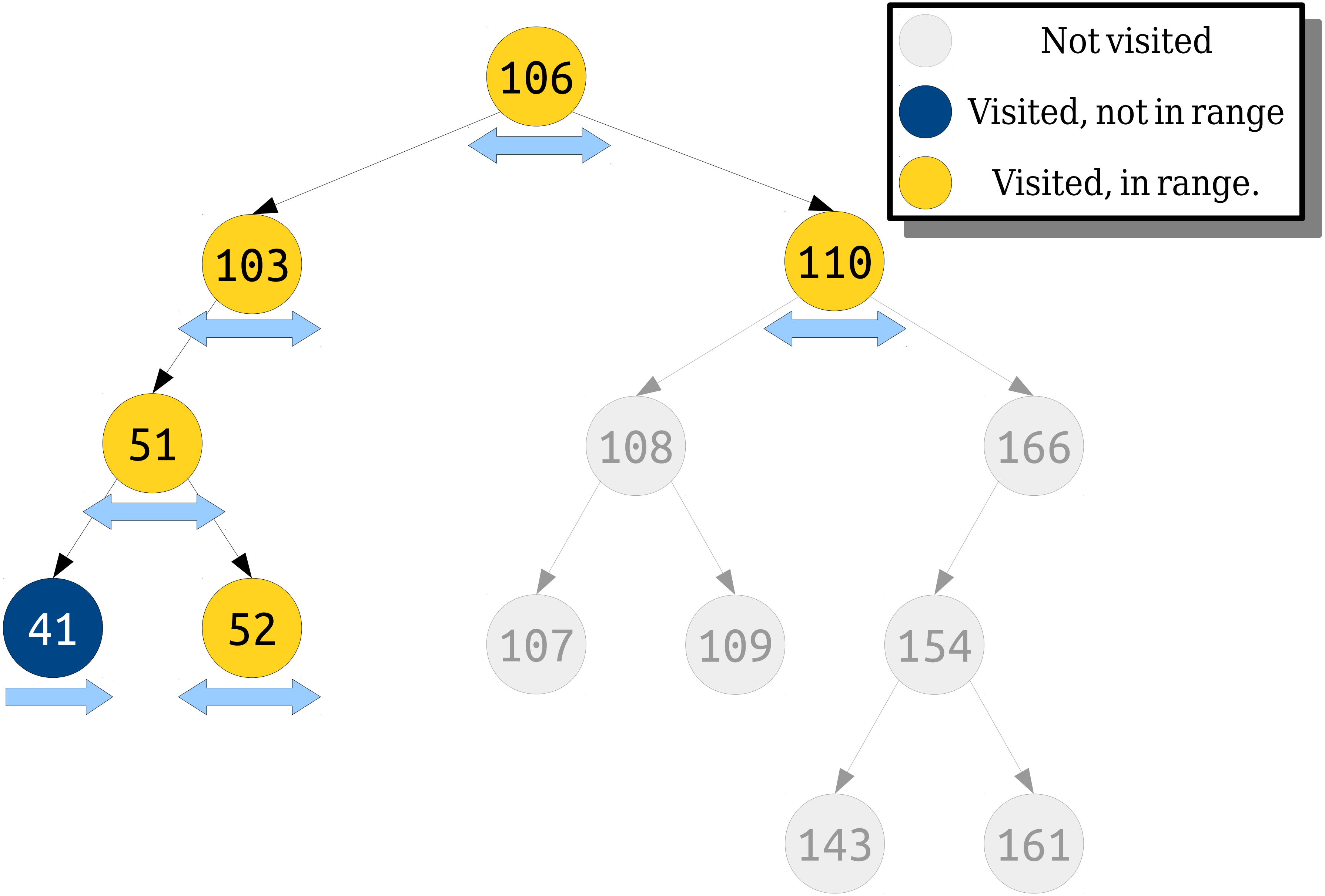
Find all elements in this tree in the range **[42, 165]**.



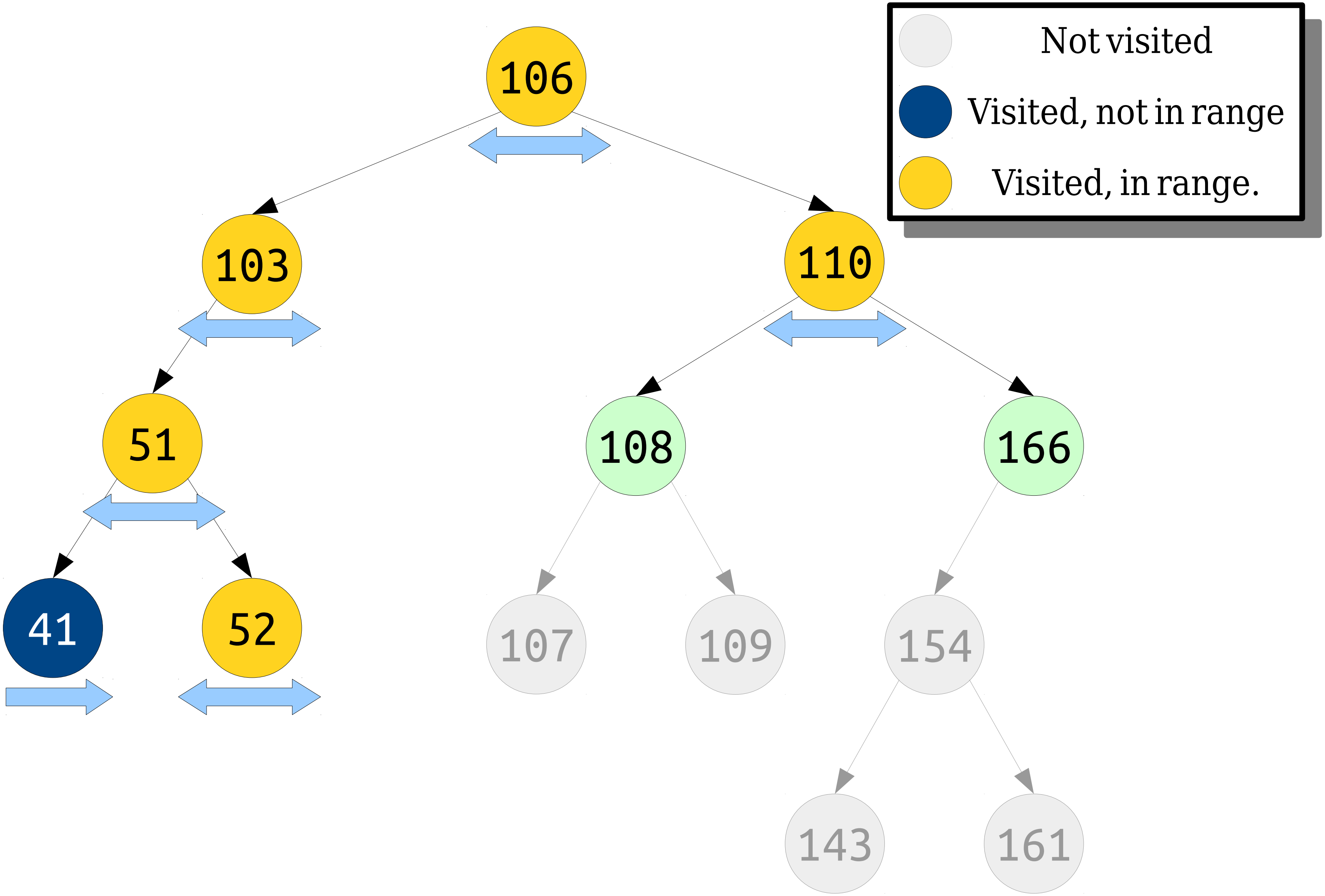
Find all elements in this tree in the range **[42, 165]**.



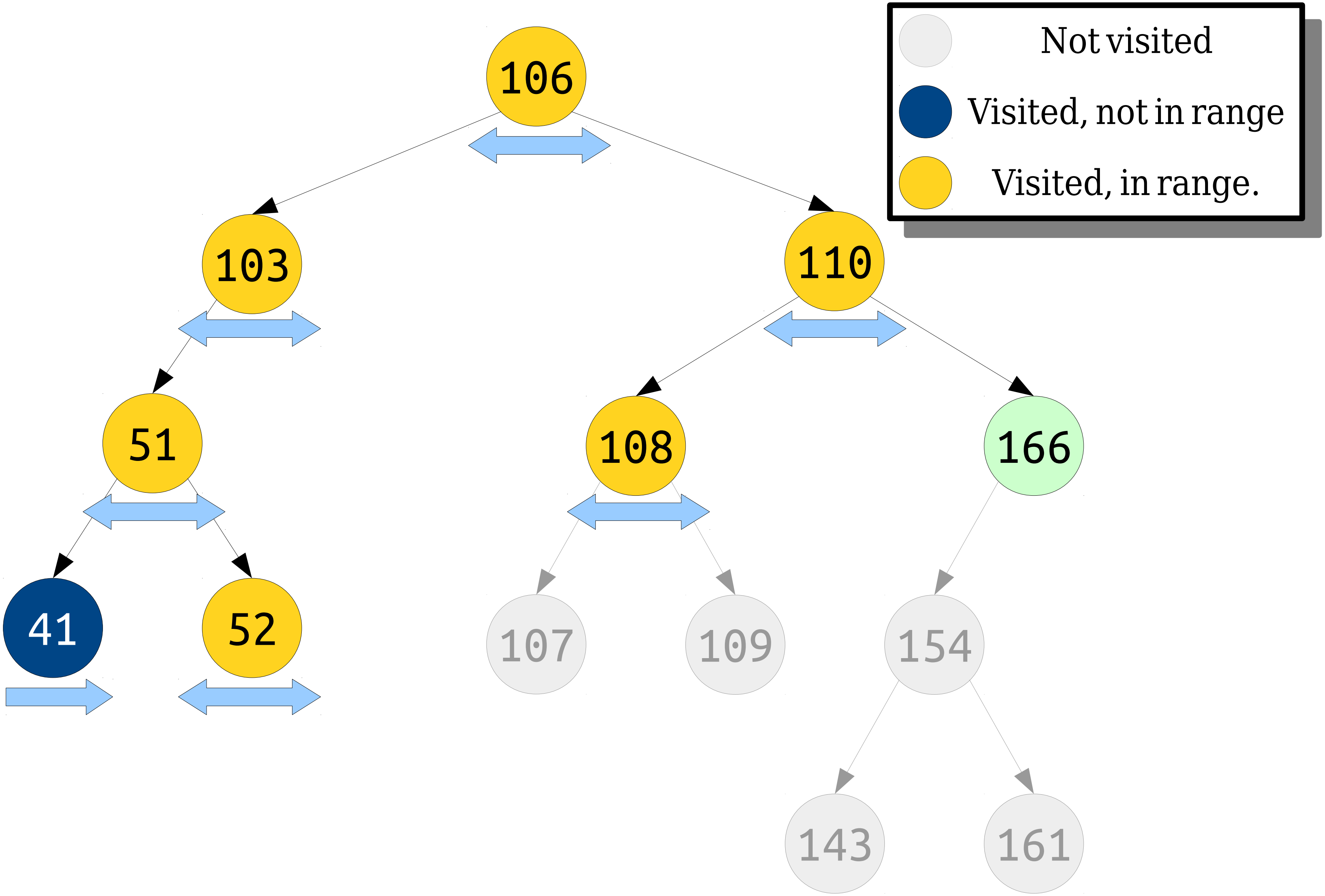
Find all elements in this tree in the range **[42, 165]**.



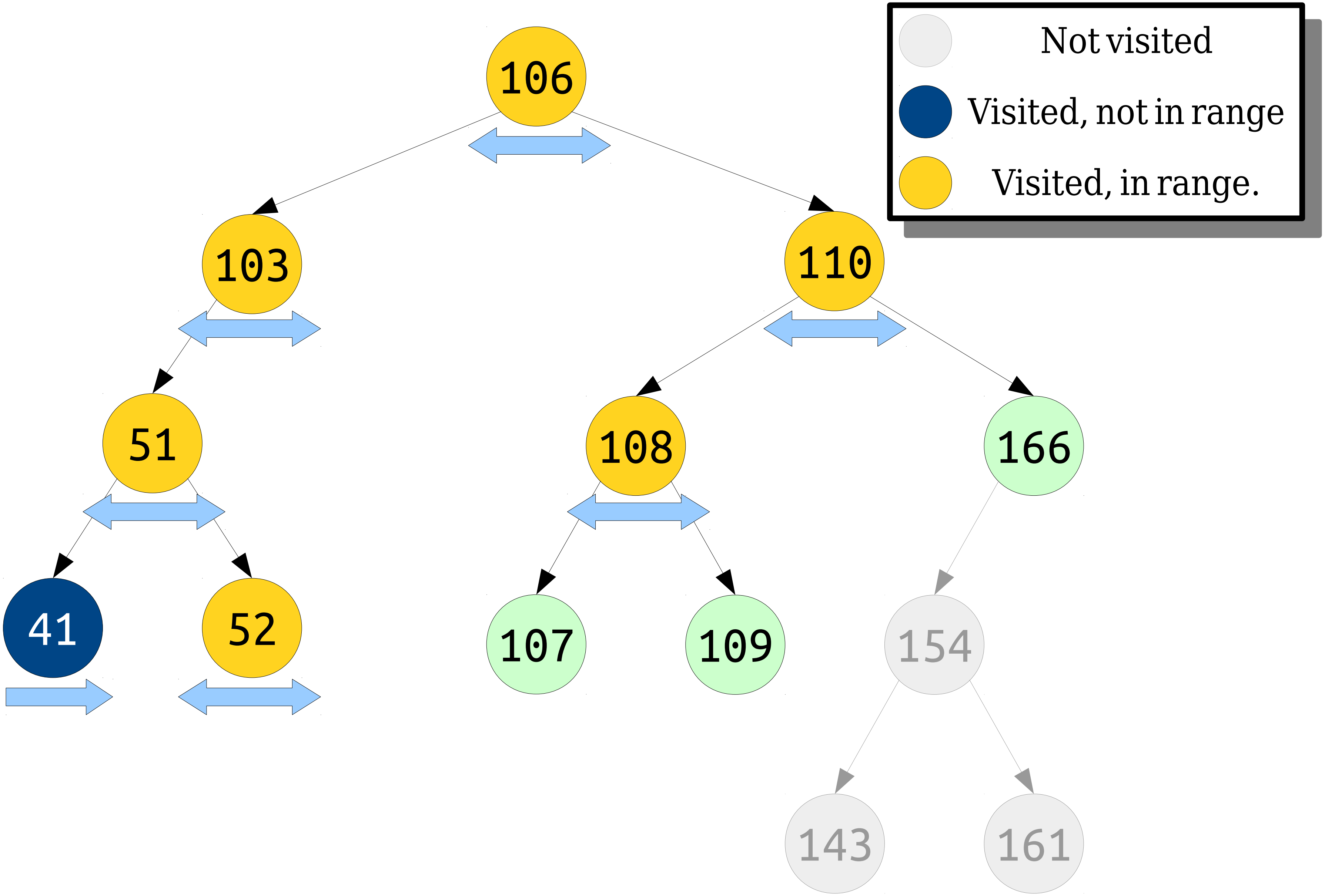
Find all elements in this tree in the range **[42, 165]**.



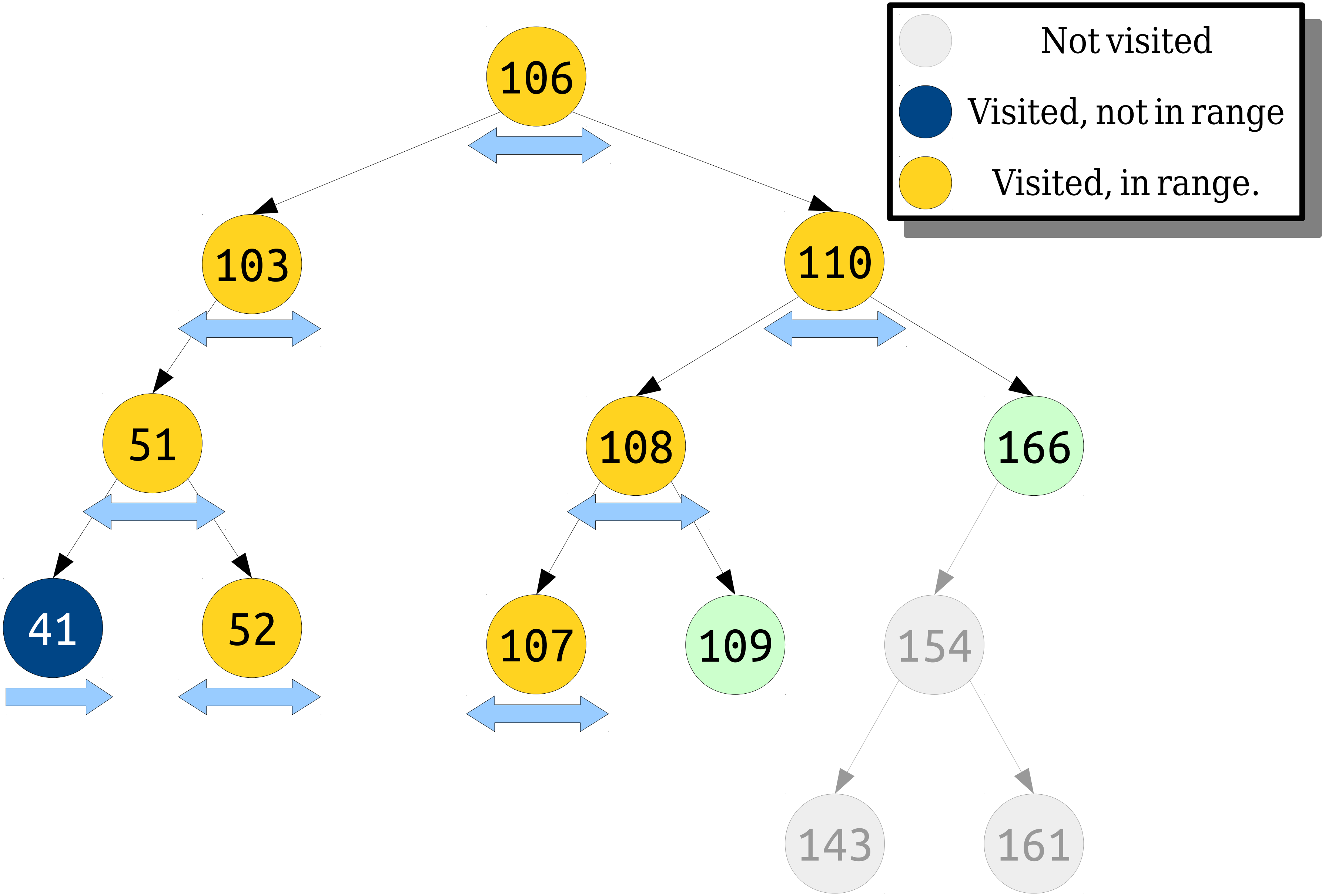
Find all elements in this tree in the range **[42, 165]**.



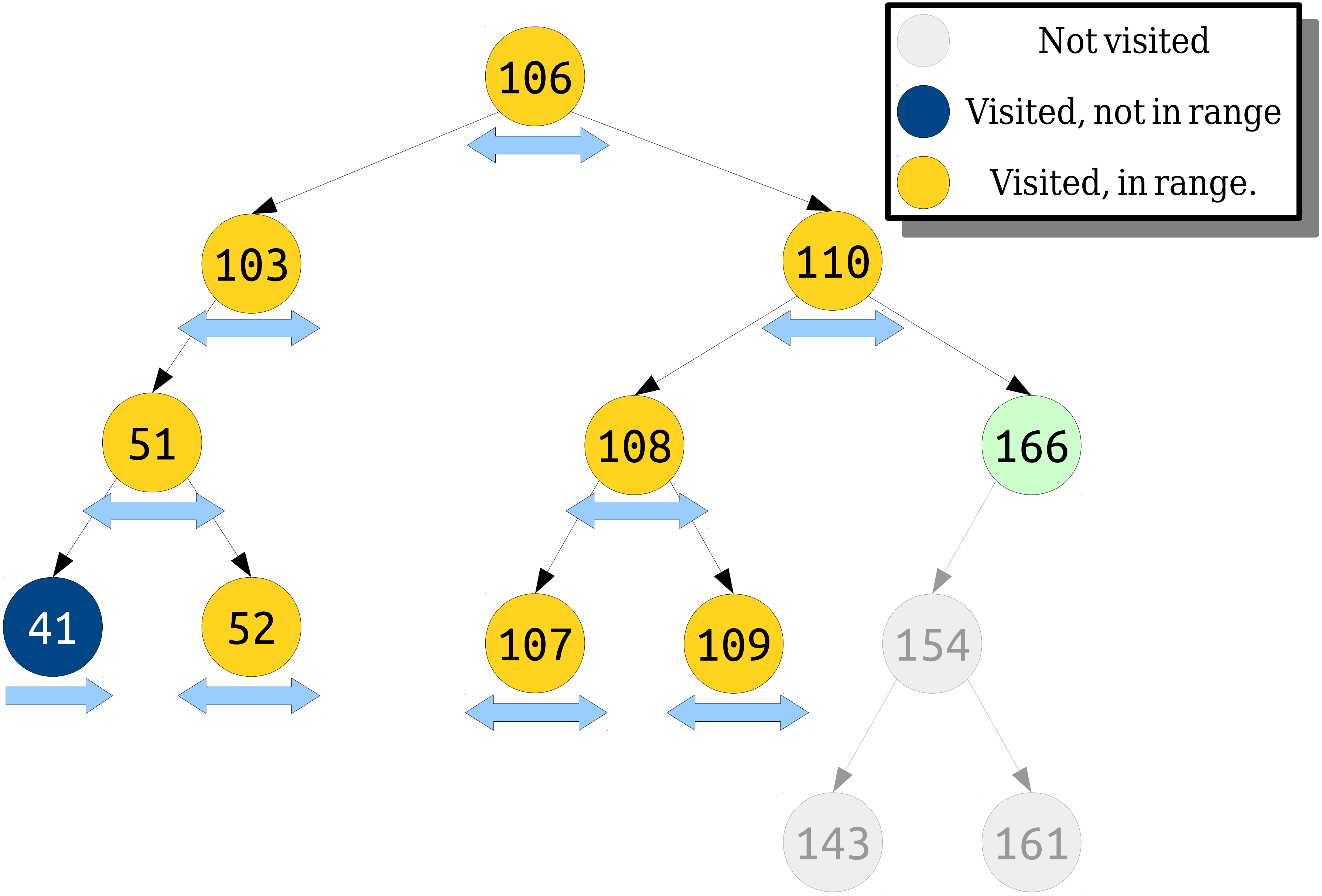
Find all elements in this tree in the range **[42, 165]**.



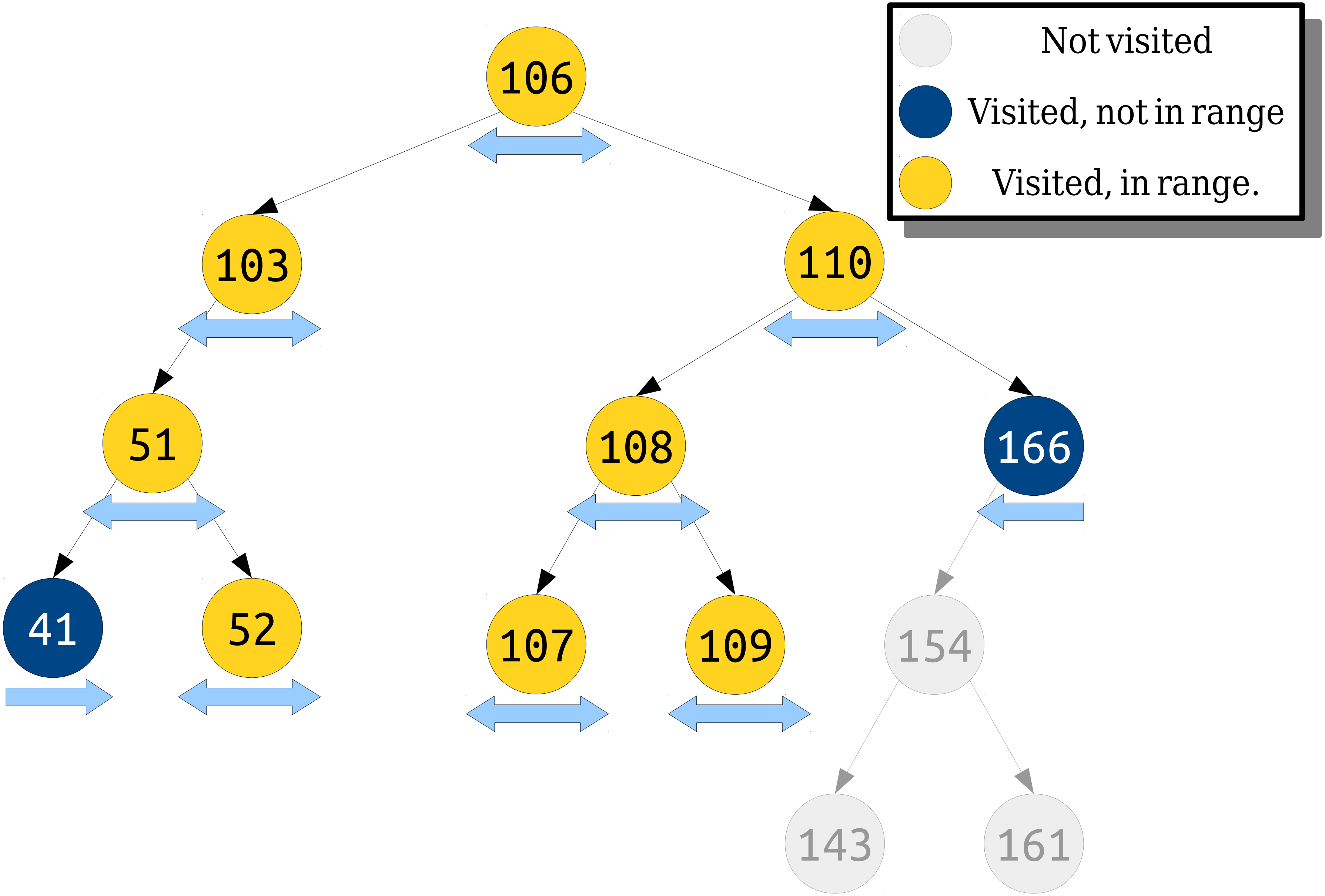
Find all elements in this tree in the range **[42, 165]**.



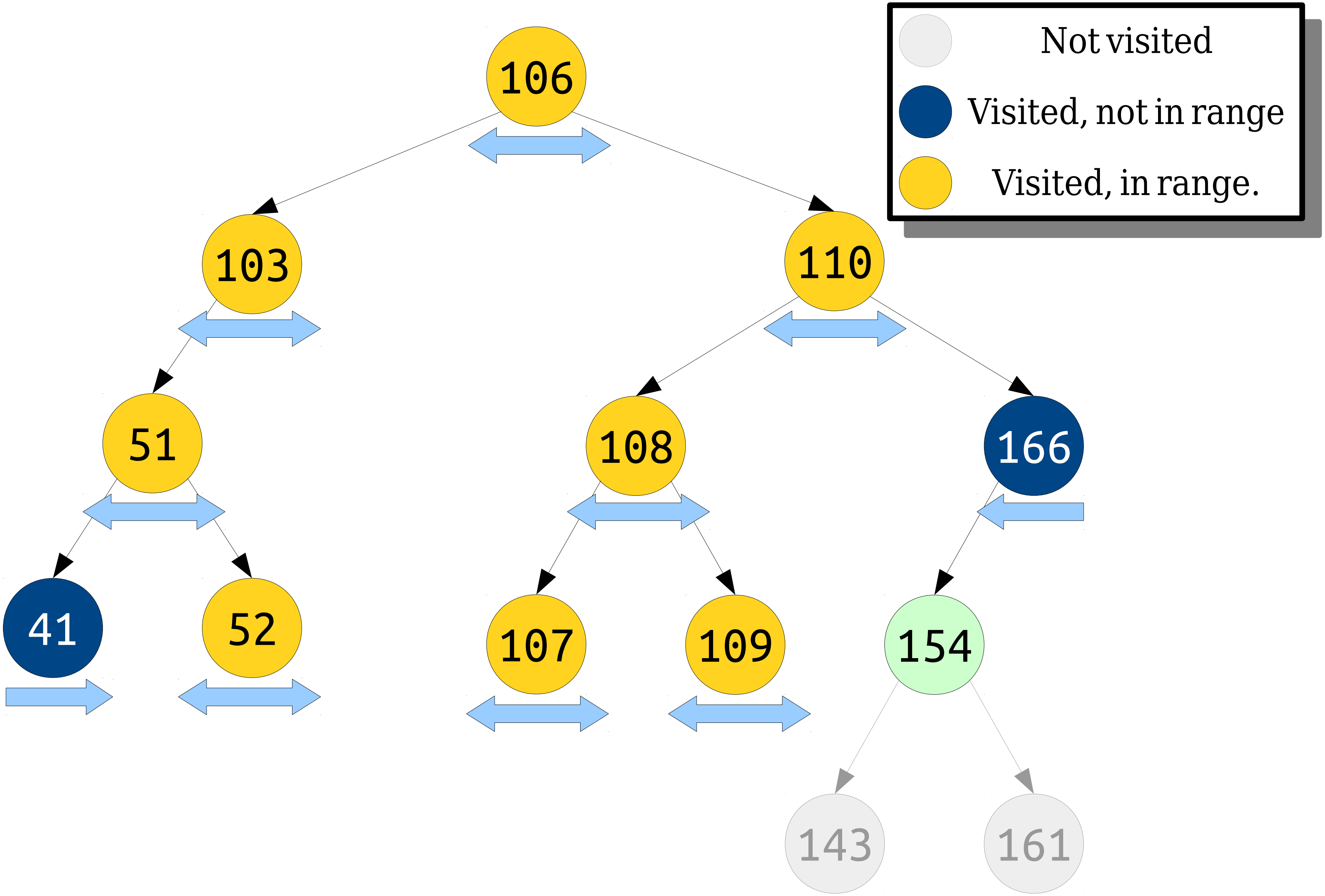
Find all elements in this tree in the range **[42, 165]**.



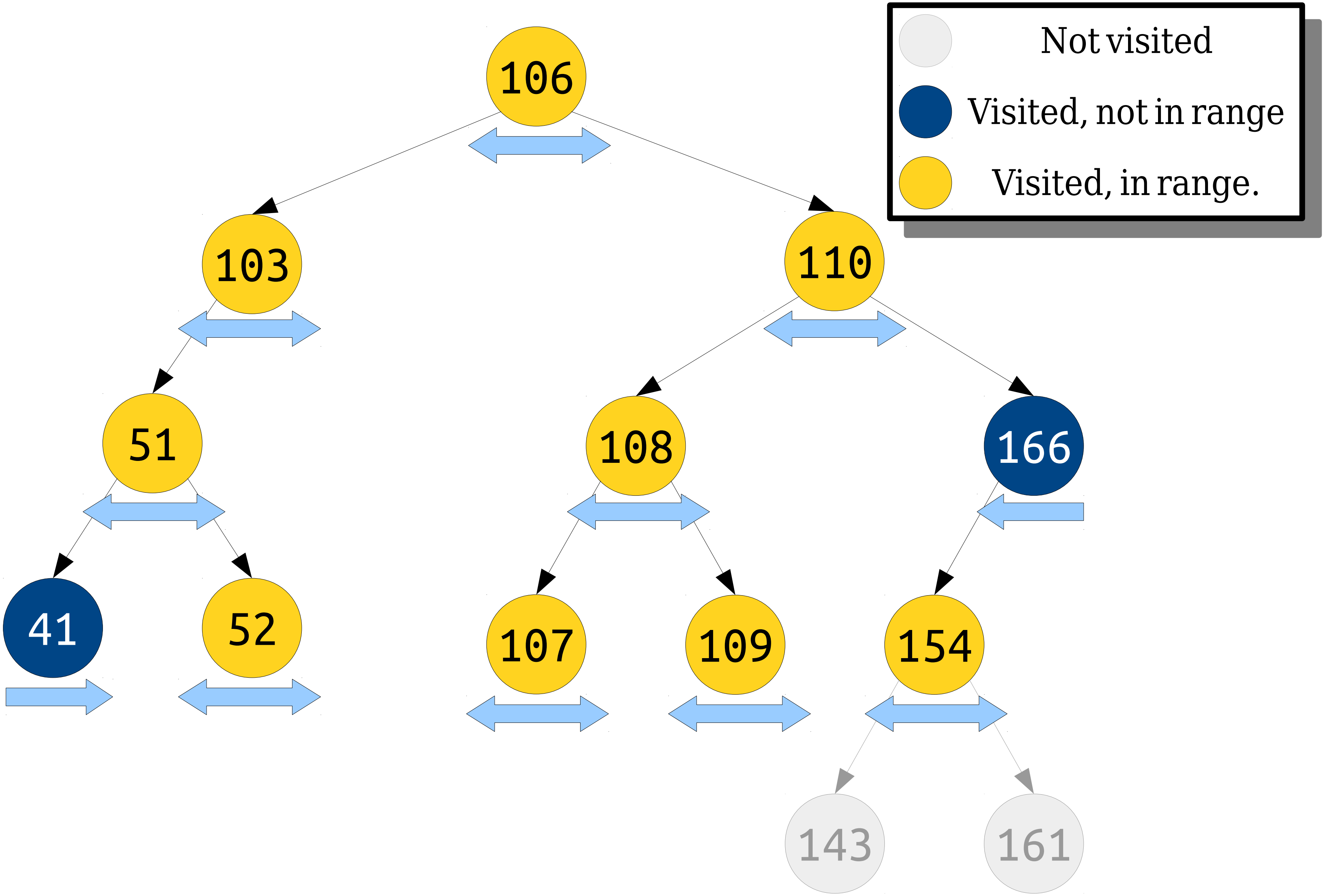
Find all elements in this tree in the range **[42, 165]**.



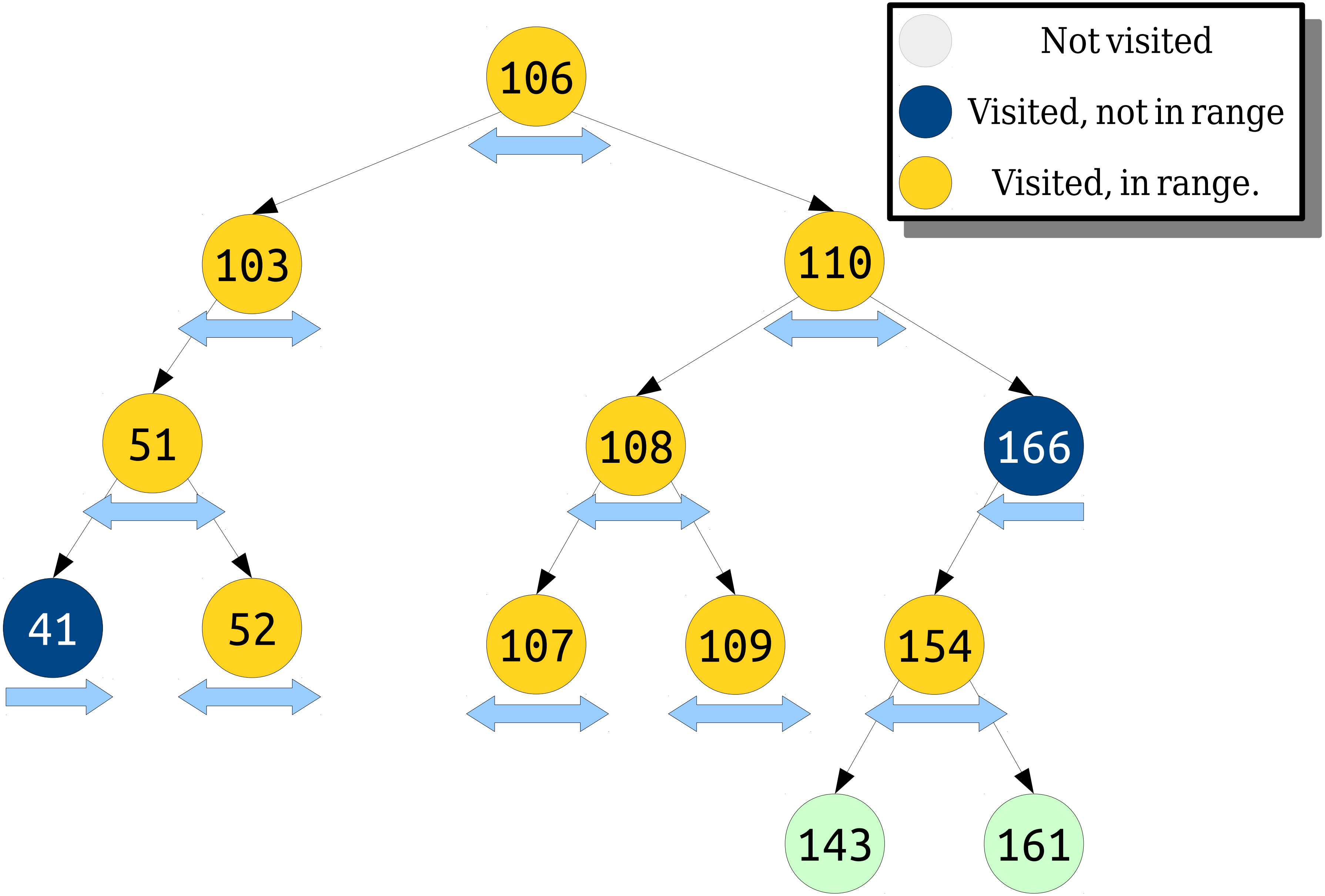
Find all elements in this tree in the range **[42, 165]**.



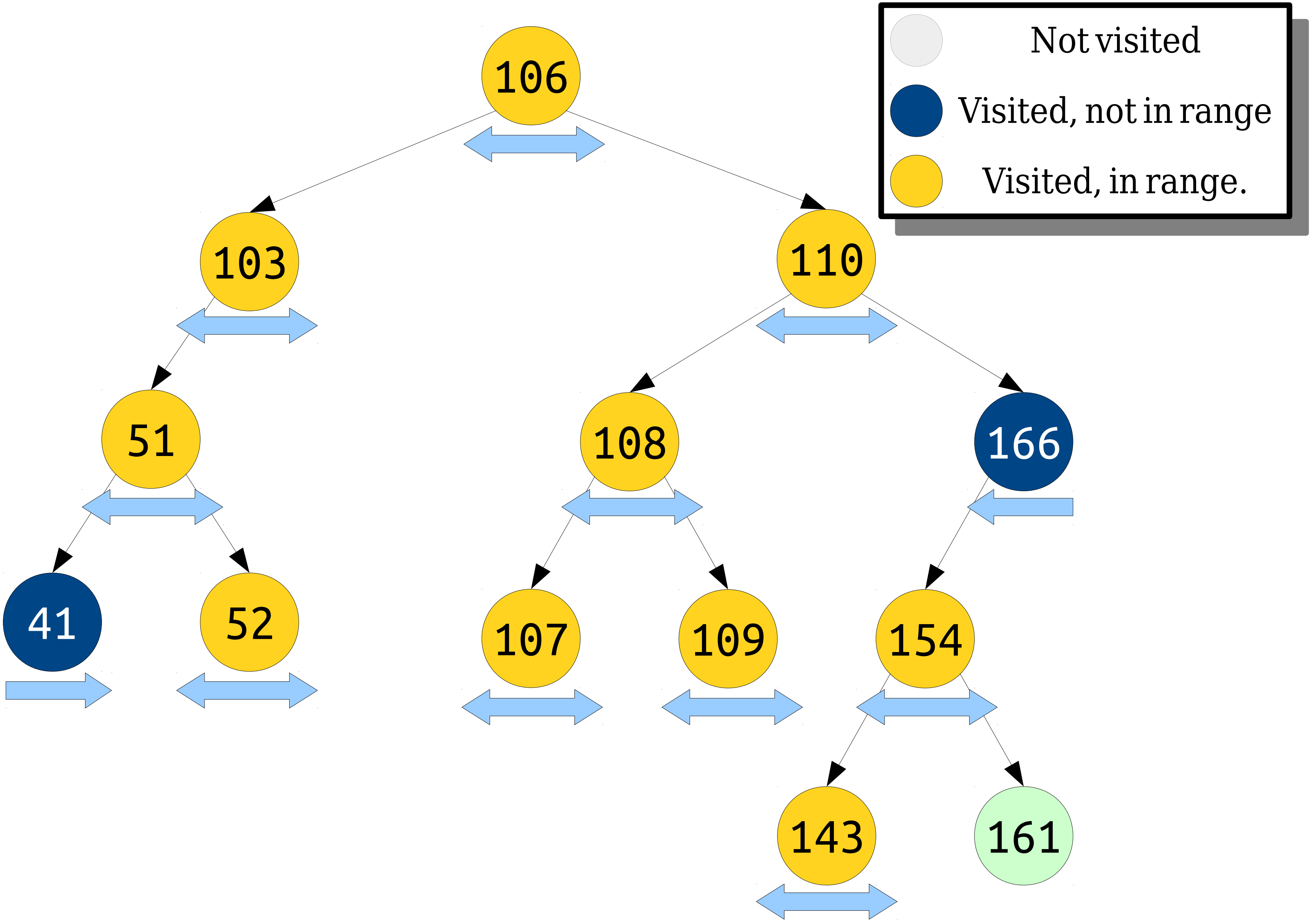
Find all elements in this tree in the range **[42, 165]**.



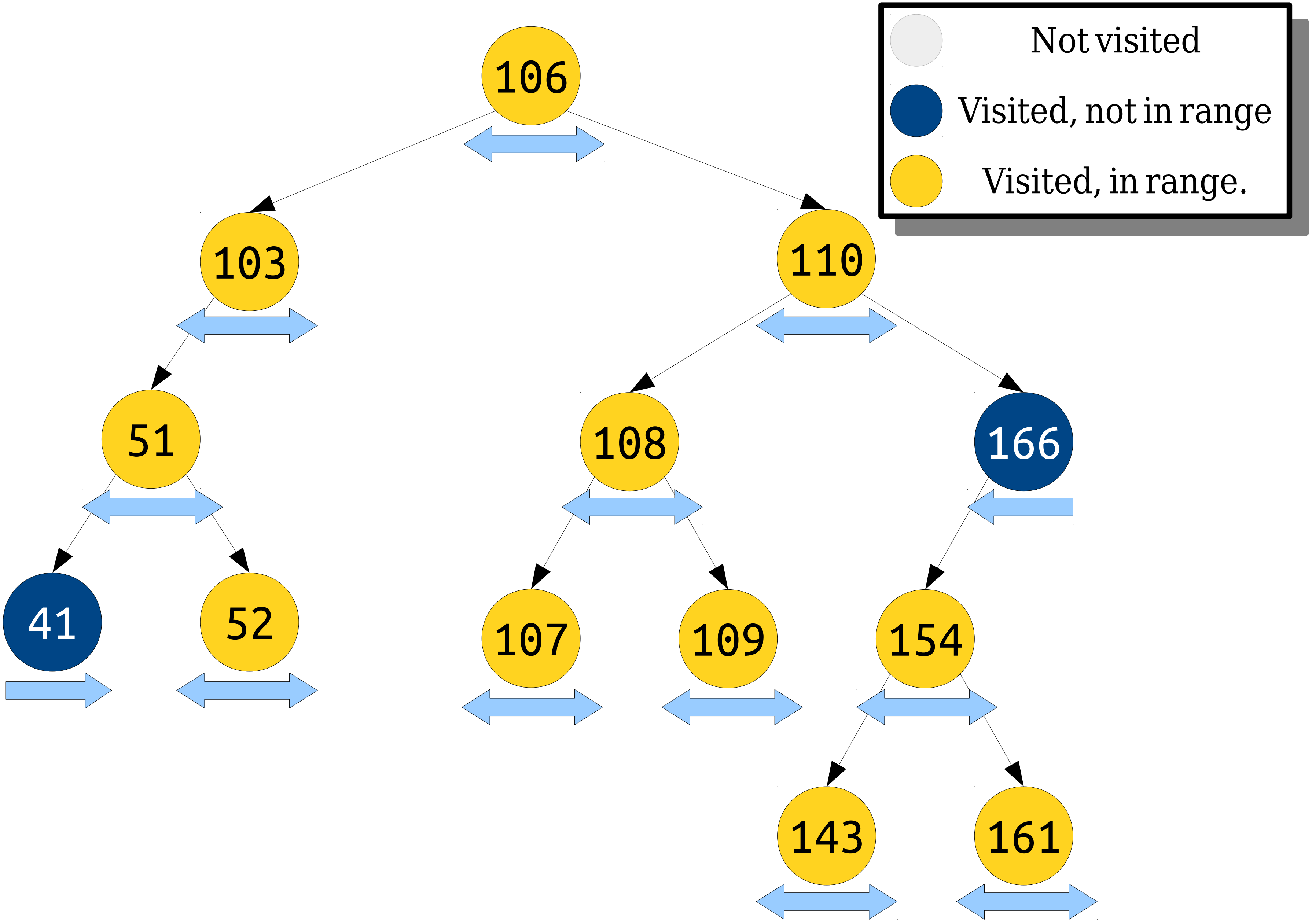
Find all elements in this tree in the range **[42, 165]**.



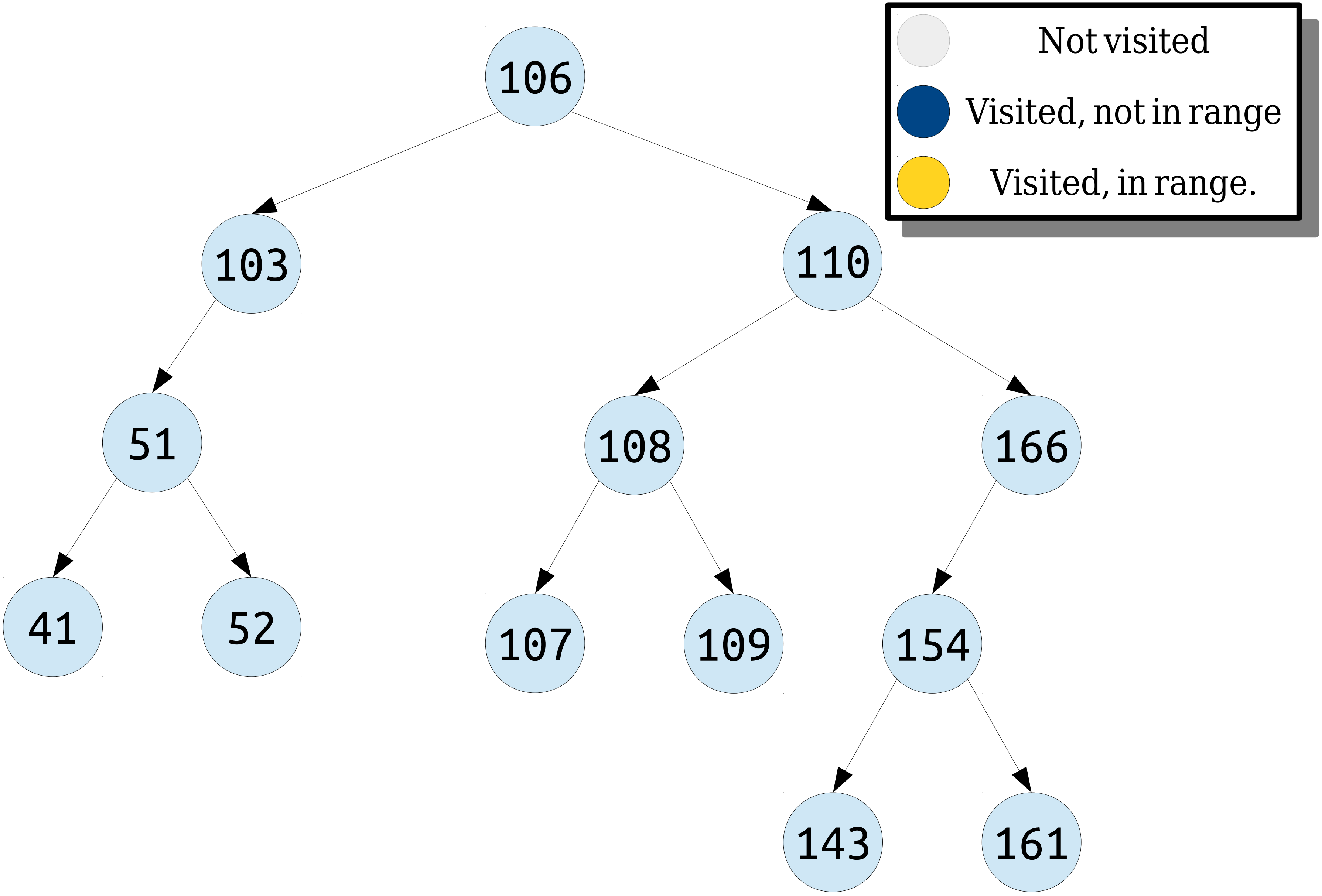
Find all elements in this tree in the range **[42, 165]**.



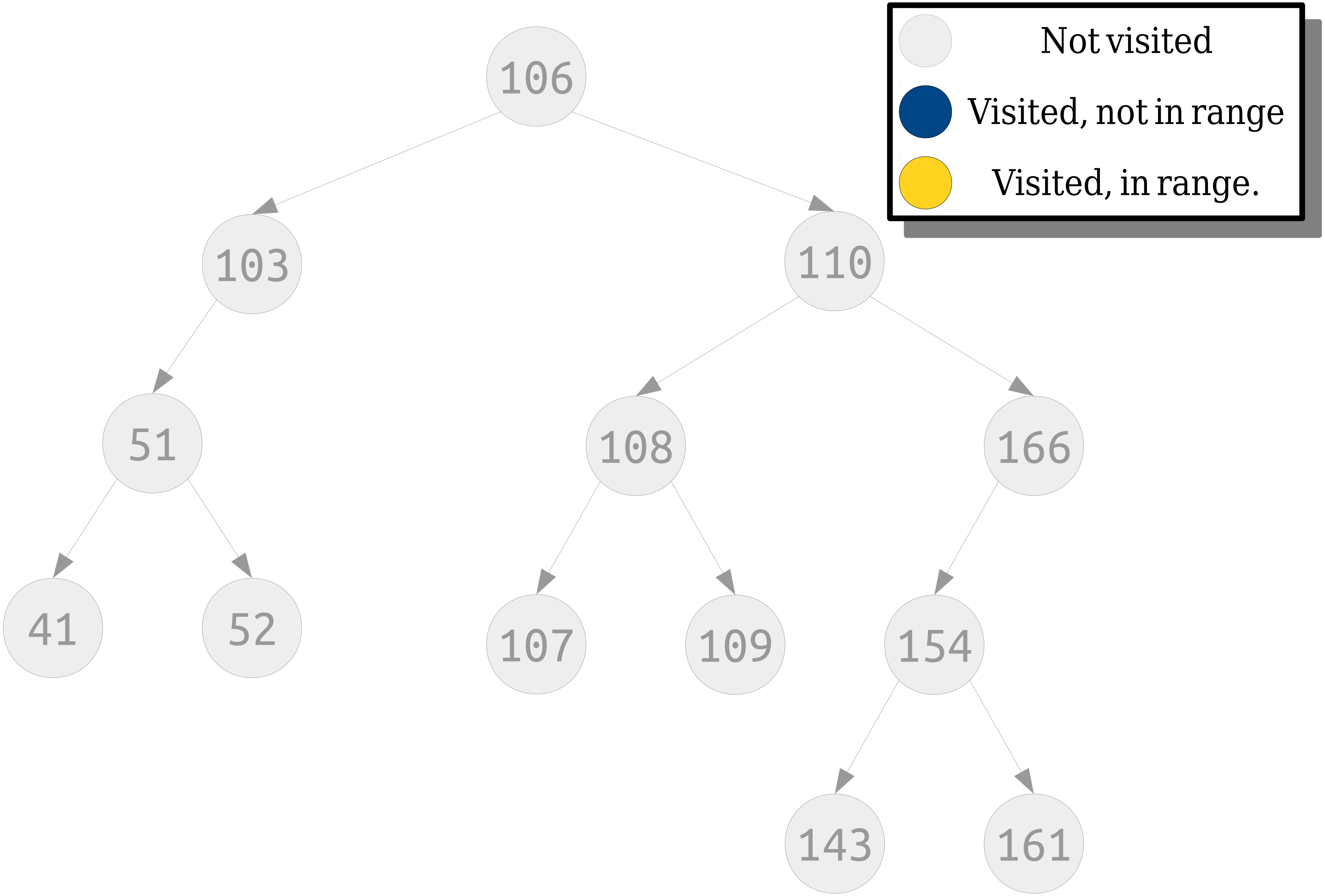
Find all elements in this tree in the range **[42, 165]**.



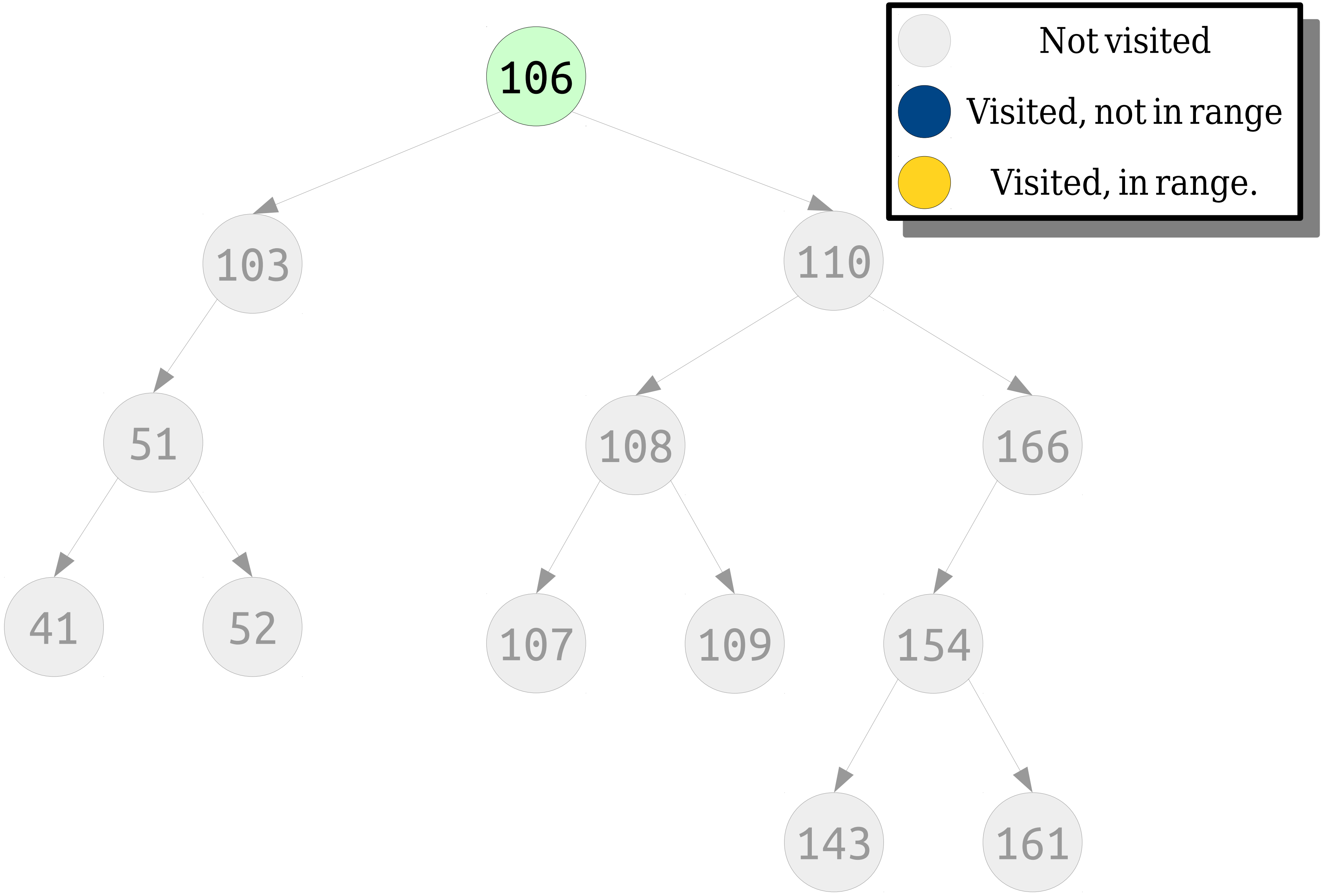
Find all elements in this tree in the range **[42, 165]**.



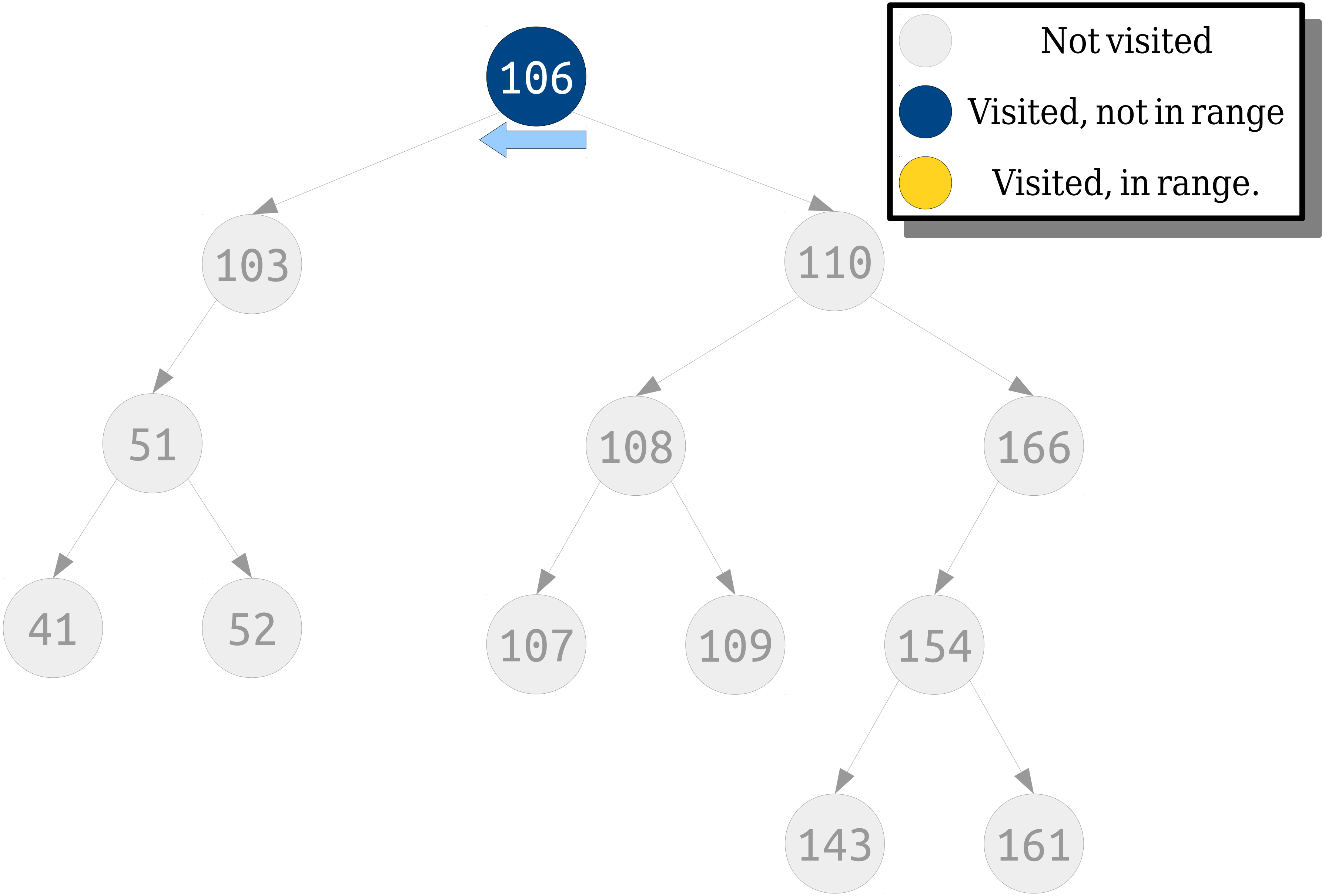
Find all elements in this tree in the range **[49, 50]**.



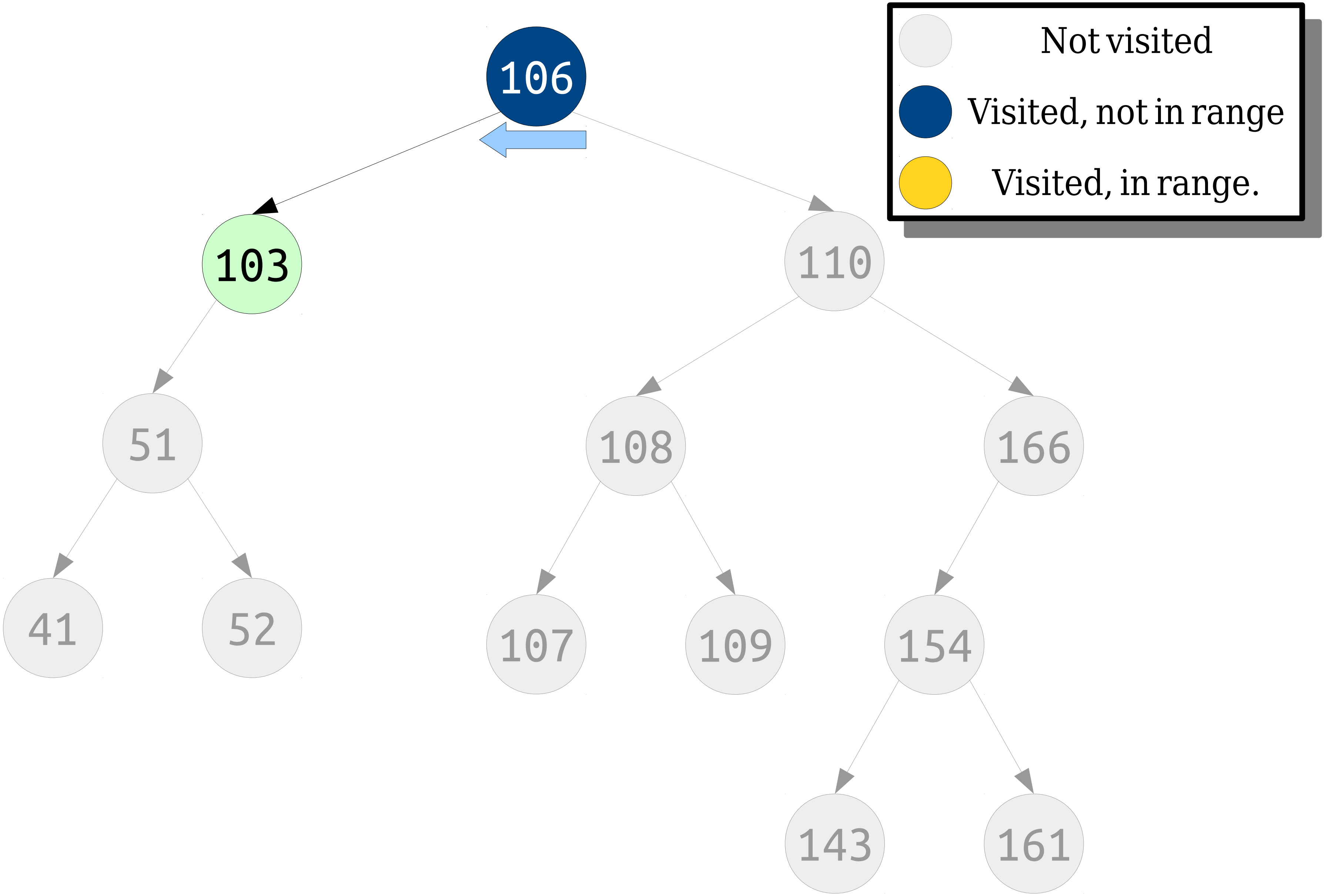
Find all elements in this tree in the range **[49, 50]**.



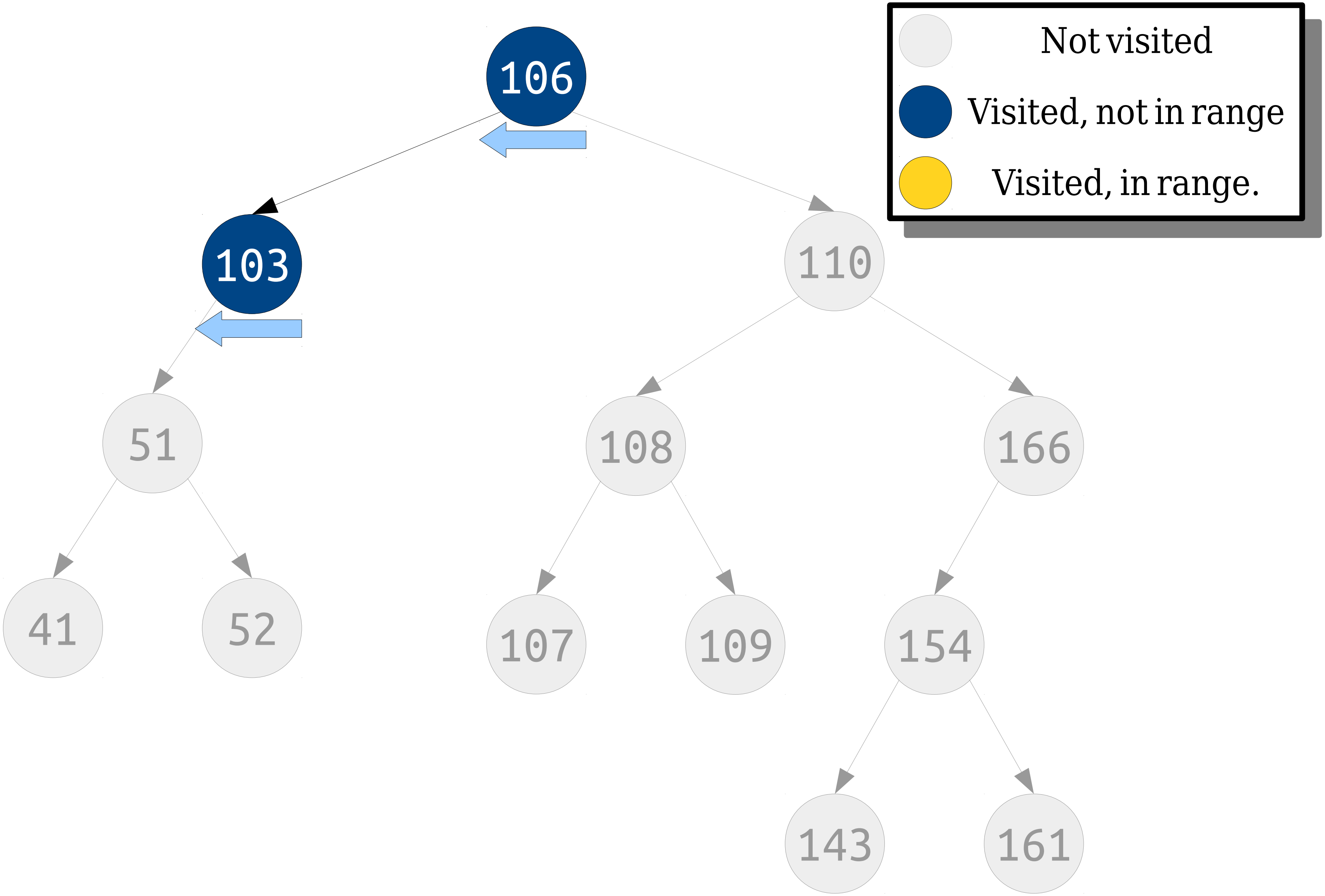
Find all elements in this tree in the range **[49, 50]**.



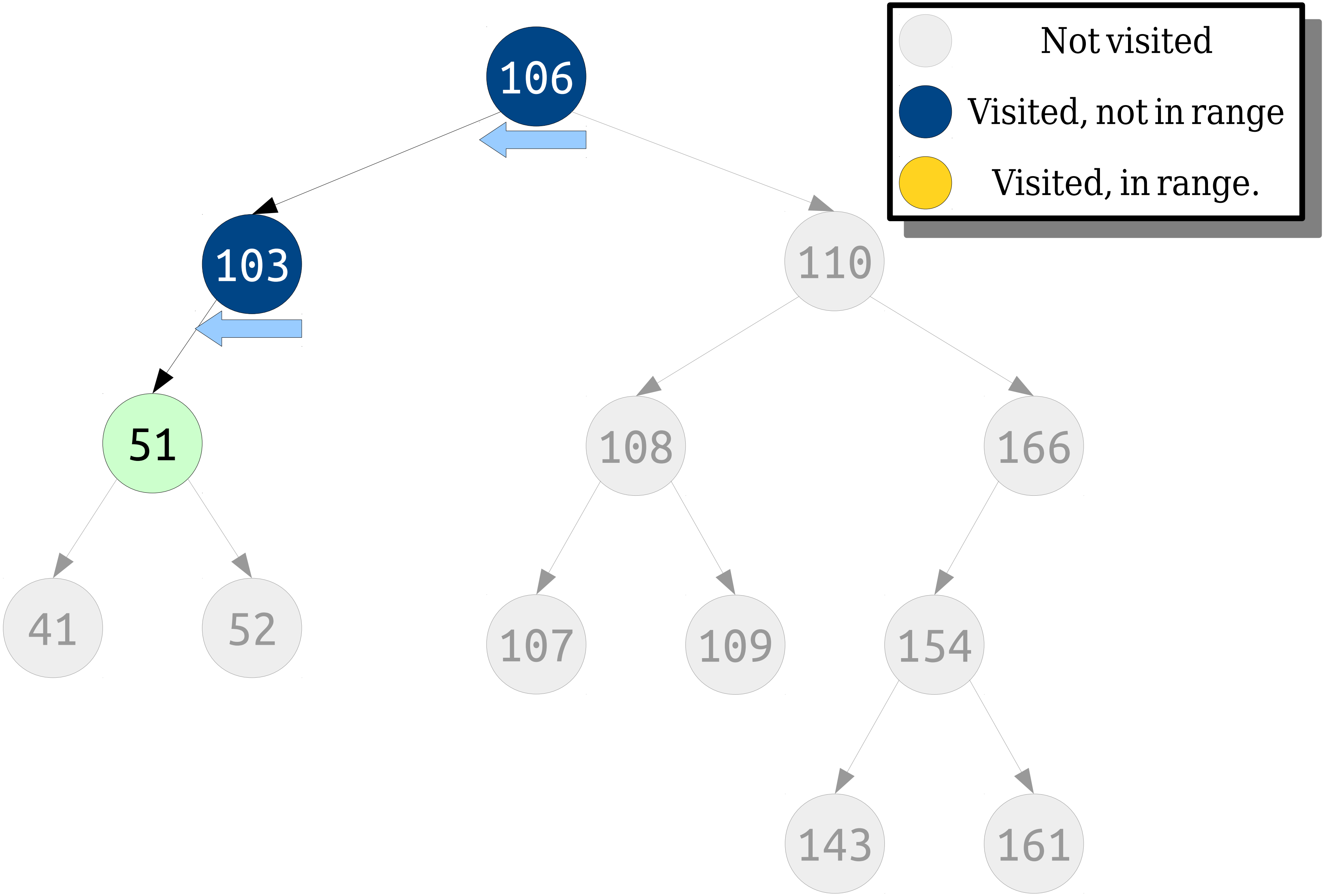
Find all elements in this tree in the range **[49, 50]**.



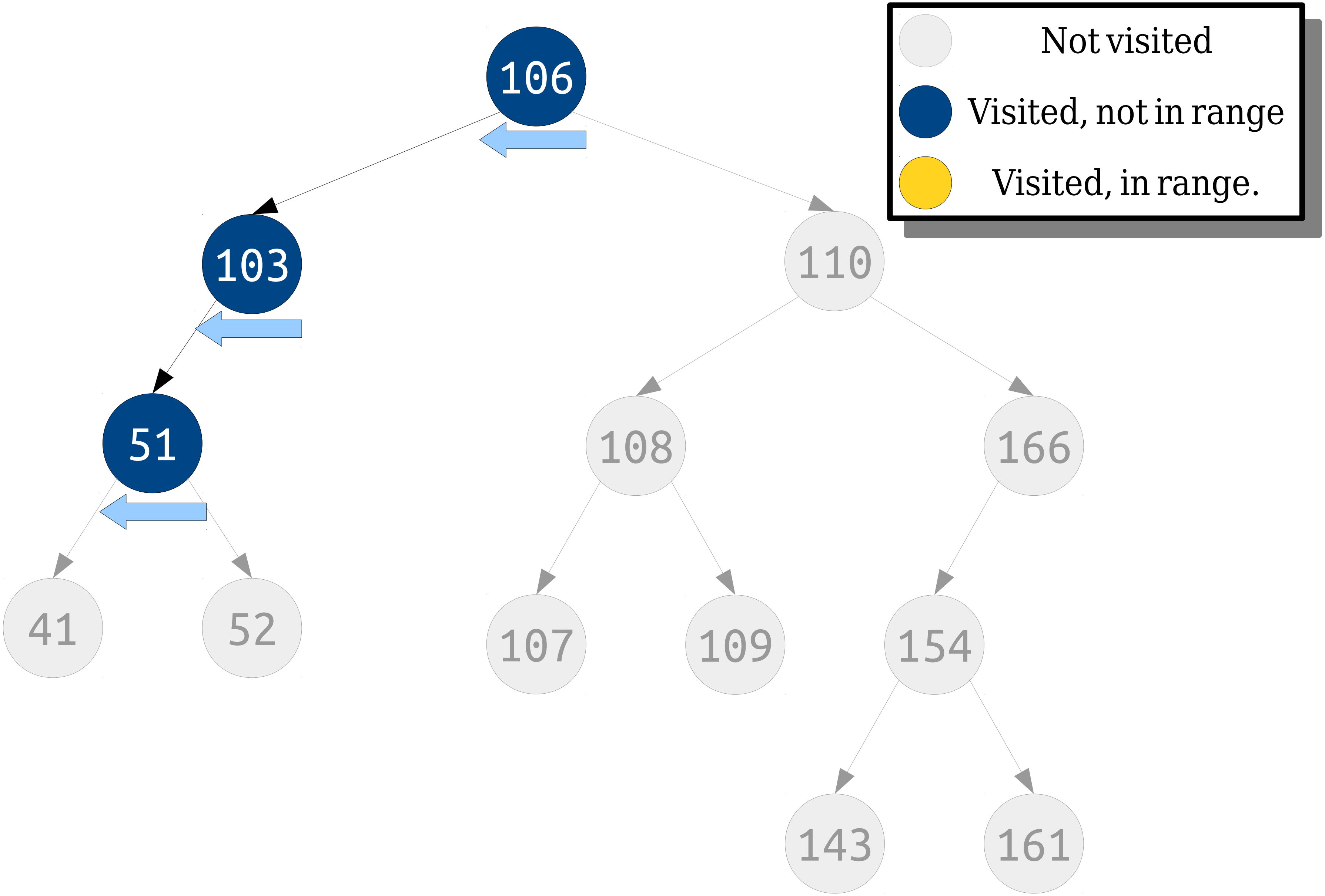
Find all elements in this tree in the range **[49, 50]**.



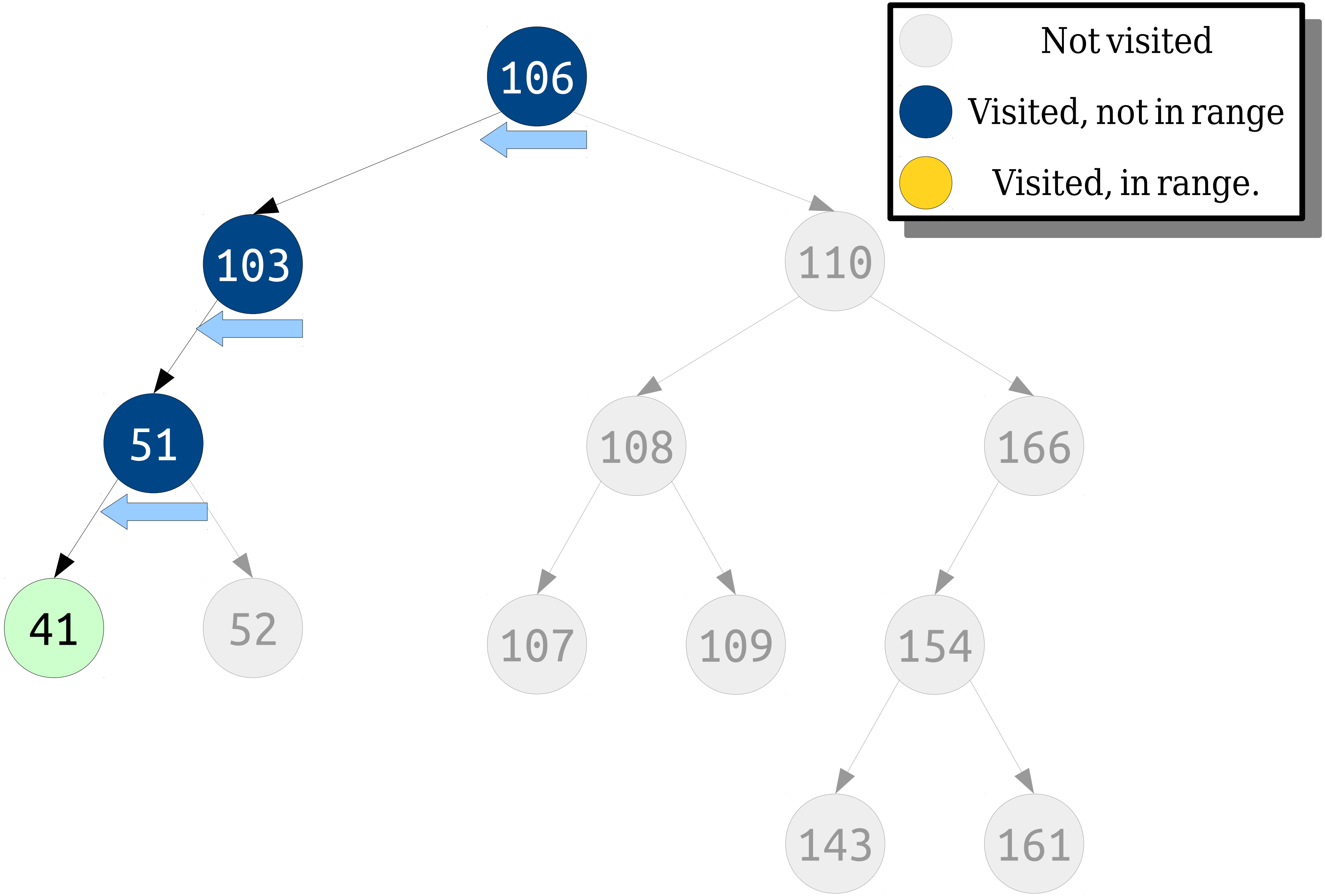
Find all elements in this tree in the range **[49, 50]**.



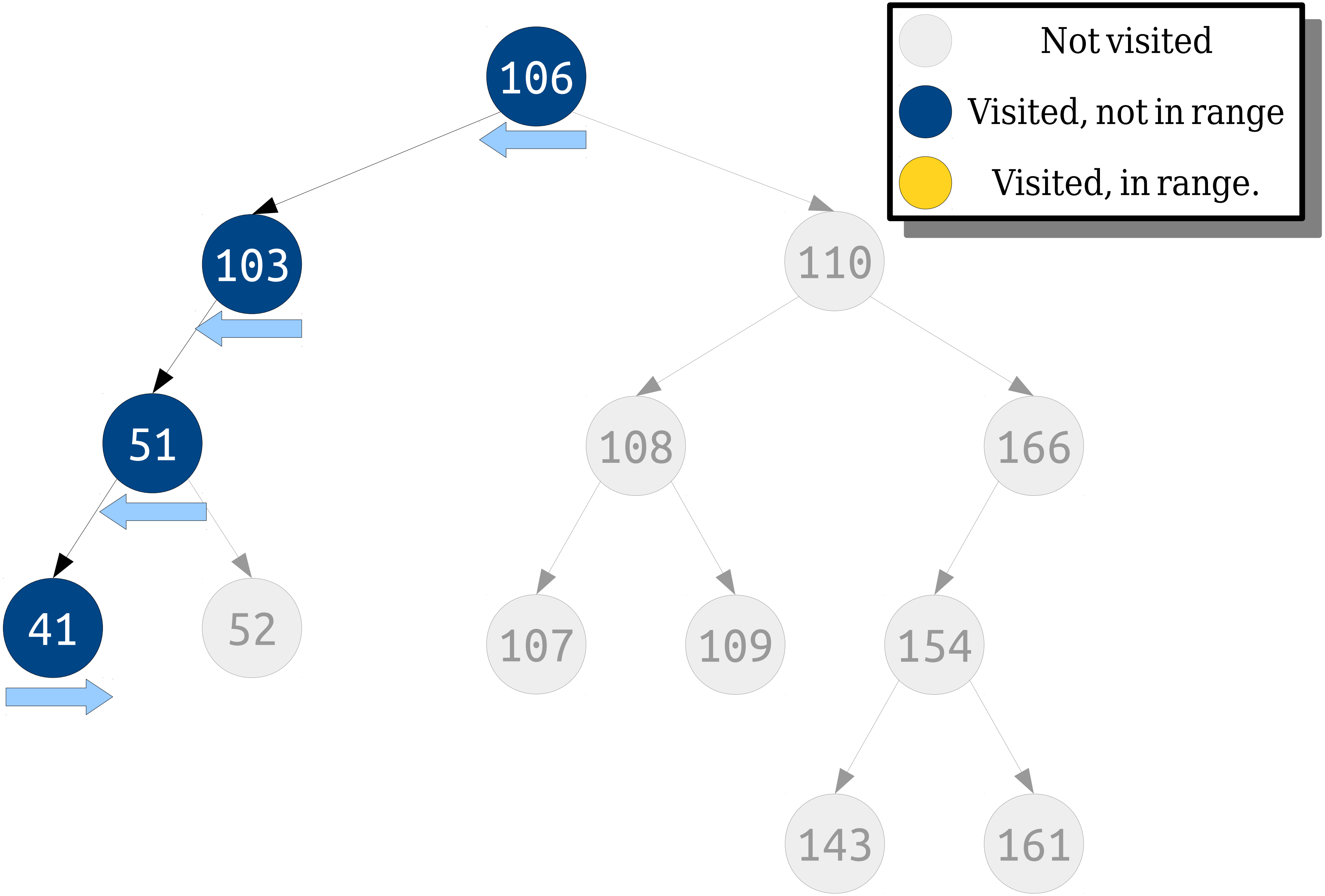
Find all elements in this tree in the range **[49, 50]**.



Find all elements in this tree in the range **[49, 50]**.



Find all elements in this tree in the range **[49, 50]**.

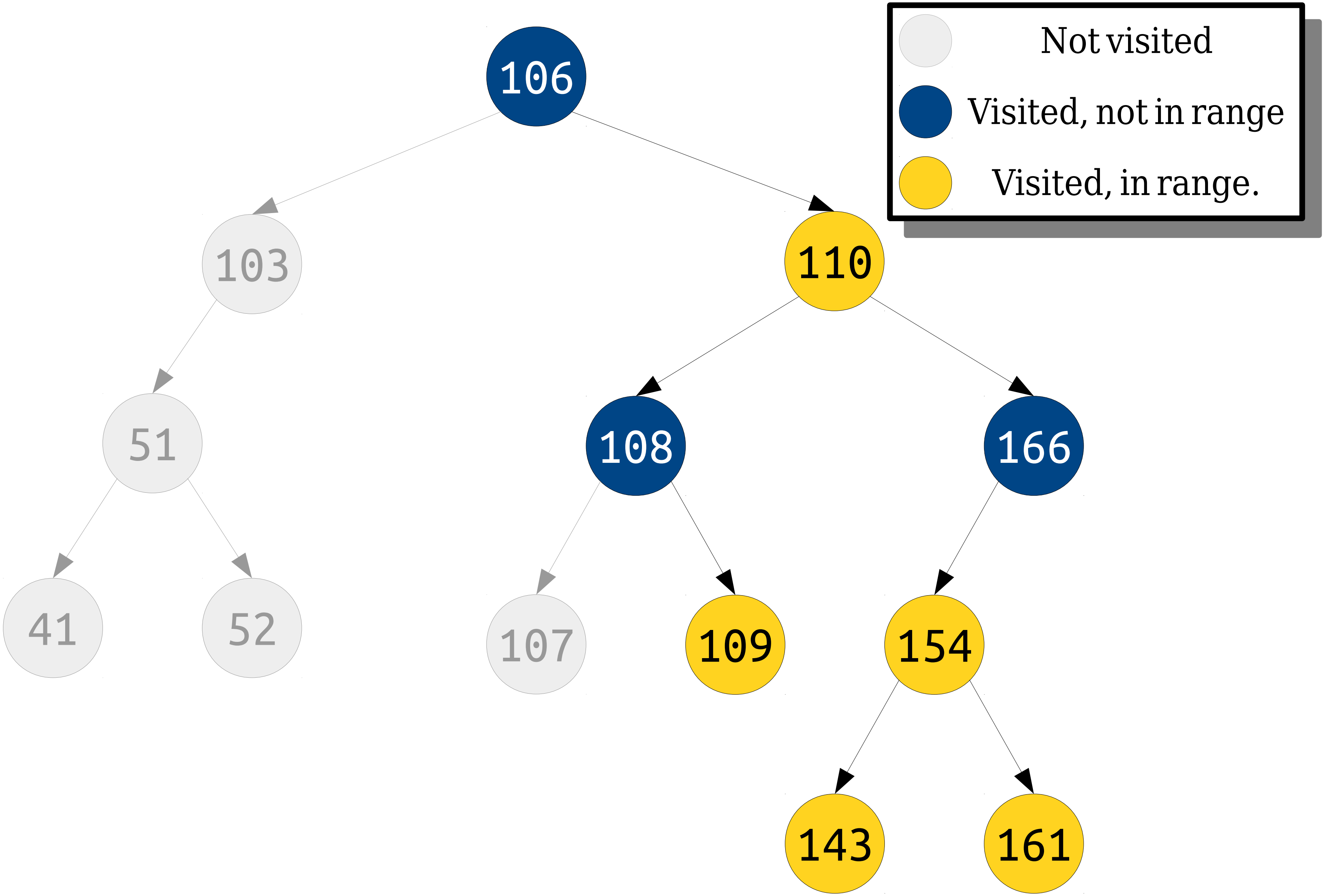


Find all elements in this tree in the range **[49, 50]**.

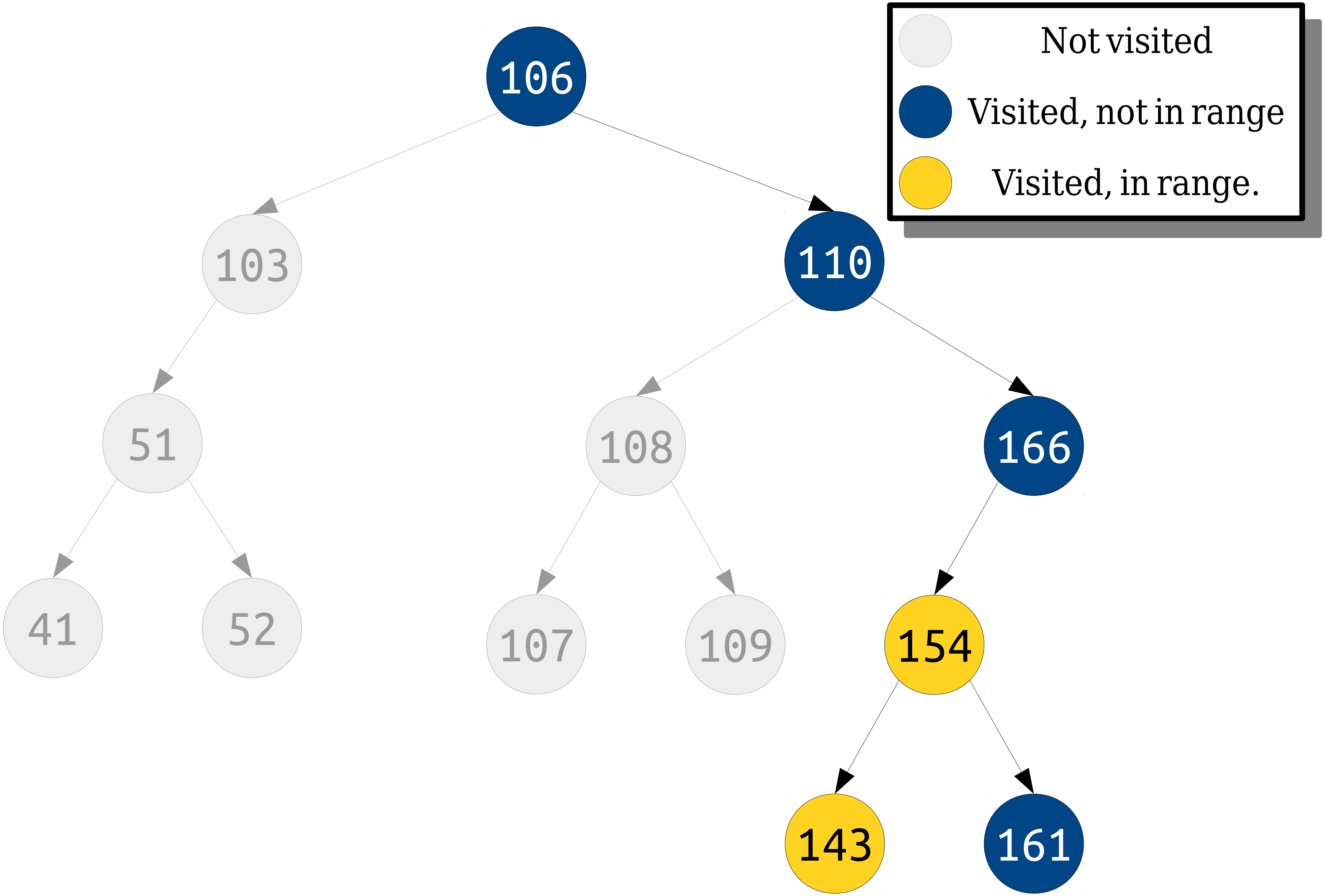
Range Searches

- A hybrid between an inorder traversal and a regular BST lookup!
- The idea:
  - If the node is in the range being searched, add it to the result.
  - Recursively explore each subtree that could potentially overlap with the range.
- ***Question to Ponder:*** Given how an inorder traversal works, how would you code this up if you wanted the values back in sorted order?

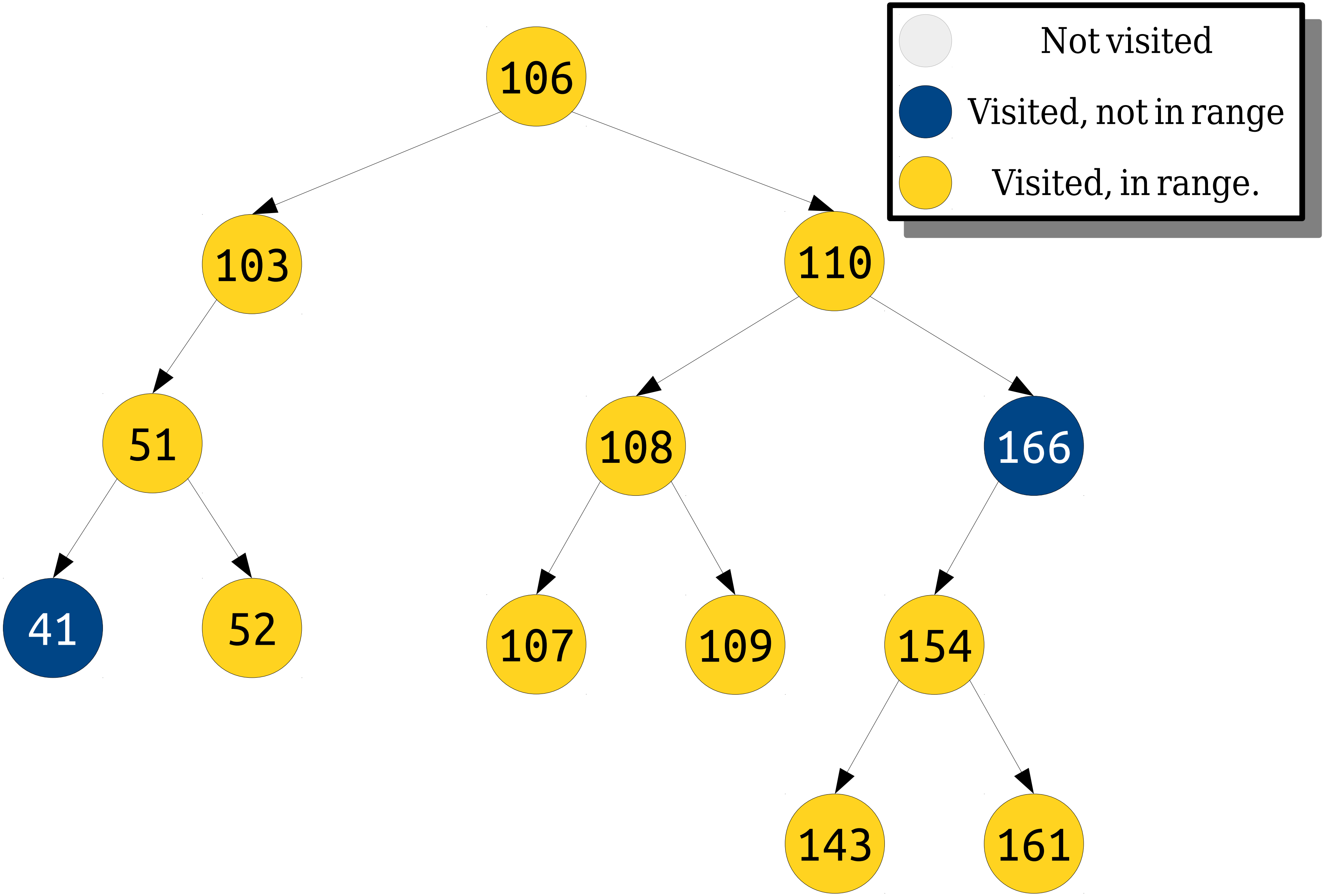
How efficient is a range search?



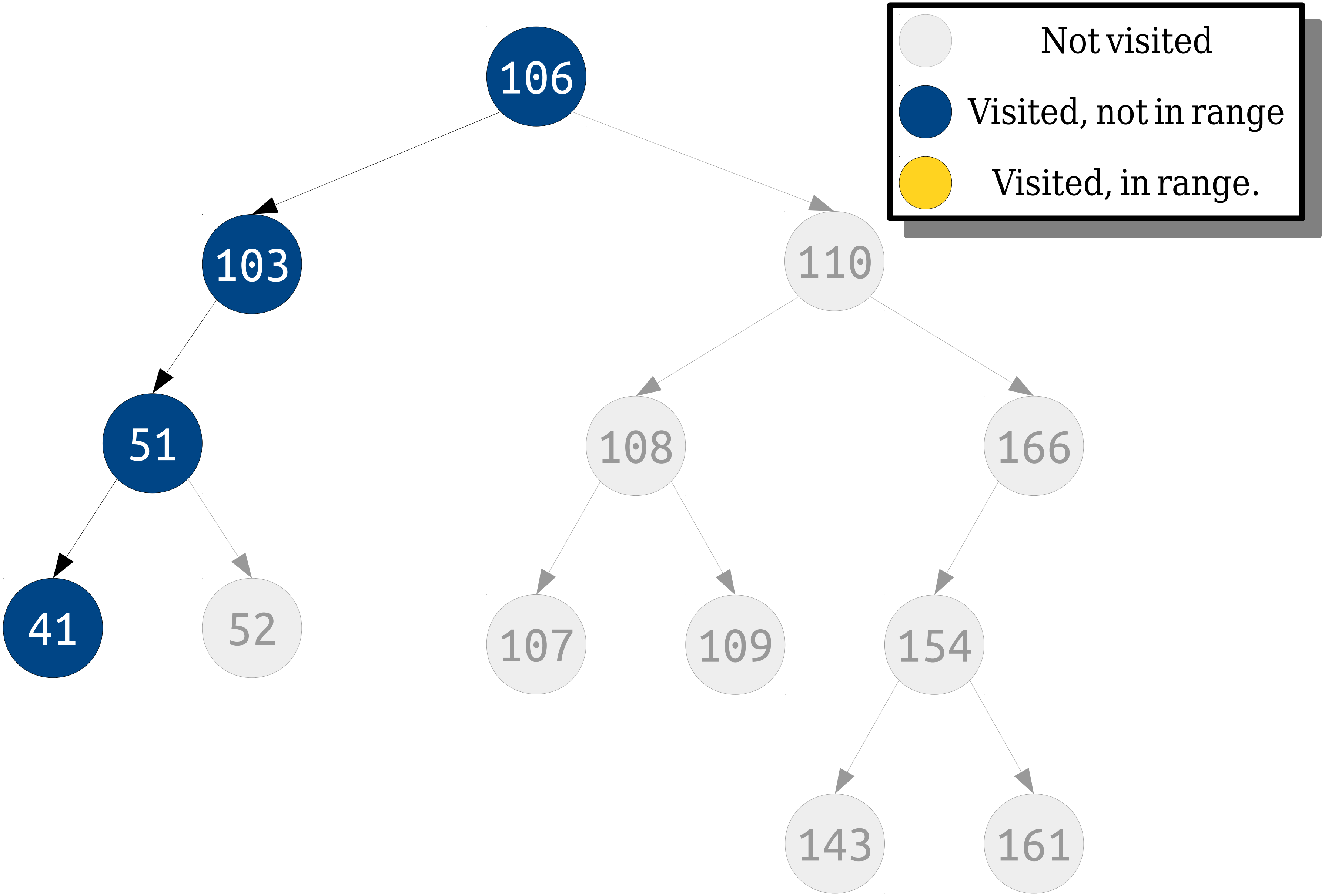
Find all elements in this tree in the range **[109, 163]**.



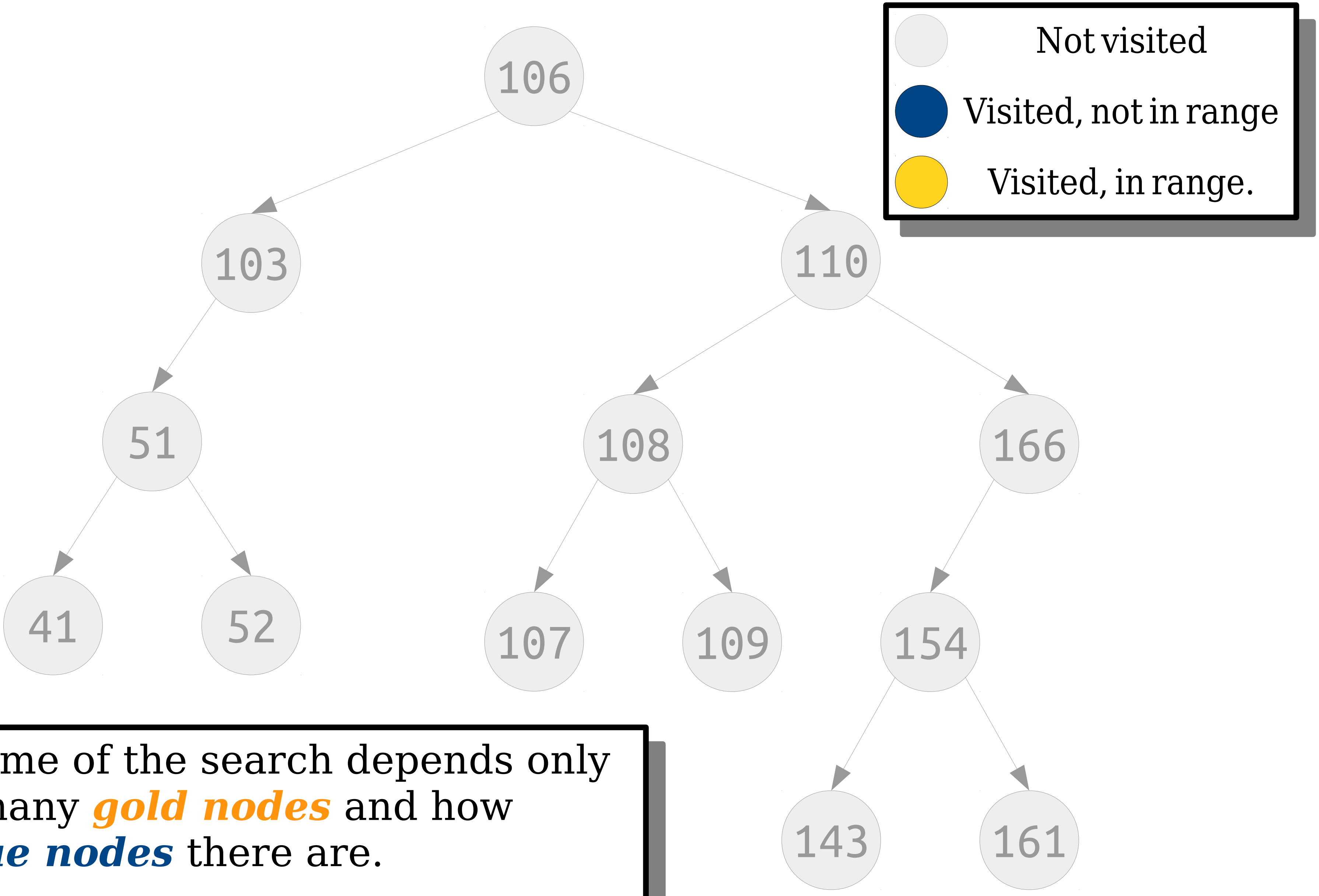
Find all elements in this tree in the range **[124, 155]**.



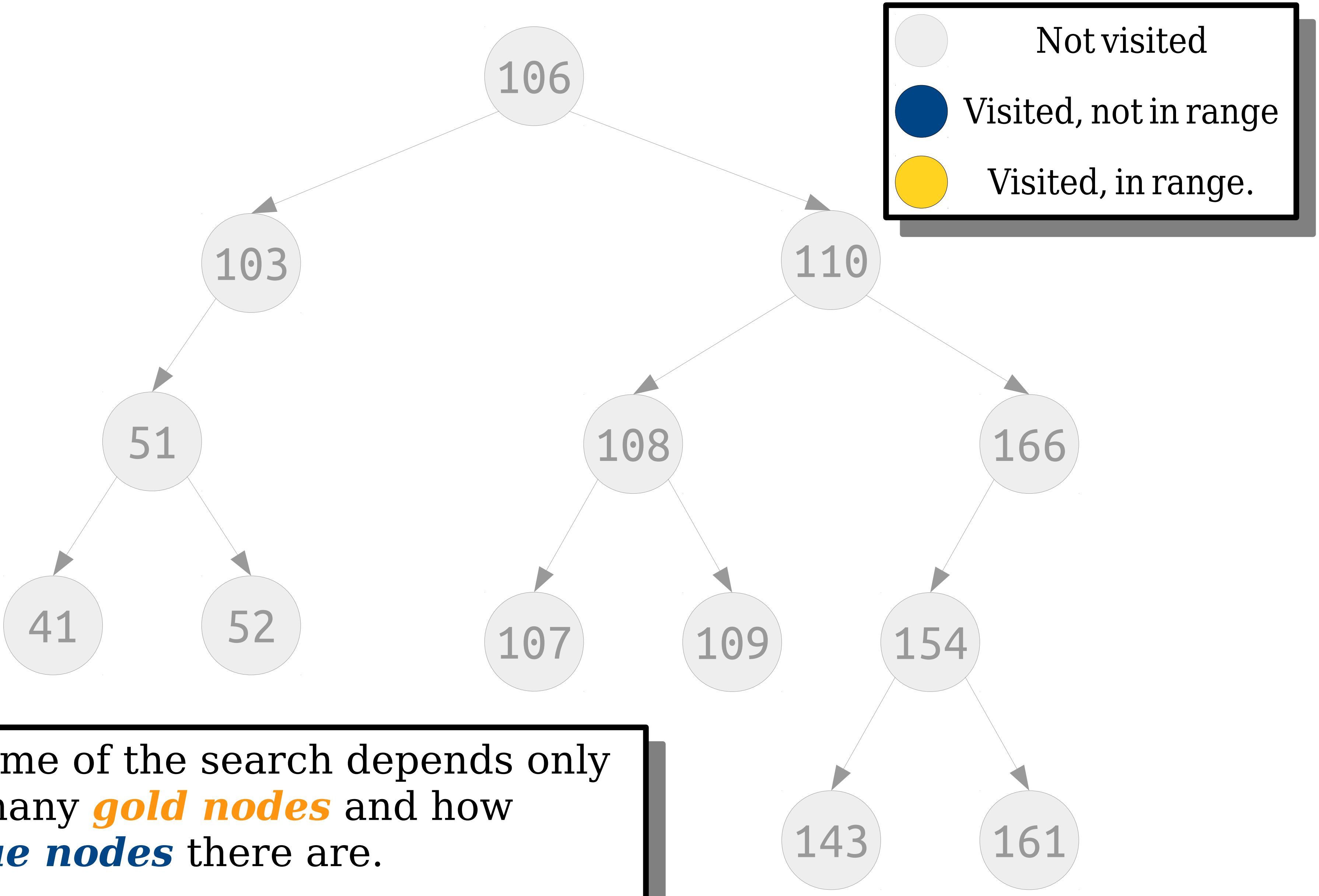
Find all elements in this tree in the range **[42, 165]**.



Find all elements in this tree in the range **[49, 50]**.

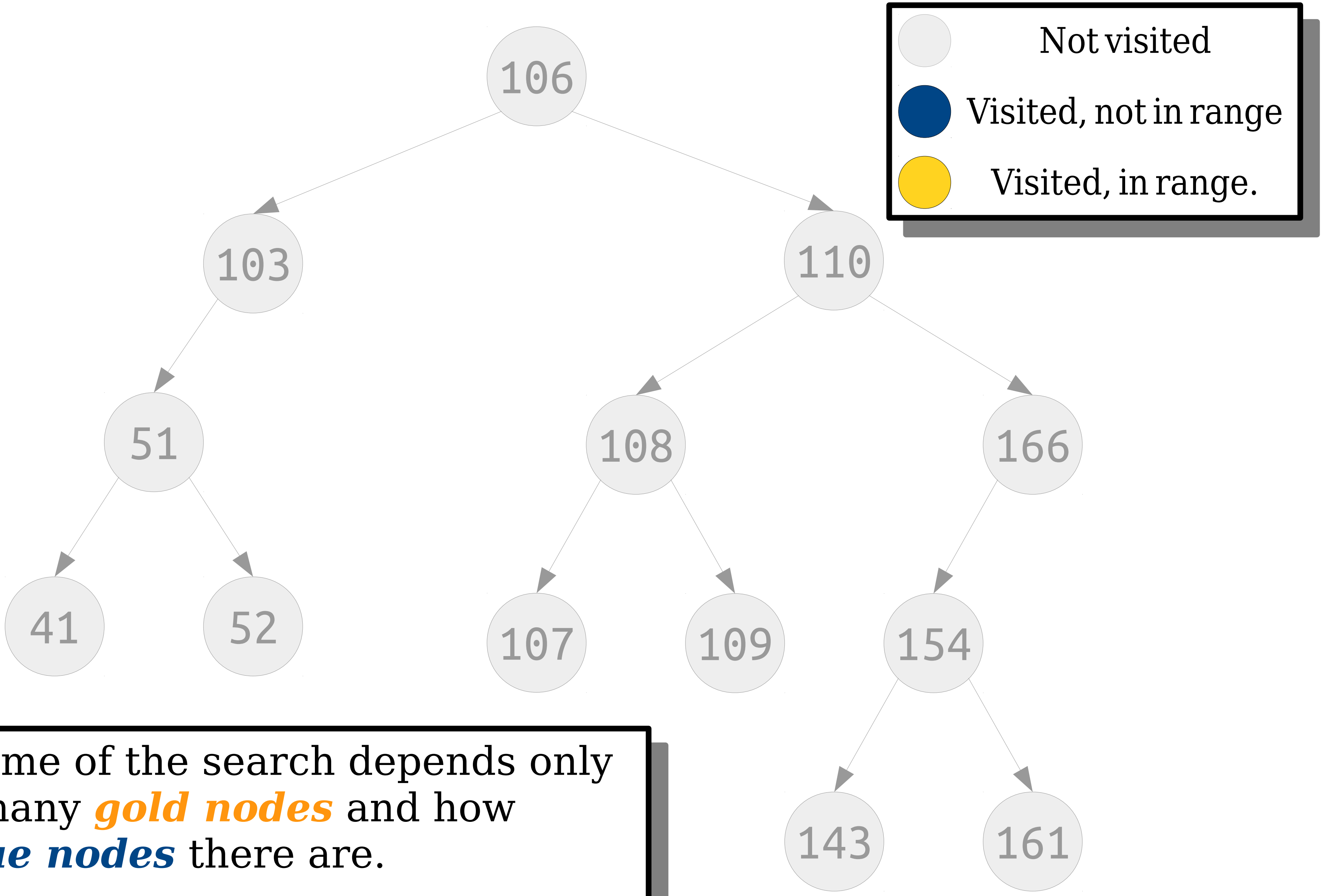


The runtime of the search depends only on how many ***gold nodes*** and how many ***blue nodes*** there are.



The runtime of the search depends only on how many ***gold nodes*** and how many ***blue nodes*** there are.

The number ***gold nodes*** is equal to the number of elements in the range.



○ Not visited

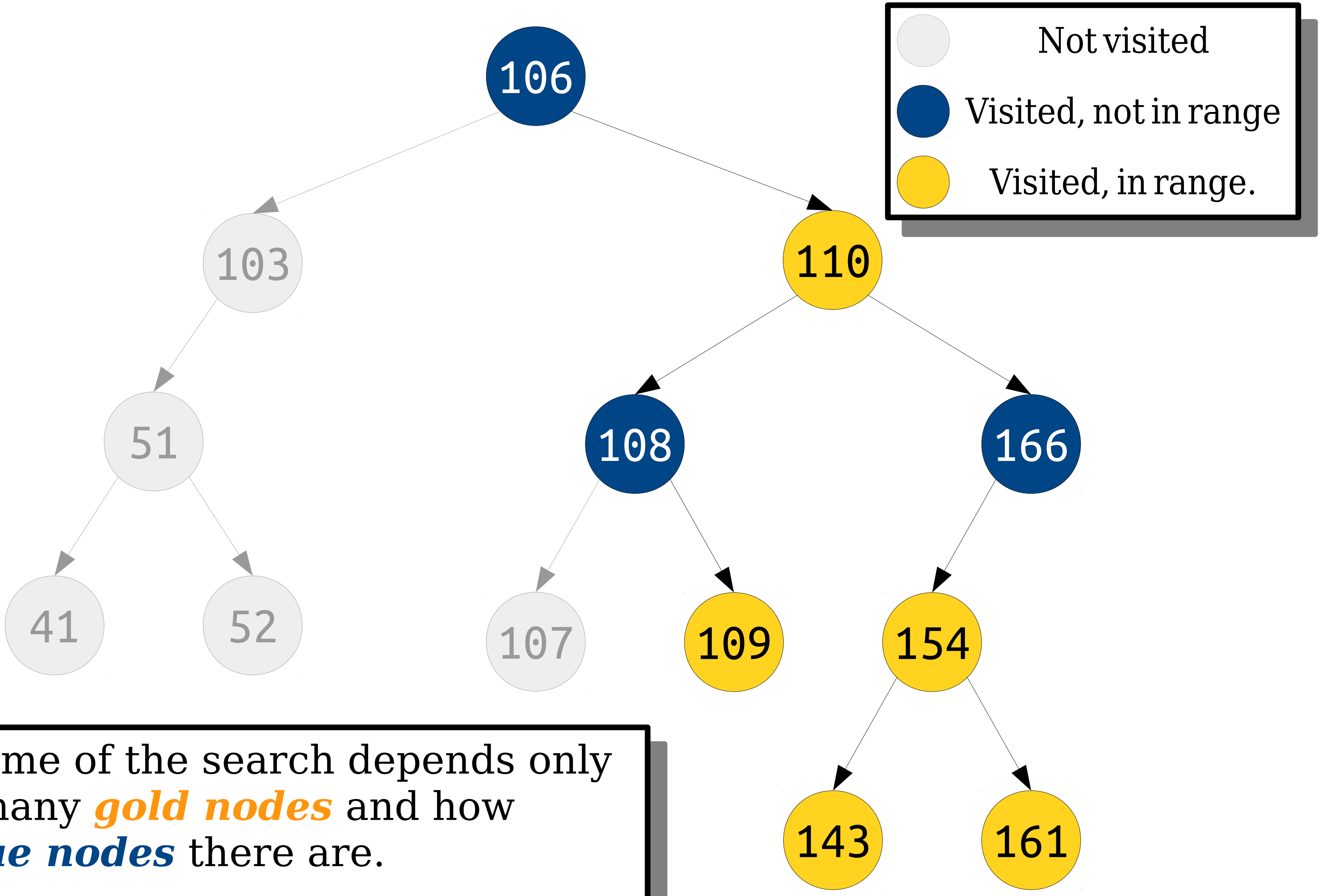
🔵 Visited, not in range

🟡 Visited, in range.

The runtime of the search depends only on how many ***gold nodes*** and how many ***blue nodes*** there are.

The number ***gold nodes*** is equal to the number of elements in the range.

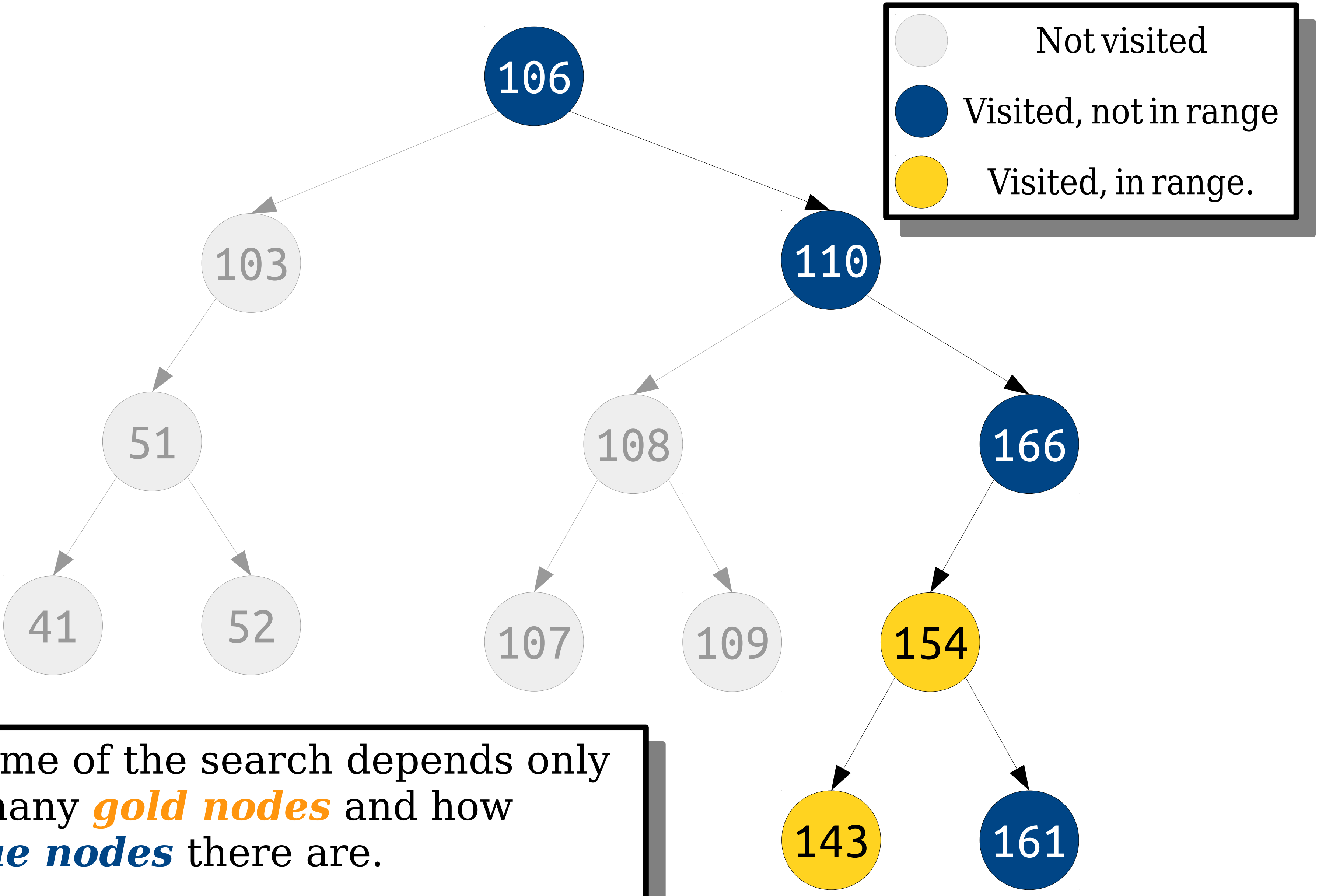
The number of ***blue nodes*** in each layer of the tree is at most two.



The runtime of the search depends only on how many ***gold nodes*** and how many ***blue nodes*** there are.

The number ***gold nodes*** is equal to the number of elements in the range.

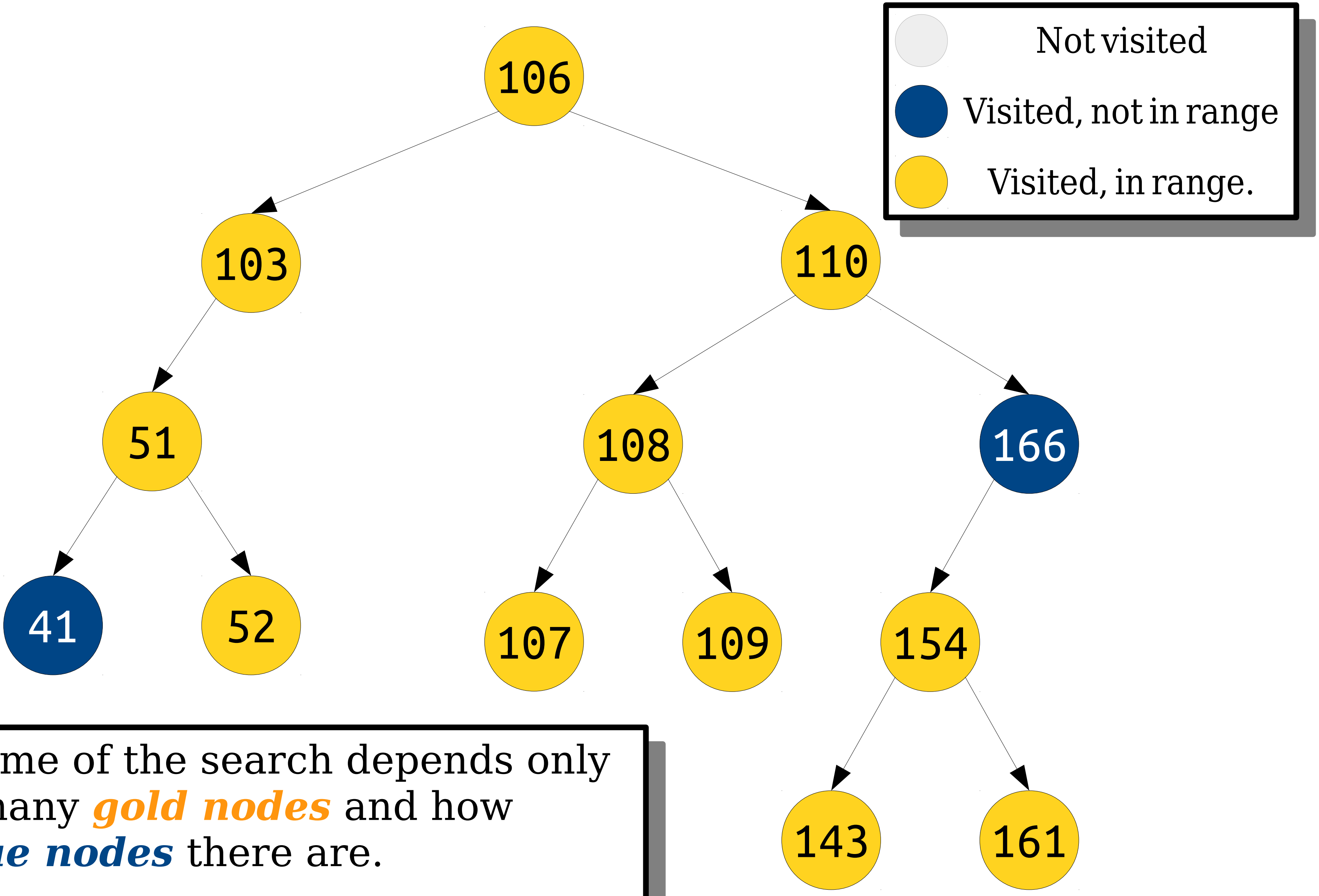
The number of ***blue nodes*** in each layer of the tree is at most two.



The runtime of the search depends only on how many ***gold nodes*** and how many ***blue nodes*** there are.

The number ***gold nodes*** is equal to the number of elements in the range.

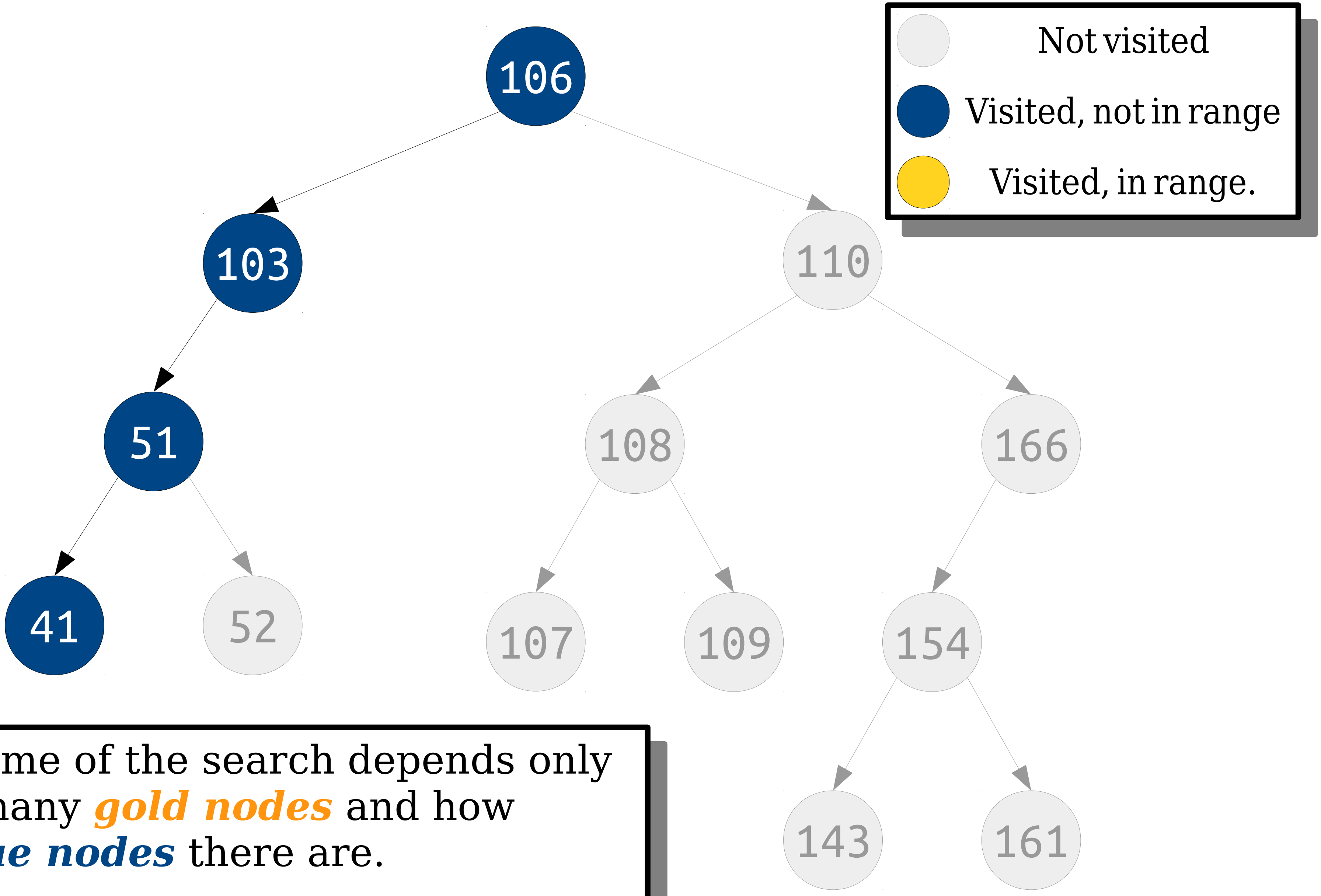
The number of ***blue nodes*** in each layer of the tree is at most two.



The runtime of the search depends only on how many ***gold nodes*** and how many ***blue nodes*** there are.

The number ***gold nodes*** is equal to the number of elements in the range.

The number of ***blue nodes*** in each layer of the tree is at most two.



The runtime of the search depends only on how many ***gold nodes*** and how many ***blue nodes*** there are.

The number ***gold nodes*** is equal to the number of elements in the range.

The number of ***blue nodes*** in each layer of the tree is at most two.

Range Searches

- The cost of a range search in a BST of height h is

  $$\mathbf{O(\mathit{h} + \mathit{k})},$$

  where k is the number of matches reported.
- Notice that
  - if k is low (close to 0), then the runtime is close to O(h), the cost of a single search; and
  - if k is high (close to n), then the runtime is close to O(n), the cost of an inorder traversal.
- This is called an ***output-sensitive algorithm***.