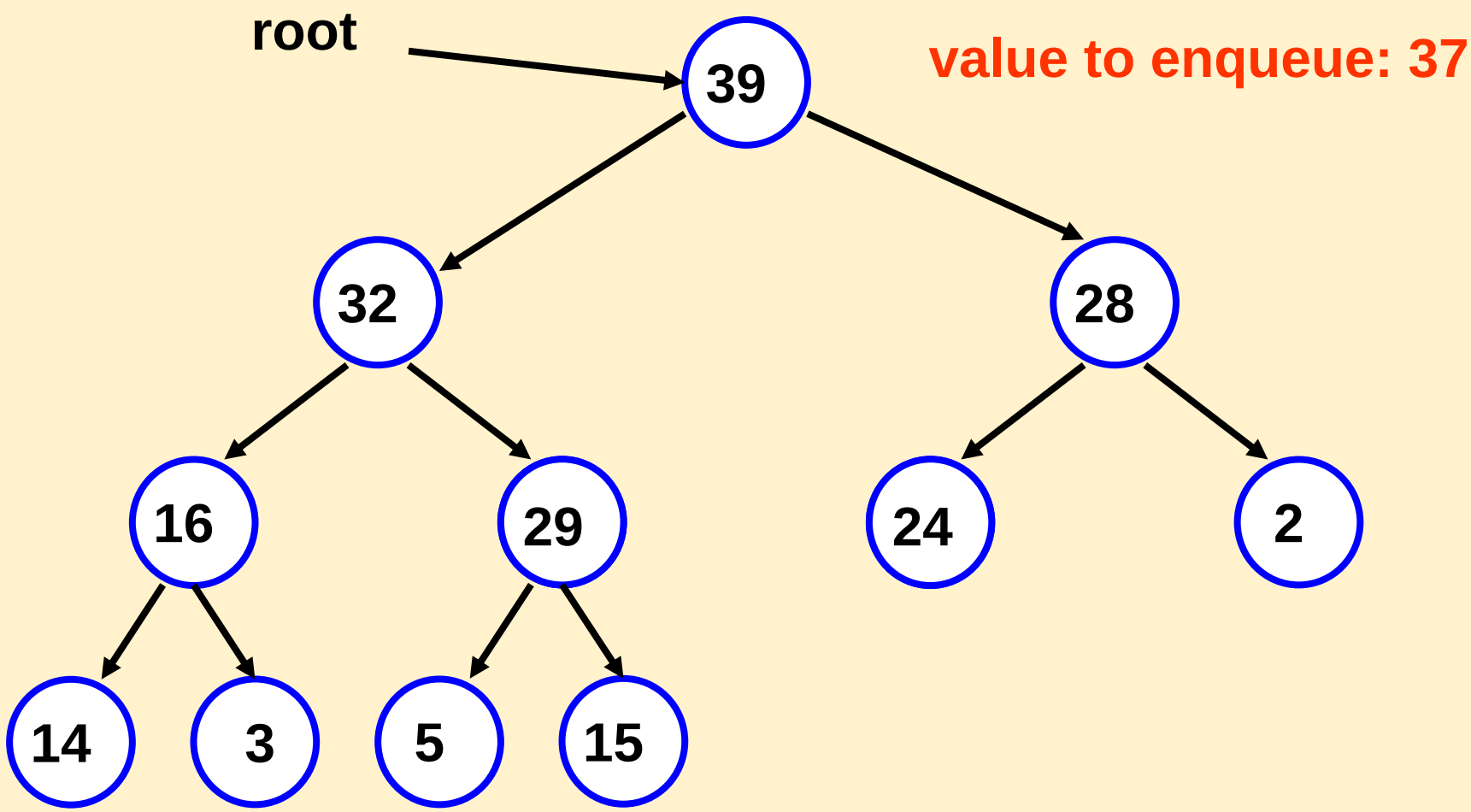
Enqueue (ReheapUP)

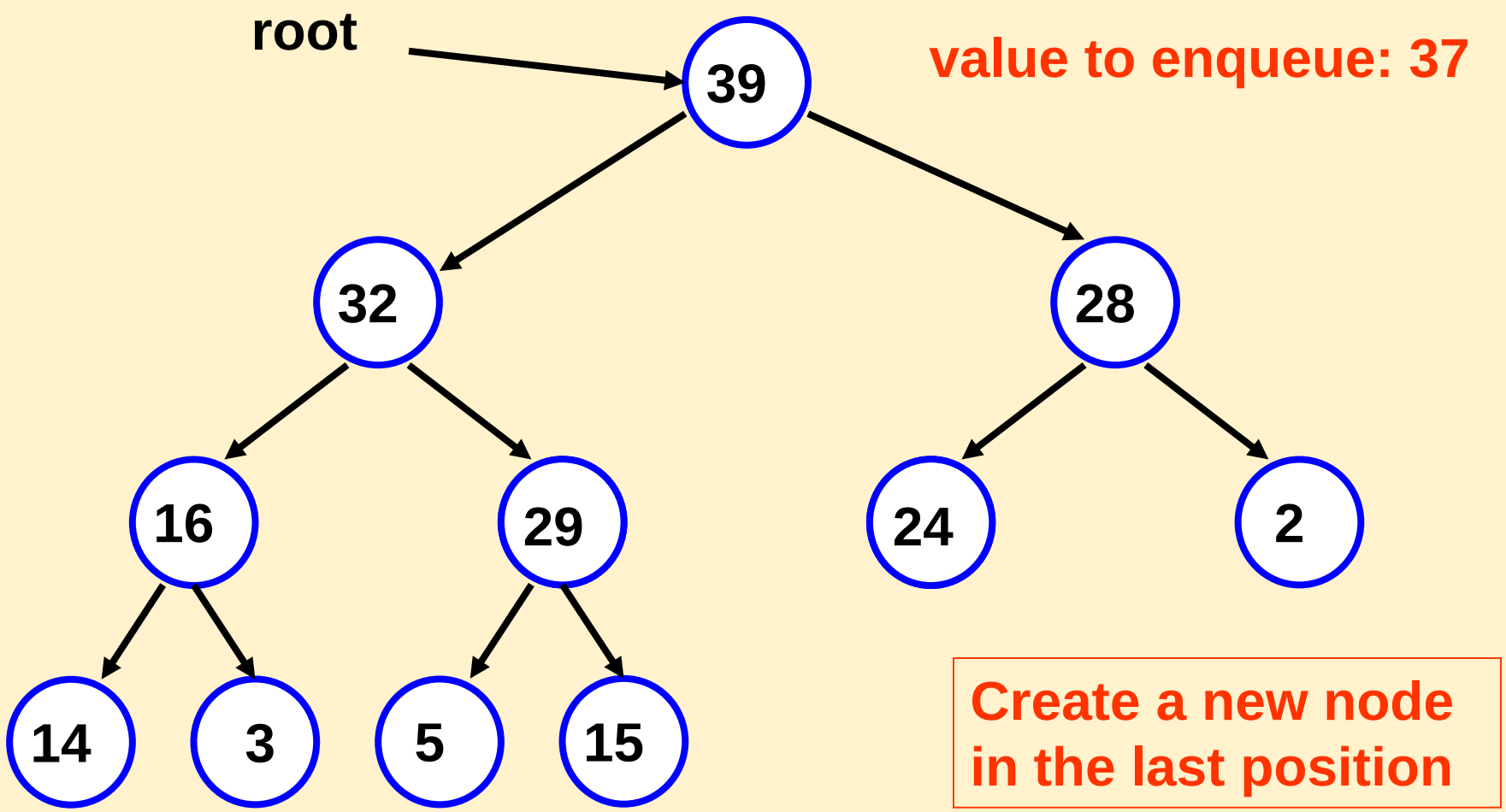
Enqueue or ReheapUp process

- Suppose we want to add or insert an element to the heap. Where do we put it? The shape property tells us that we first place the new element in the last position of the heap, which probably won't make a heap unless we are very lucky.
- We compare the new element in the last node with the element of its parent. If its parent has a smaller value, then we swap elements. Now the parent has a larger value, but it still might not work for its parent.
- We keep repeating this process until we find a parent with a value greater than or equal to the element we just swapped with.

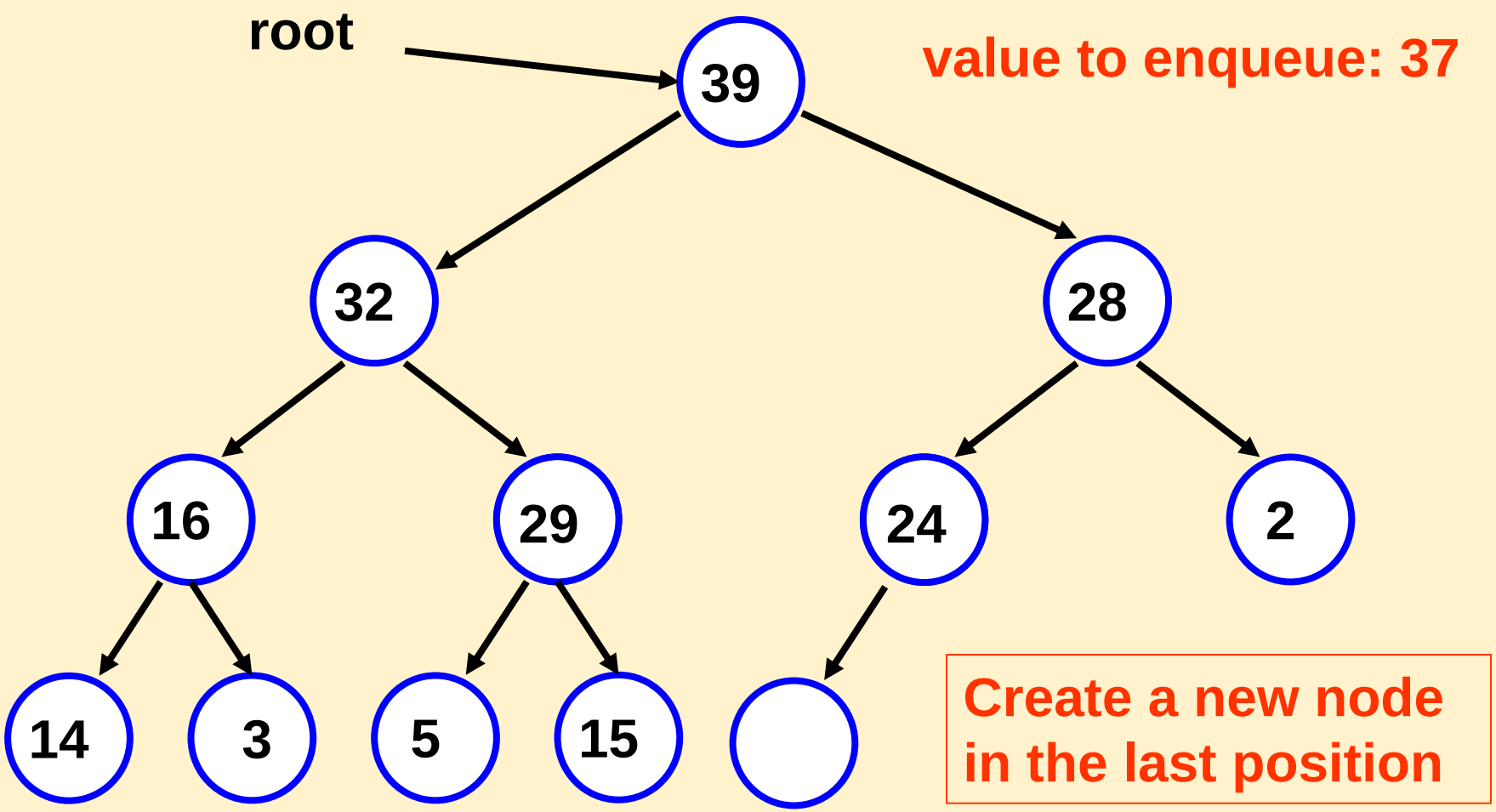
Enqueue



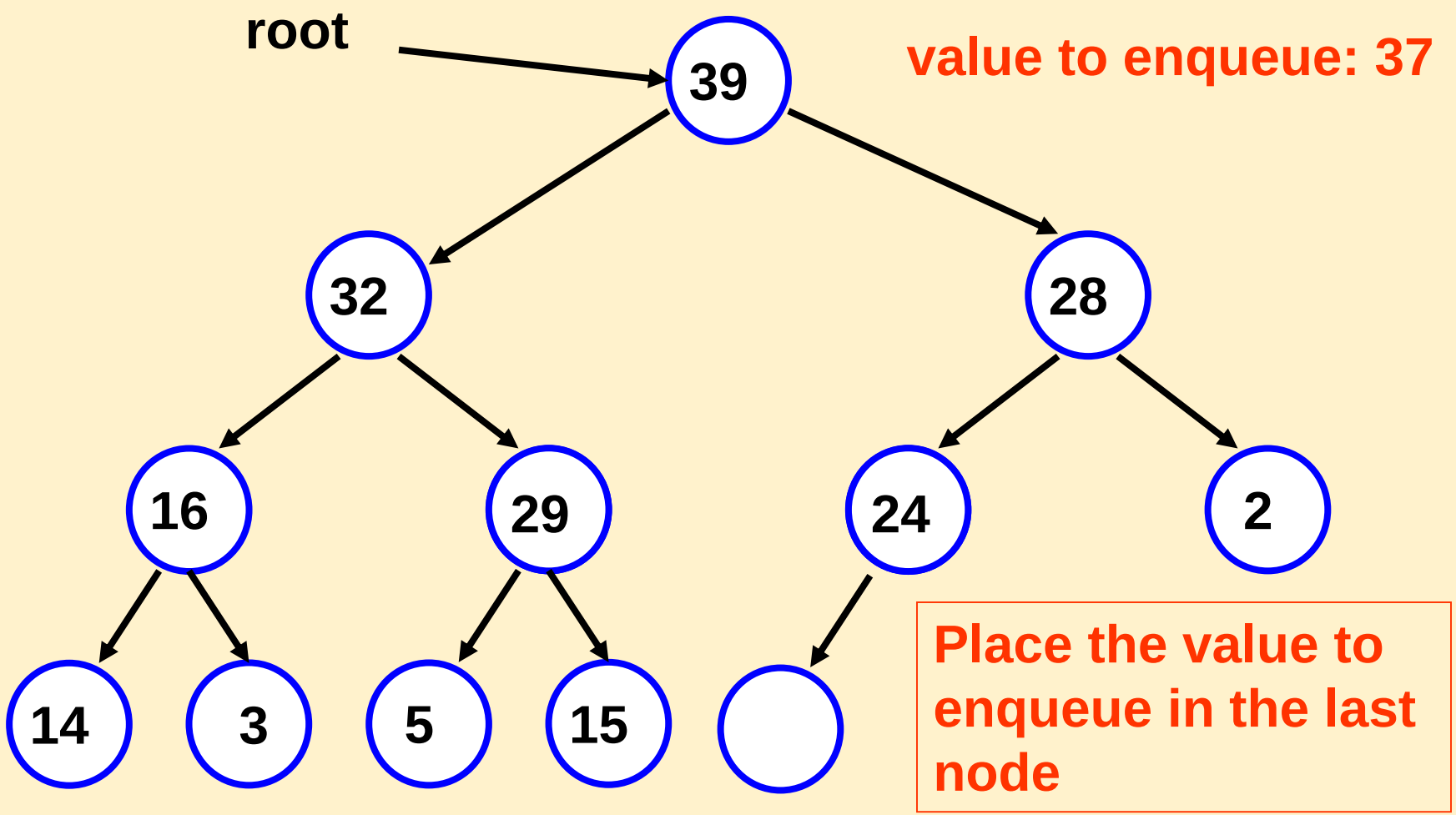
Enqueue (cont.)



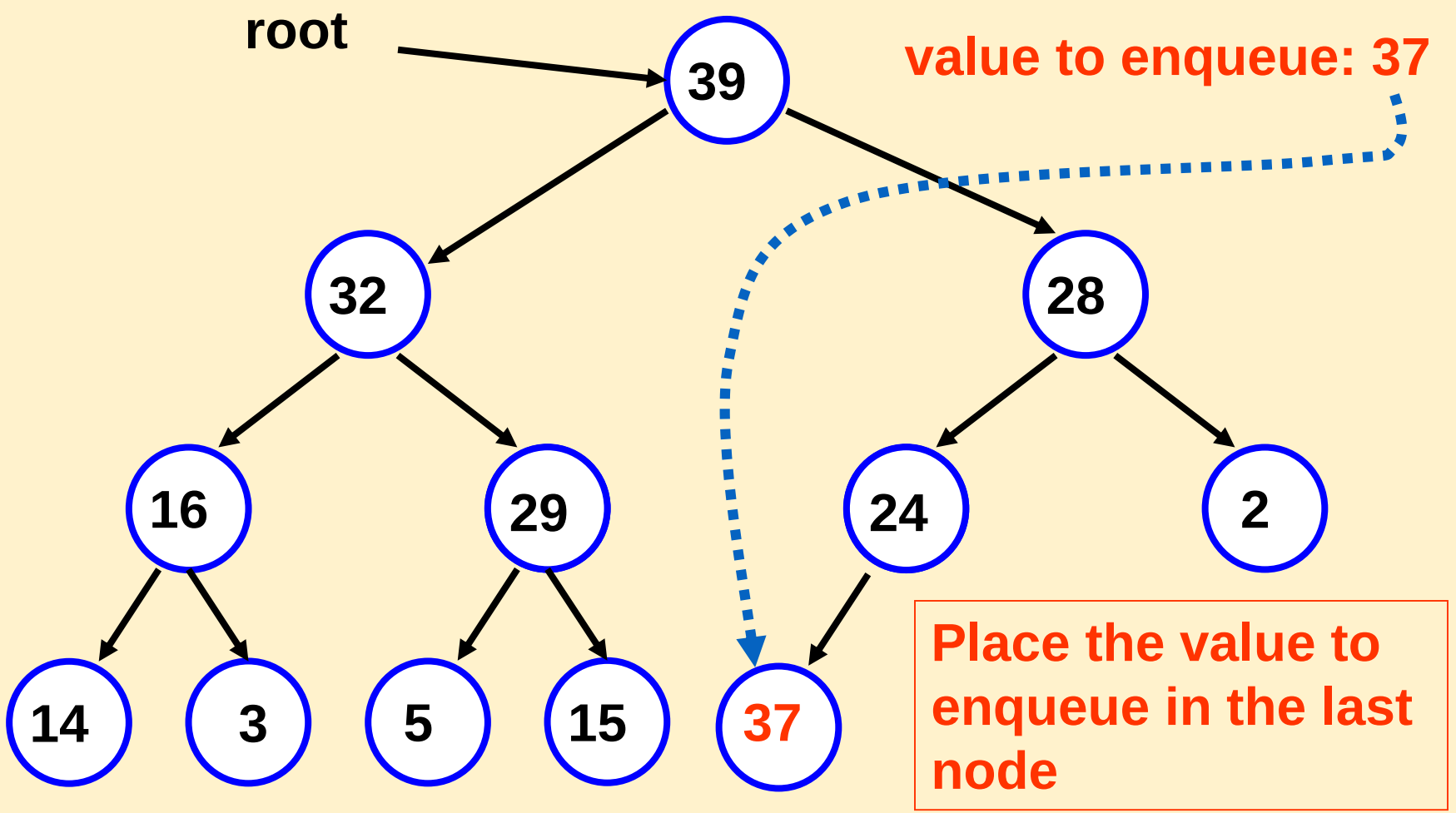
Enqueue (cont.)



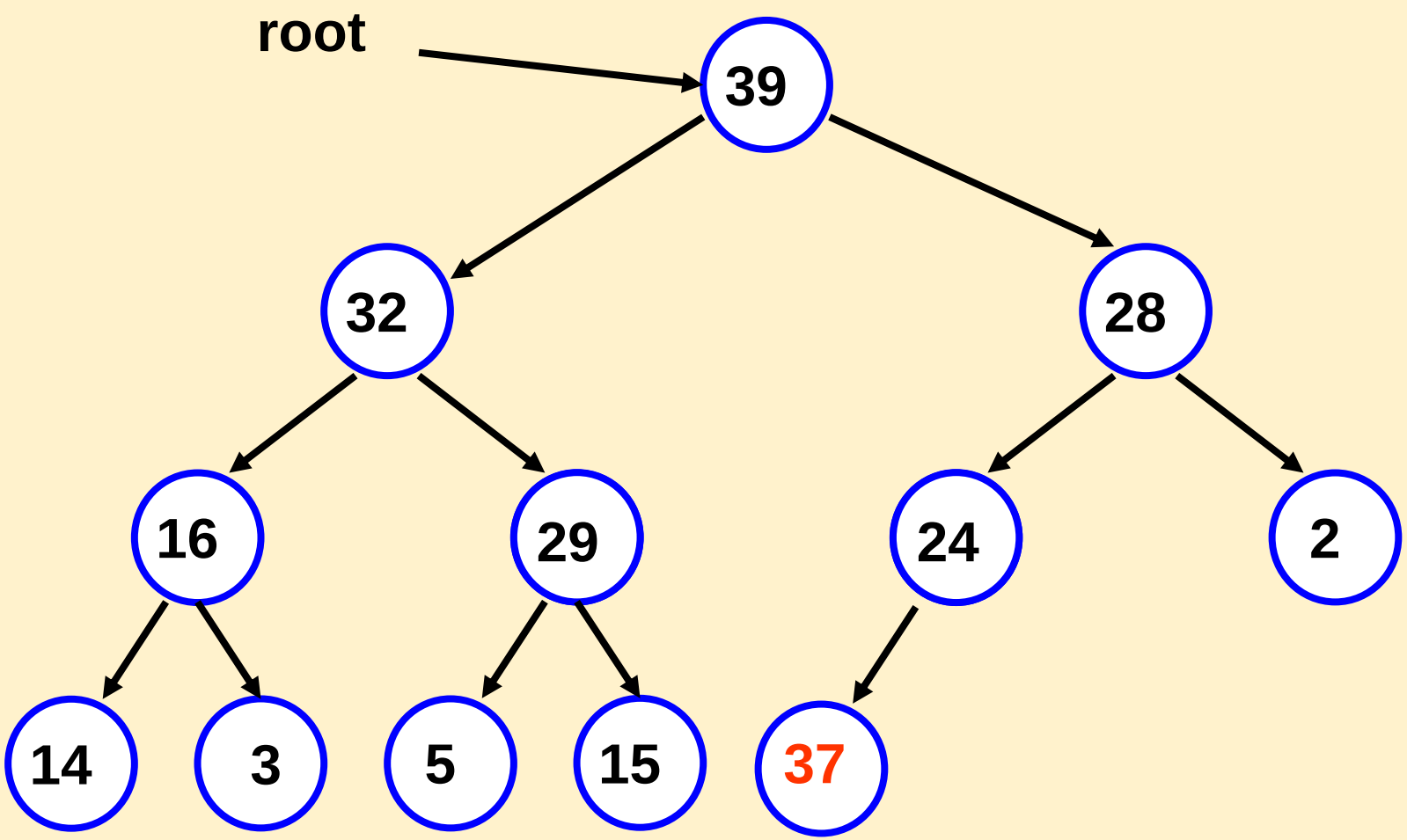
Enqueue (cont.)



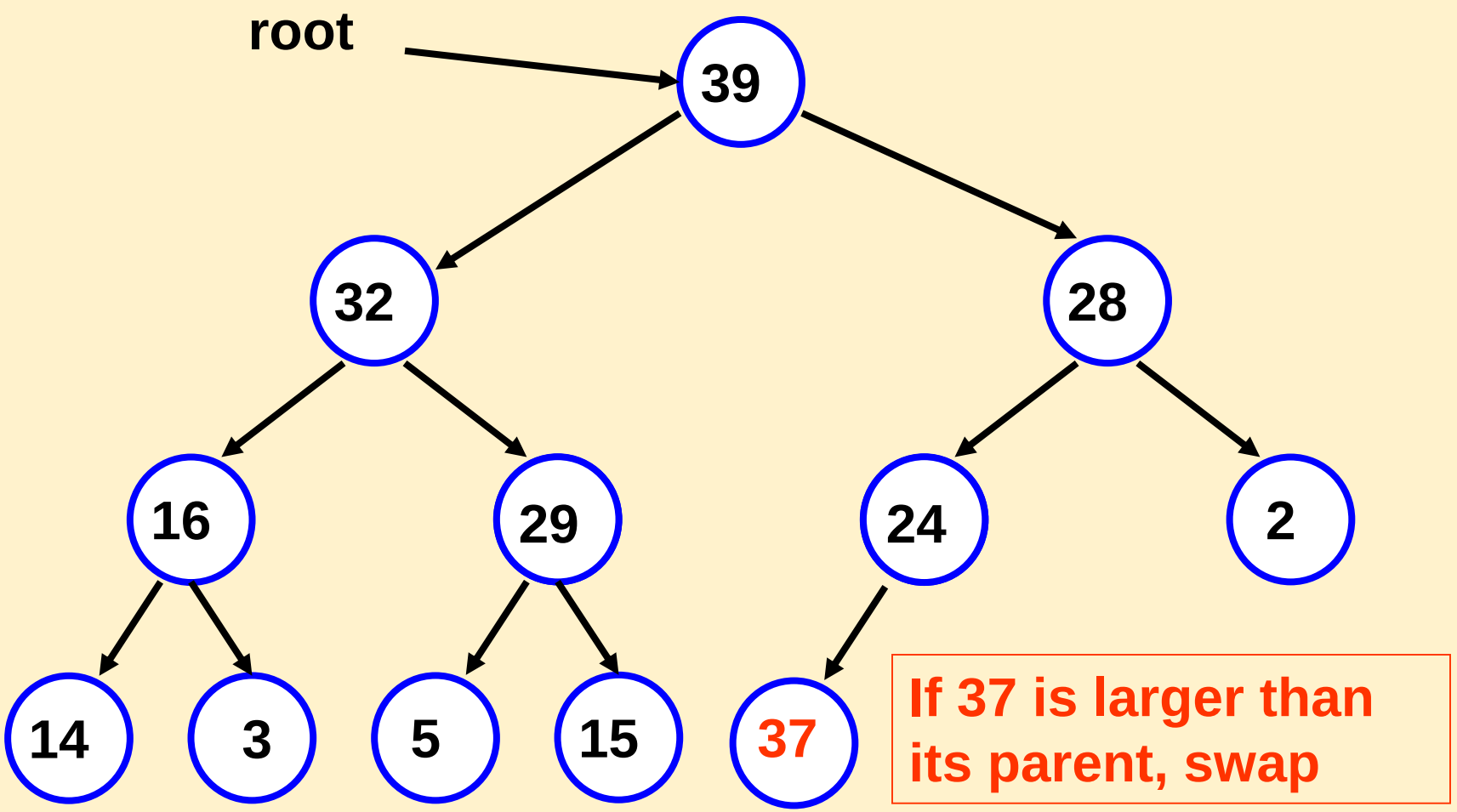
Enqueue (cont.)



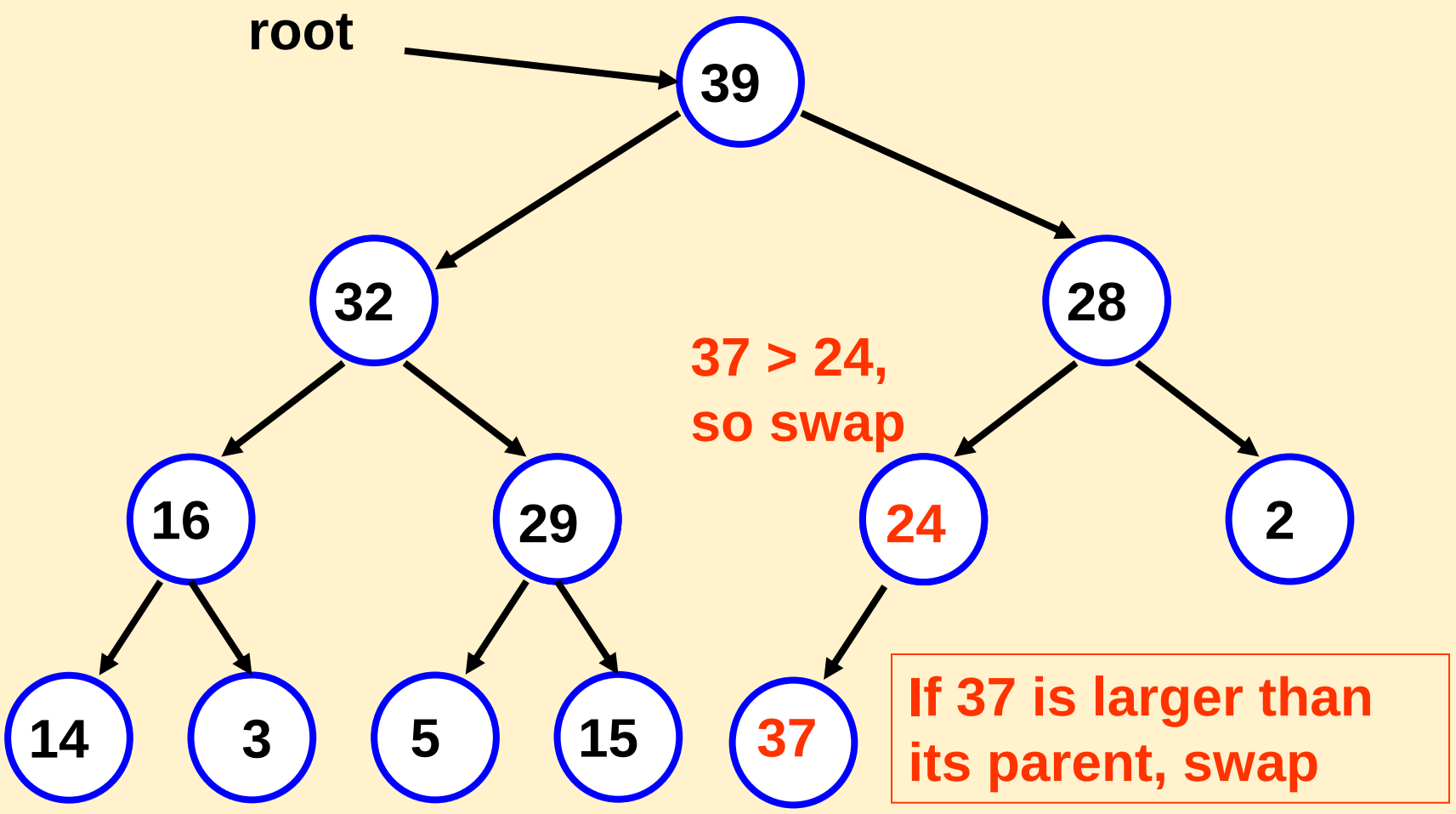
Enqueue (cont.)



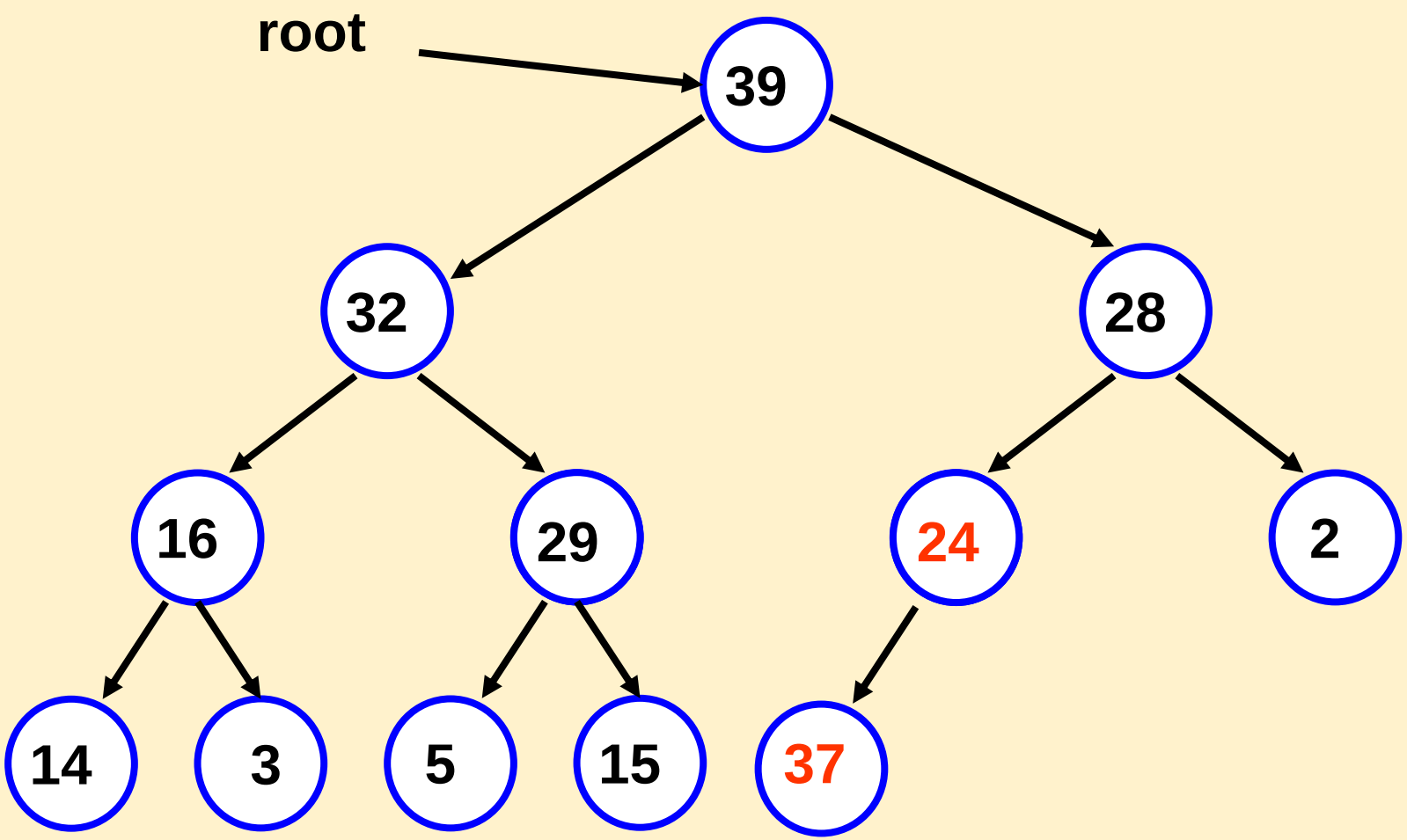
Enqueue (cont.)



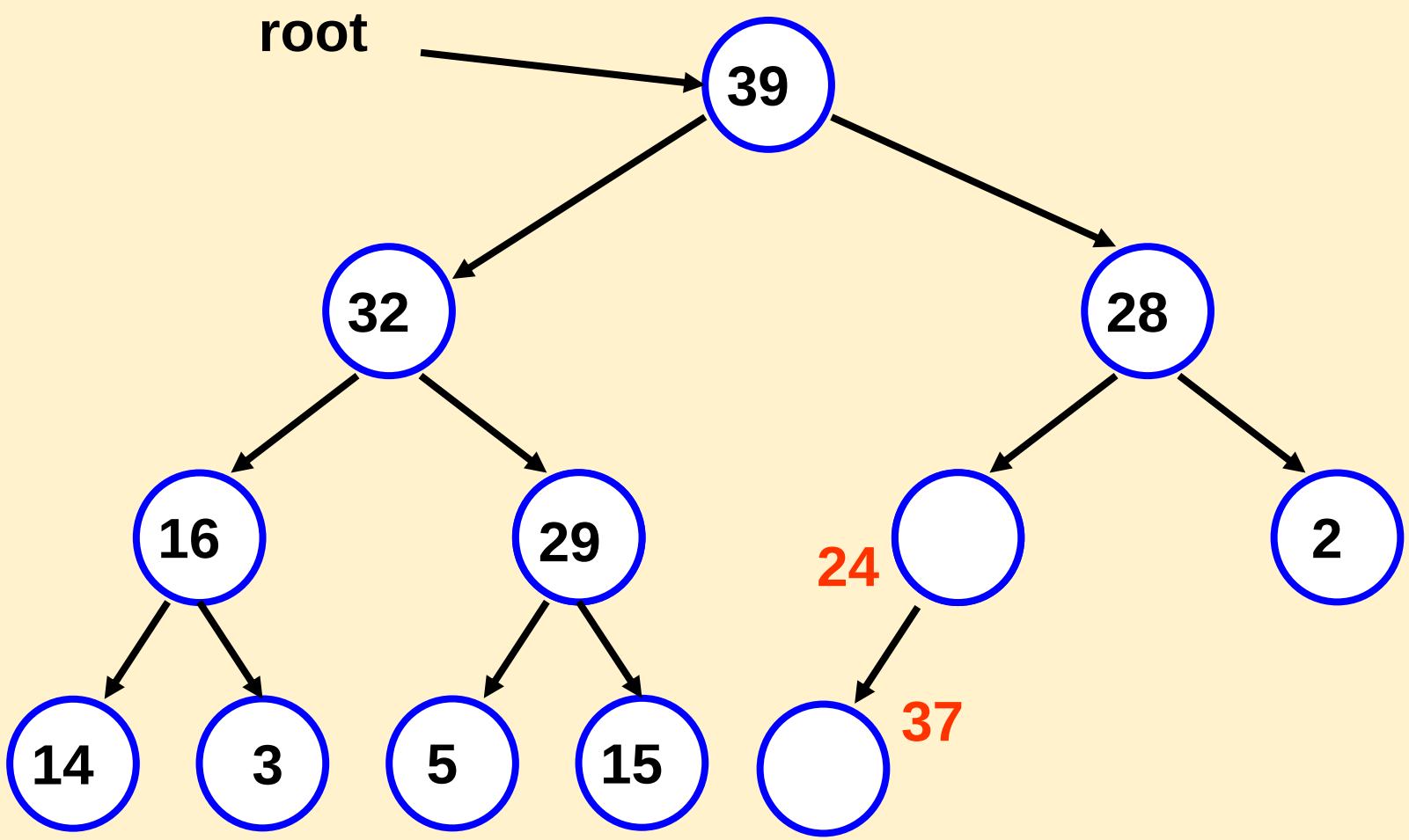
Enqueue (cont.)



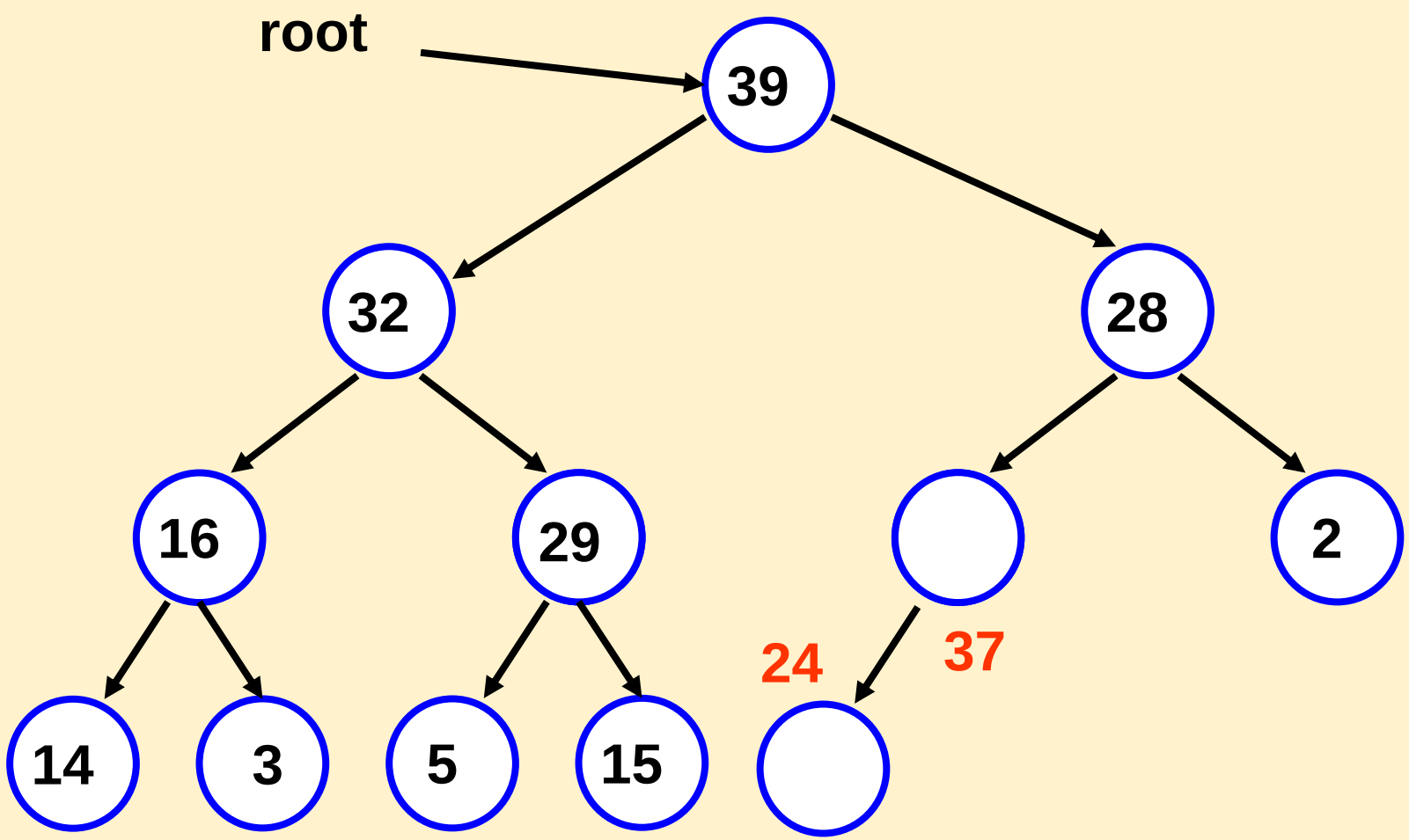
Enqueue (cont.)



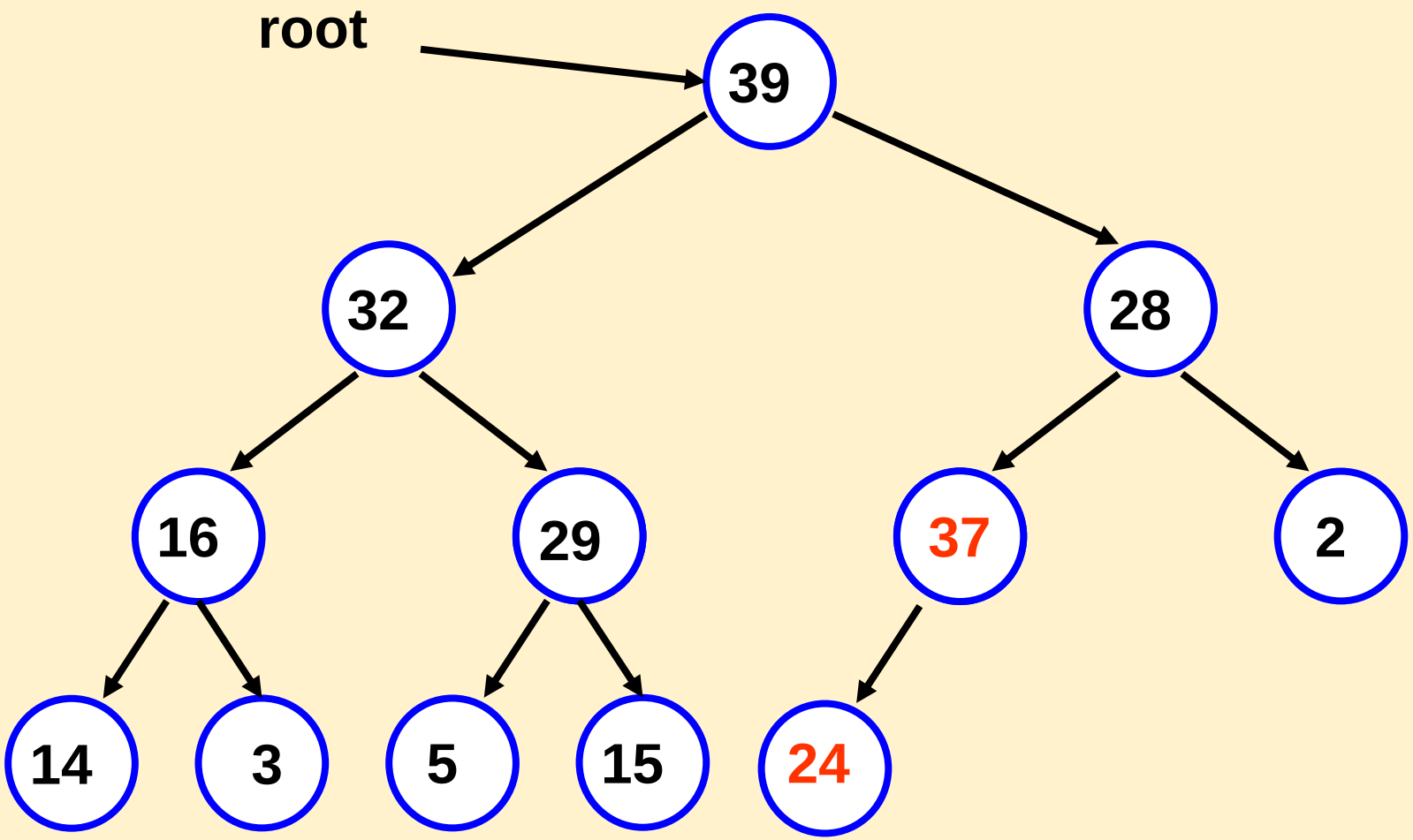
Enqueue (cont.)



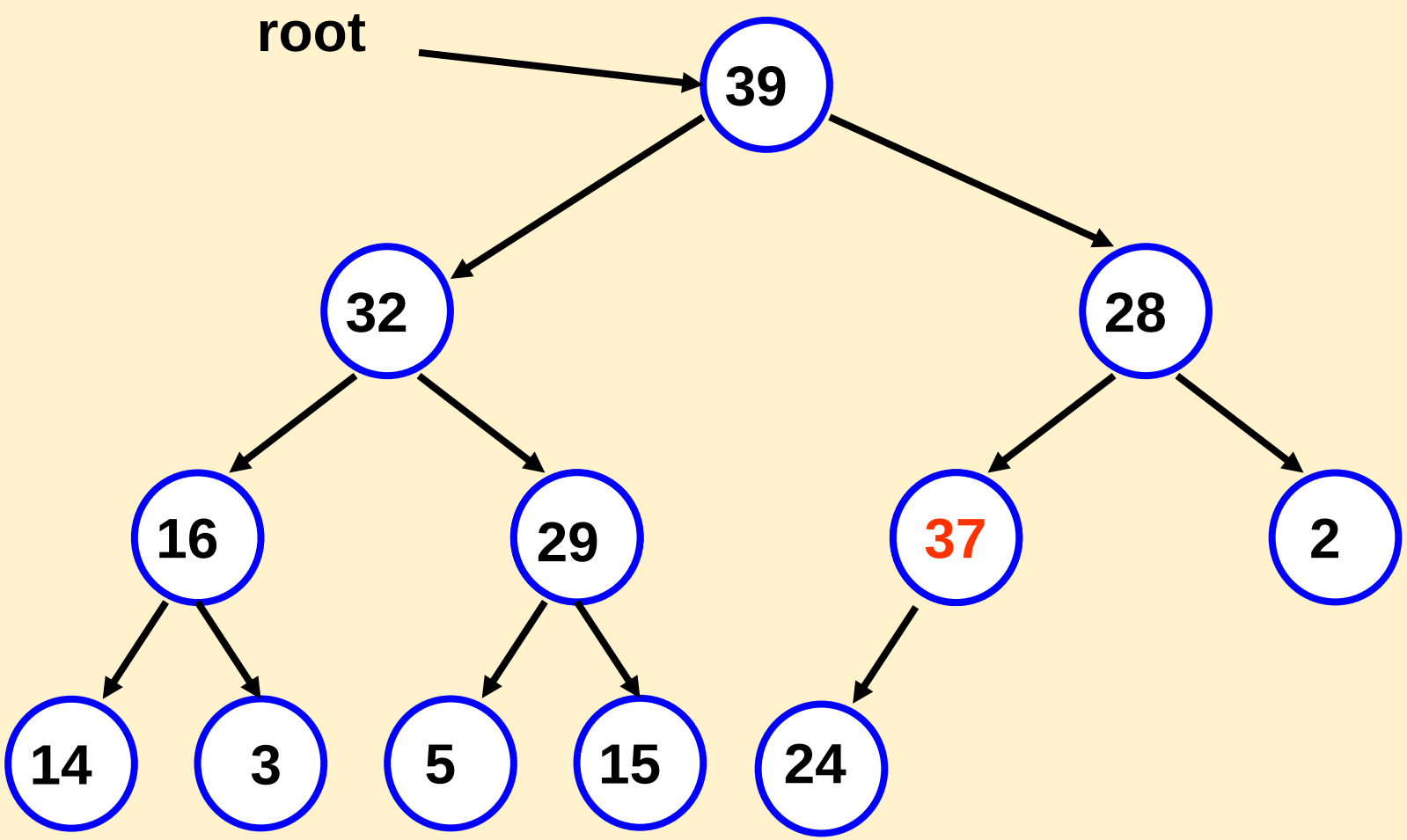
Enqueue (cont.)



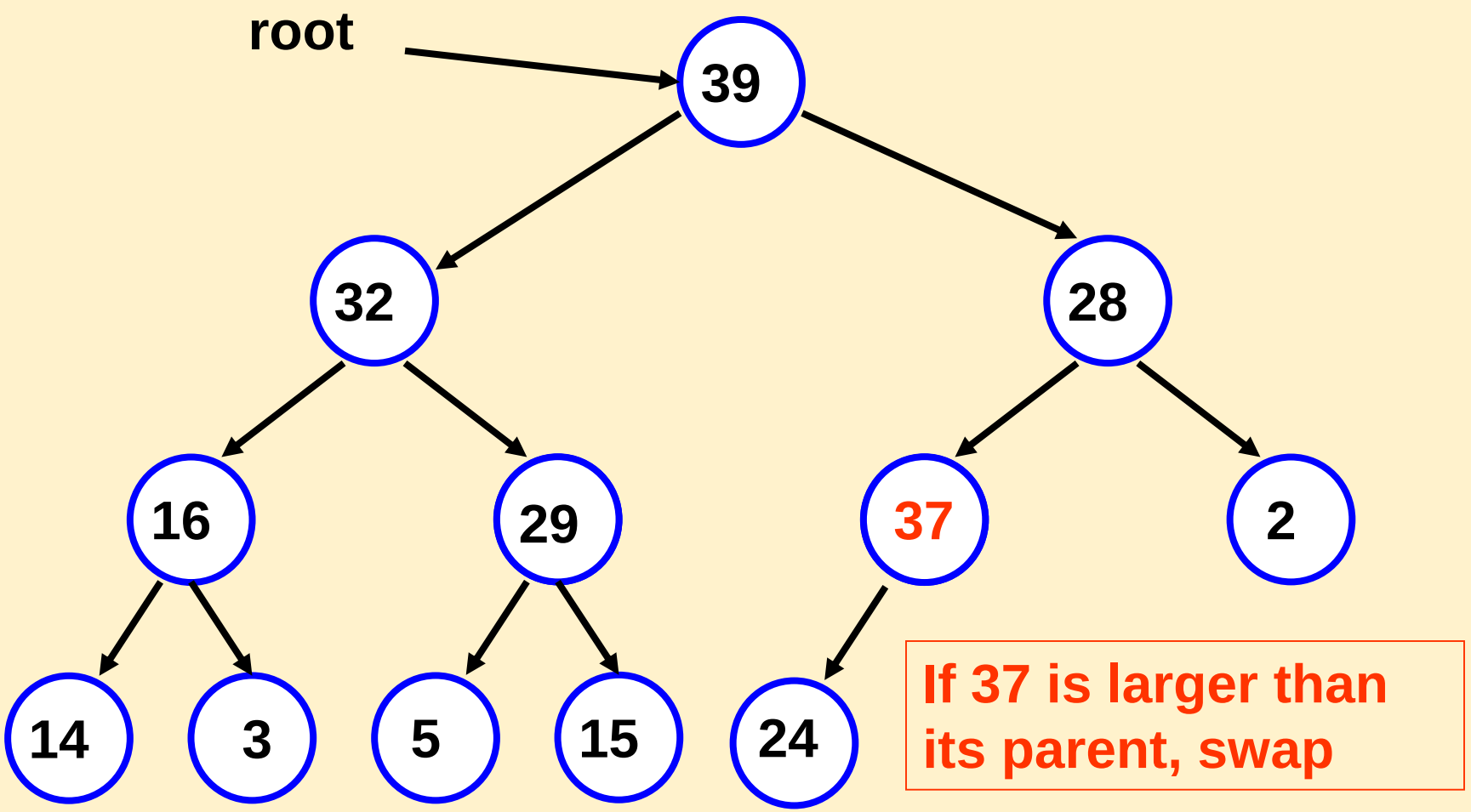
Enqueue (cont.)



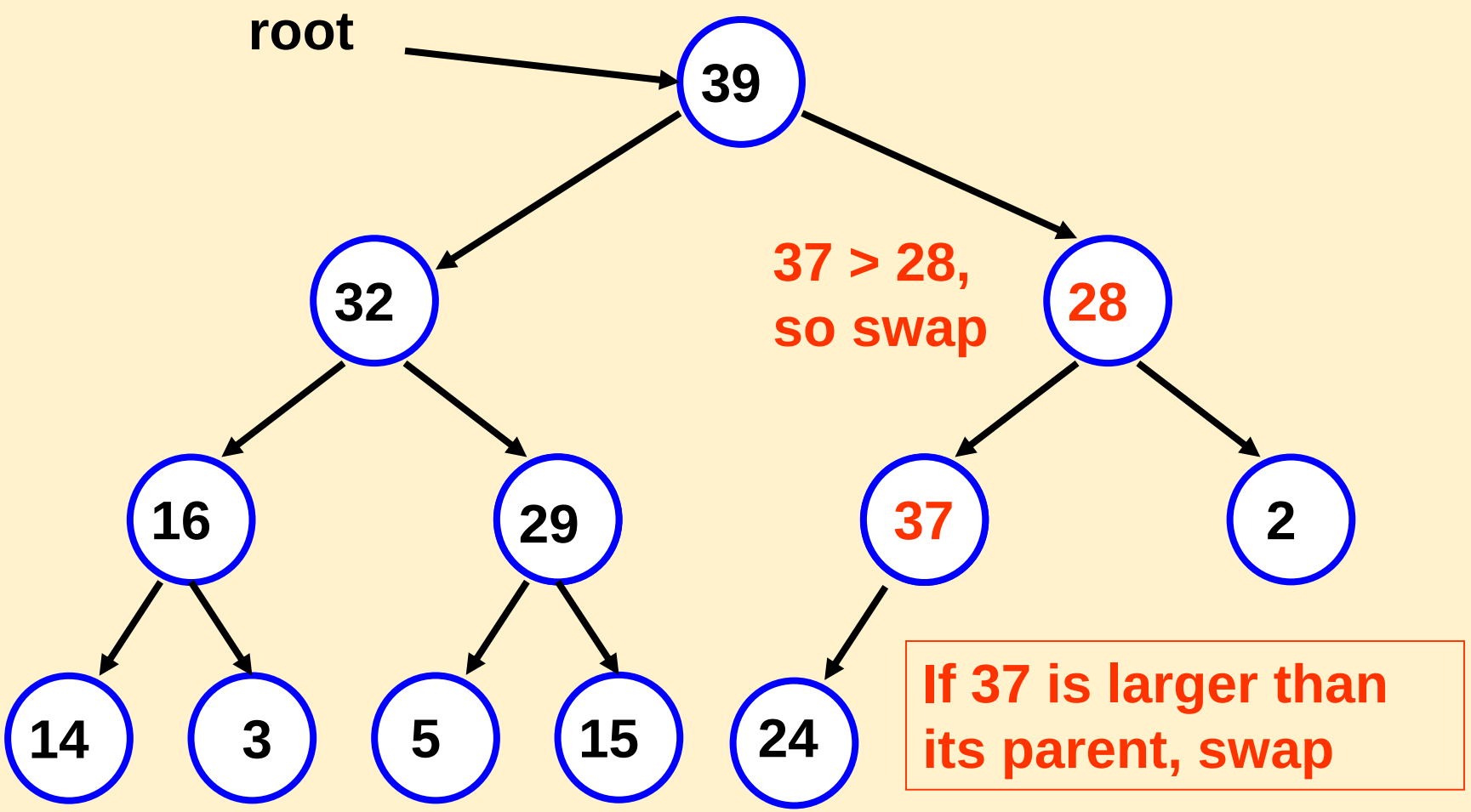
Enqueue (cont.)



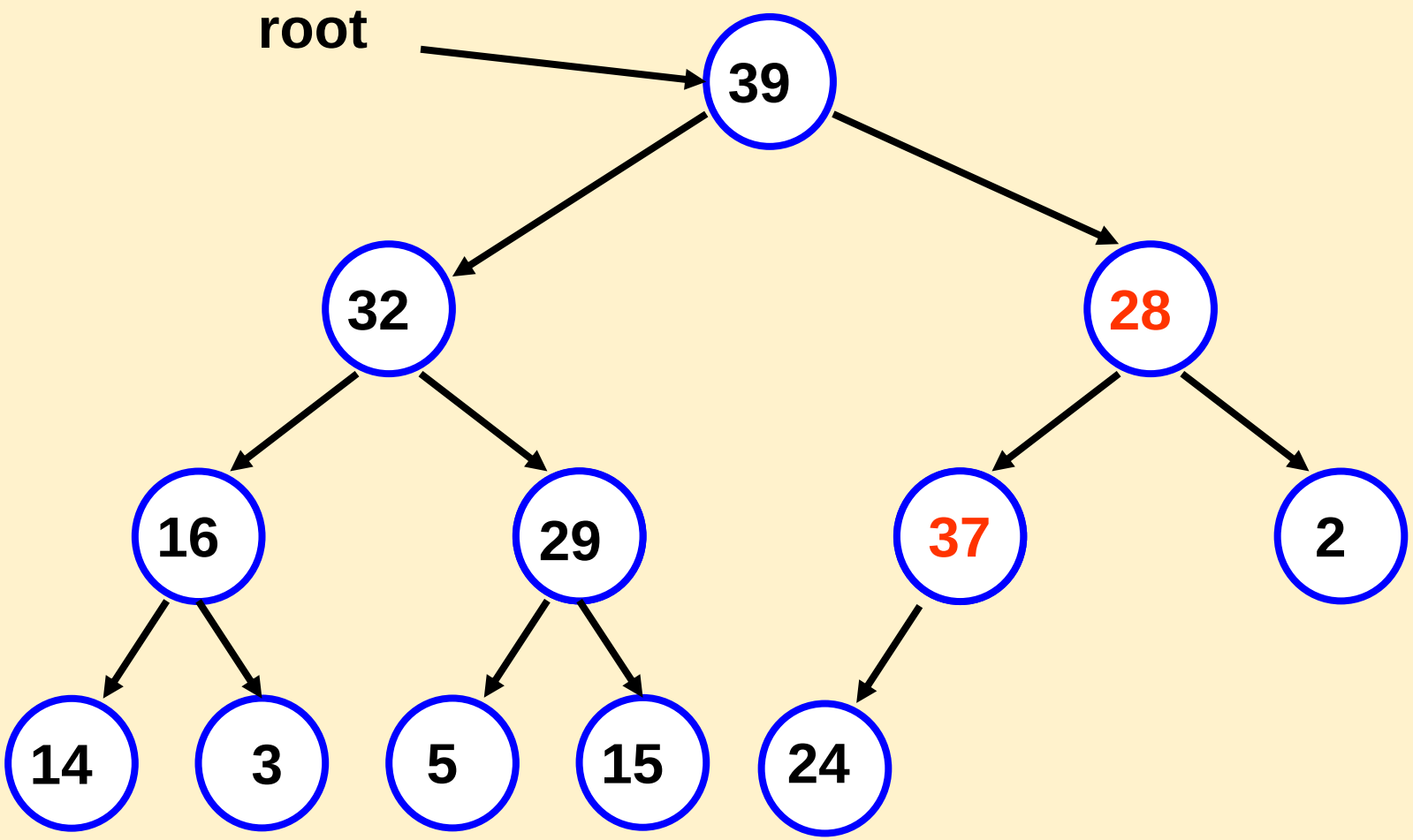
Enqueue (cont.)



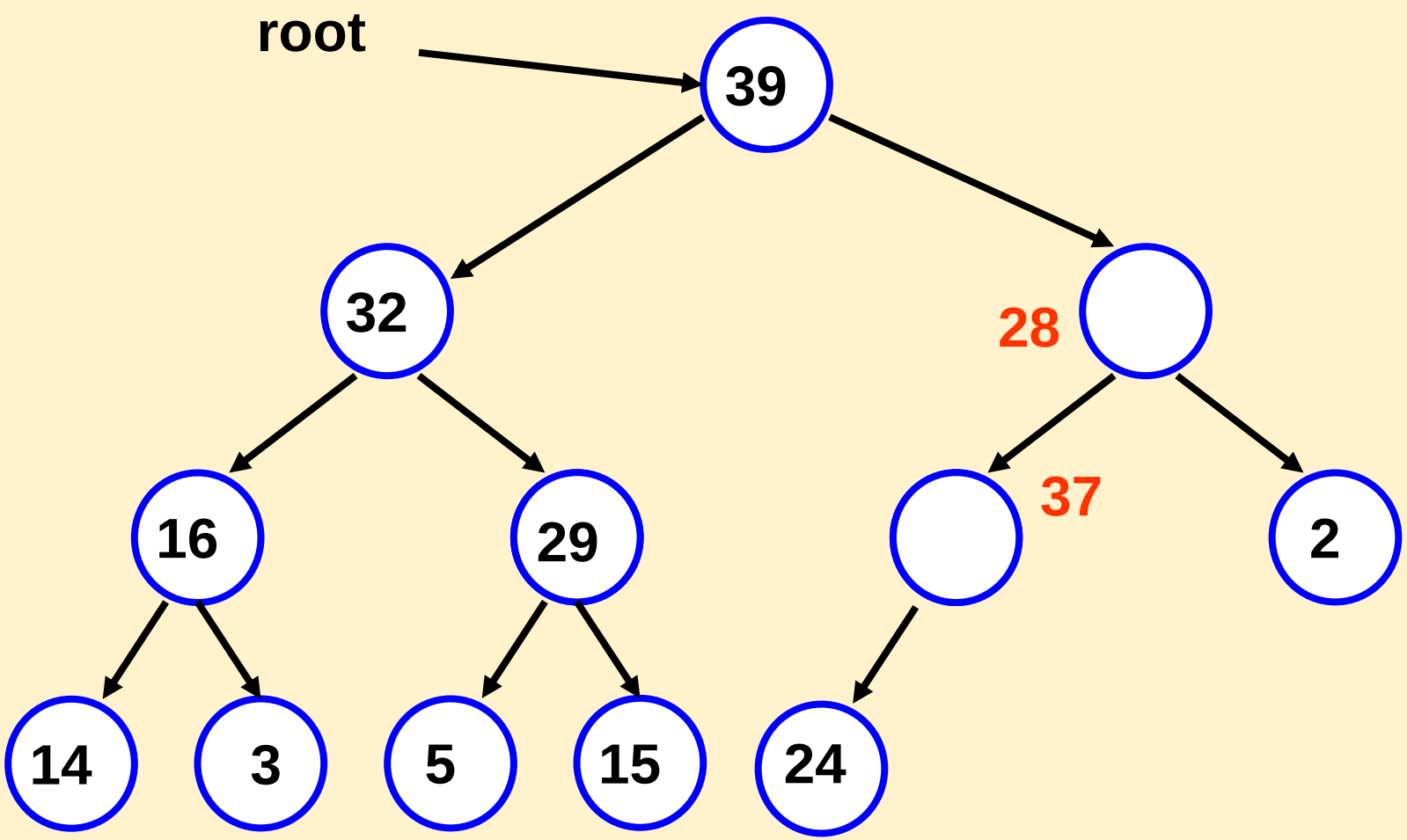
Enqueue (cont.)



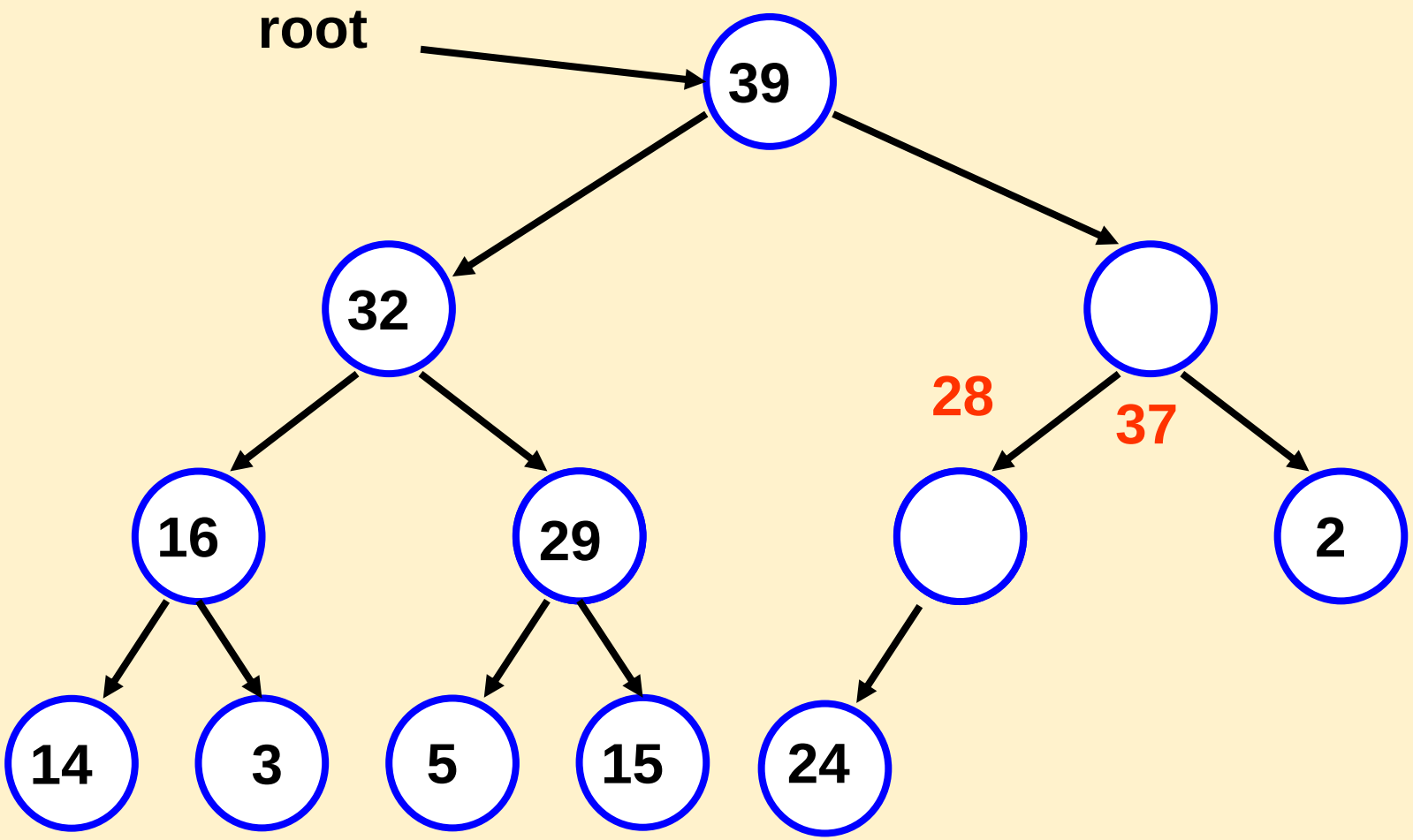
Enqueue (cont.)



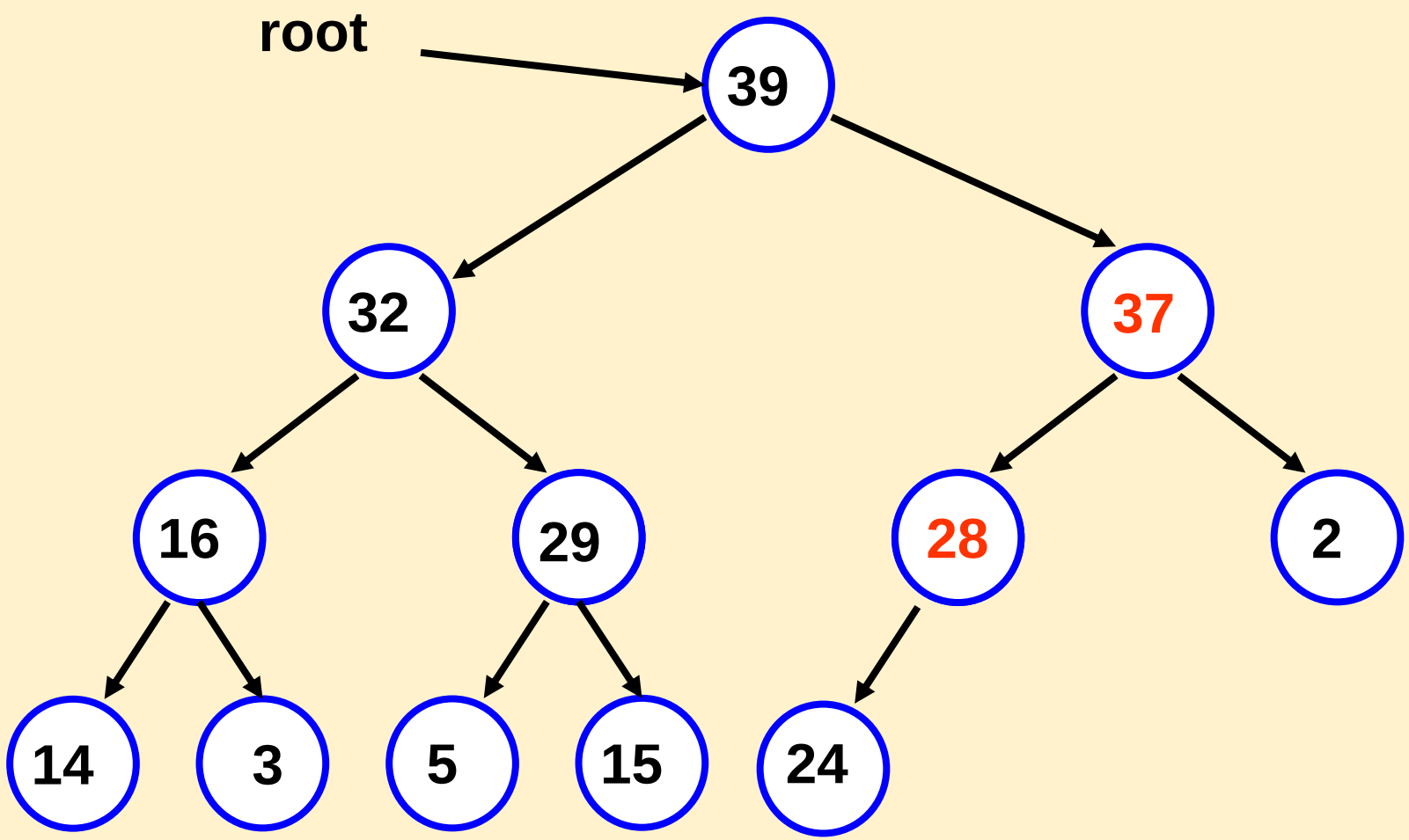
Enqueue (cont.)



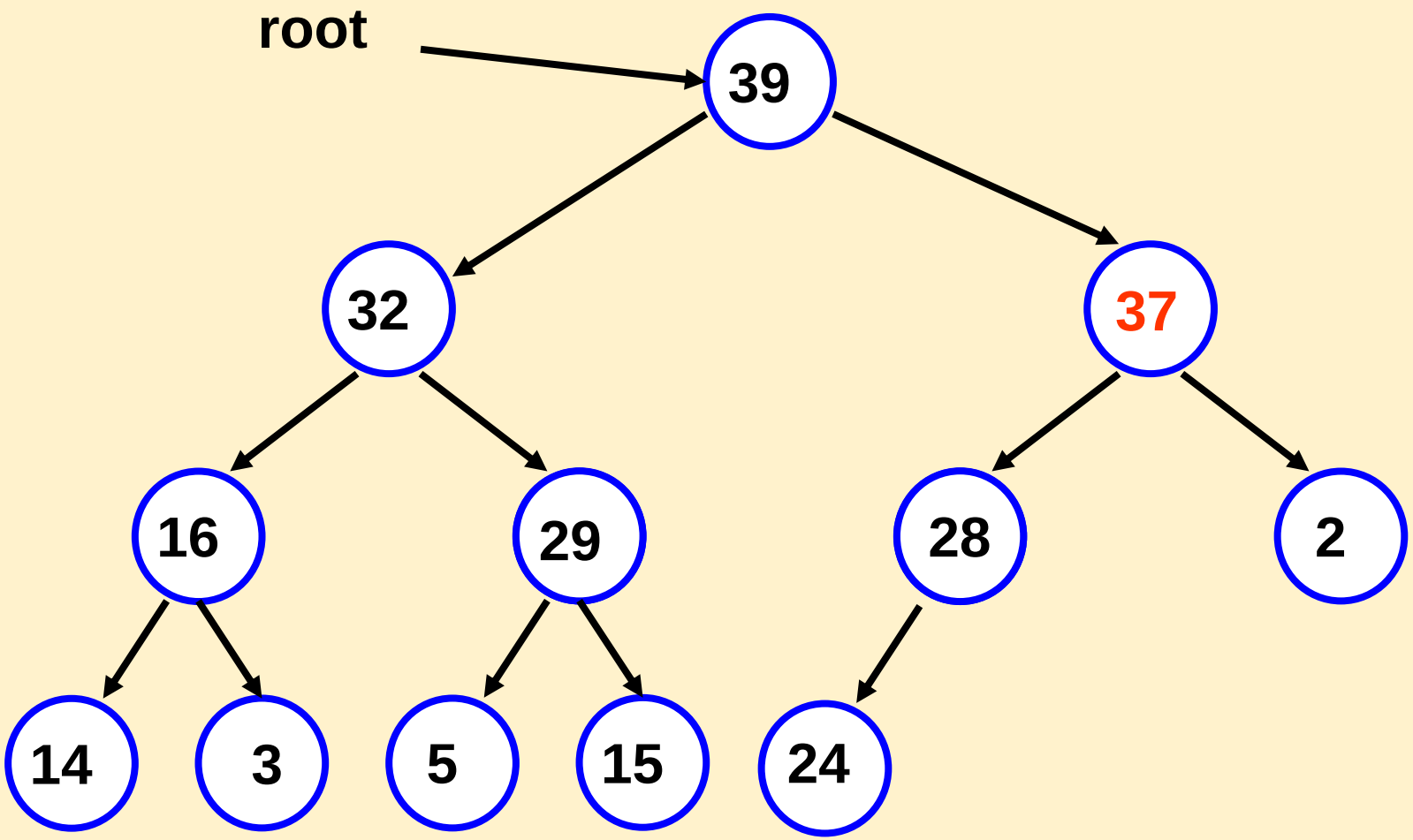
Enqueue (cont.)



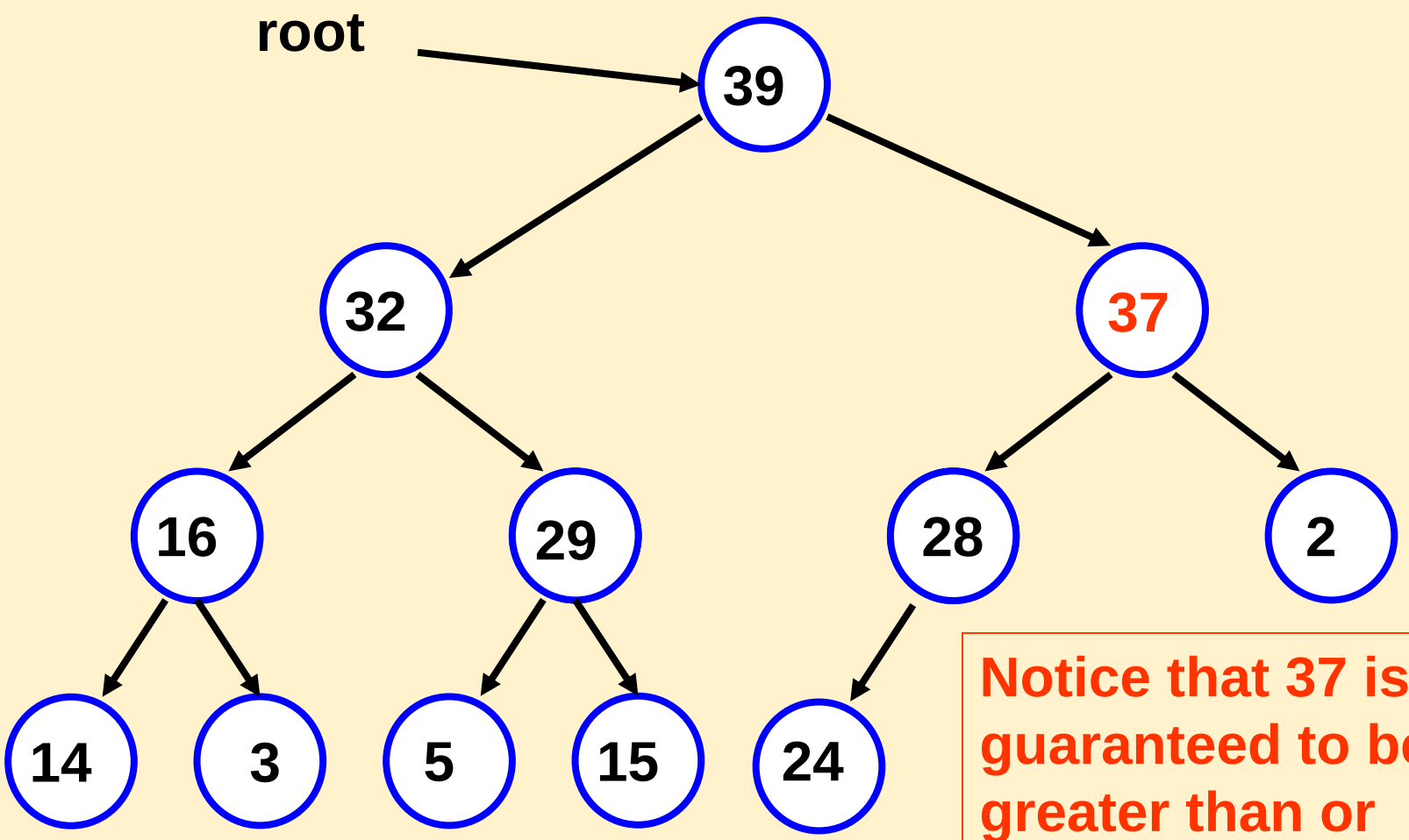
Enqueue (cont.)



Enqueue (cont.)

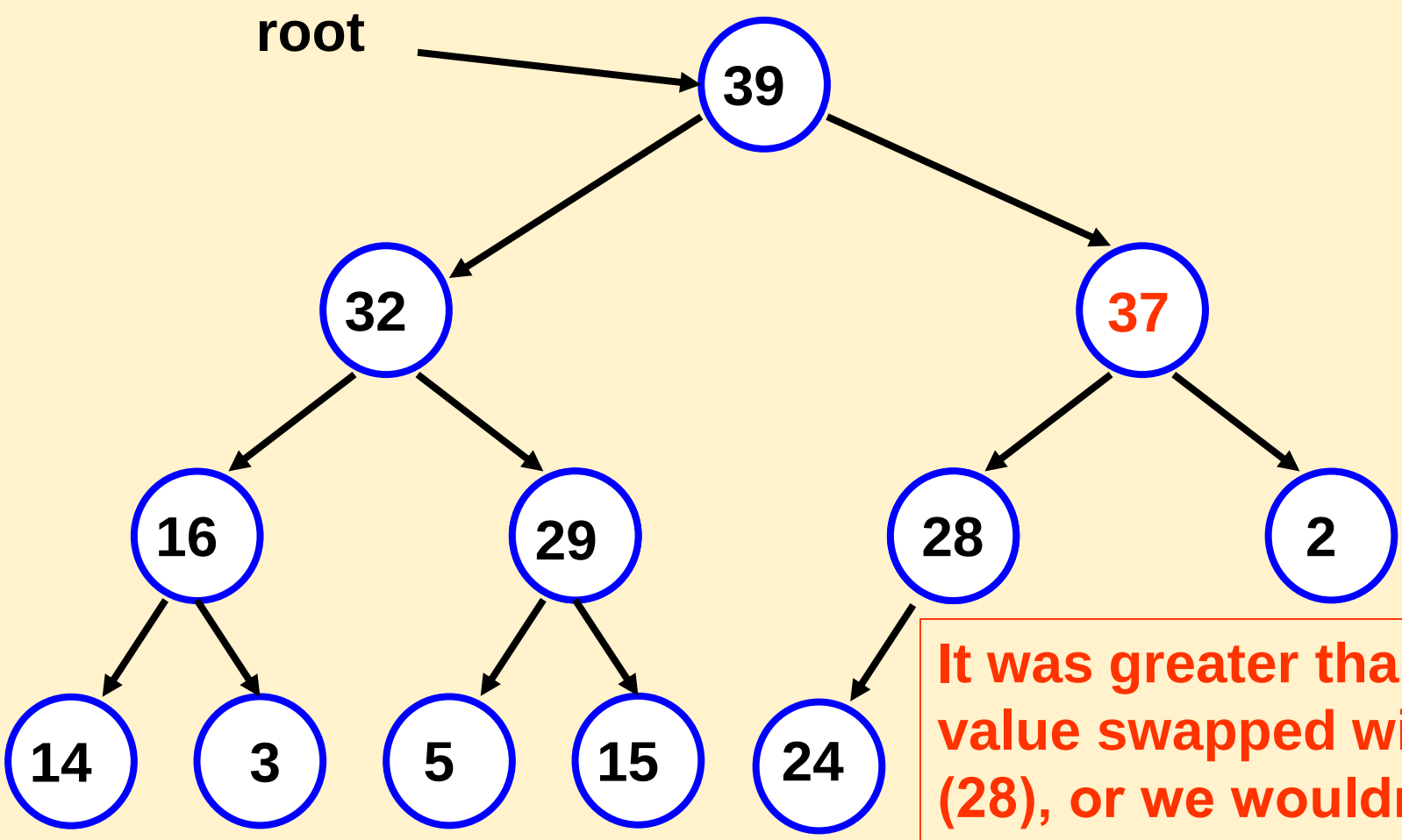


Enqueue (cont.)



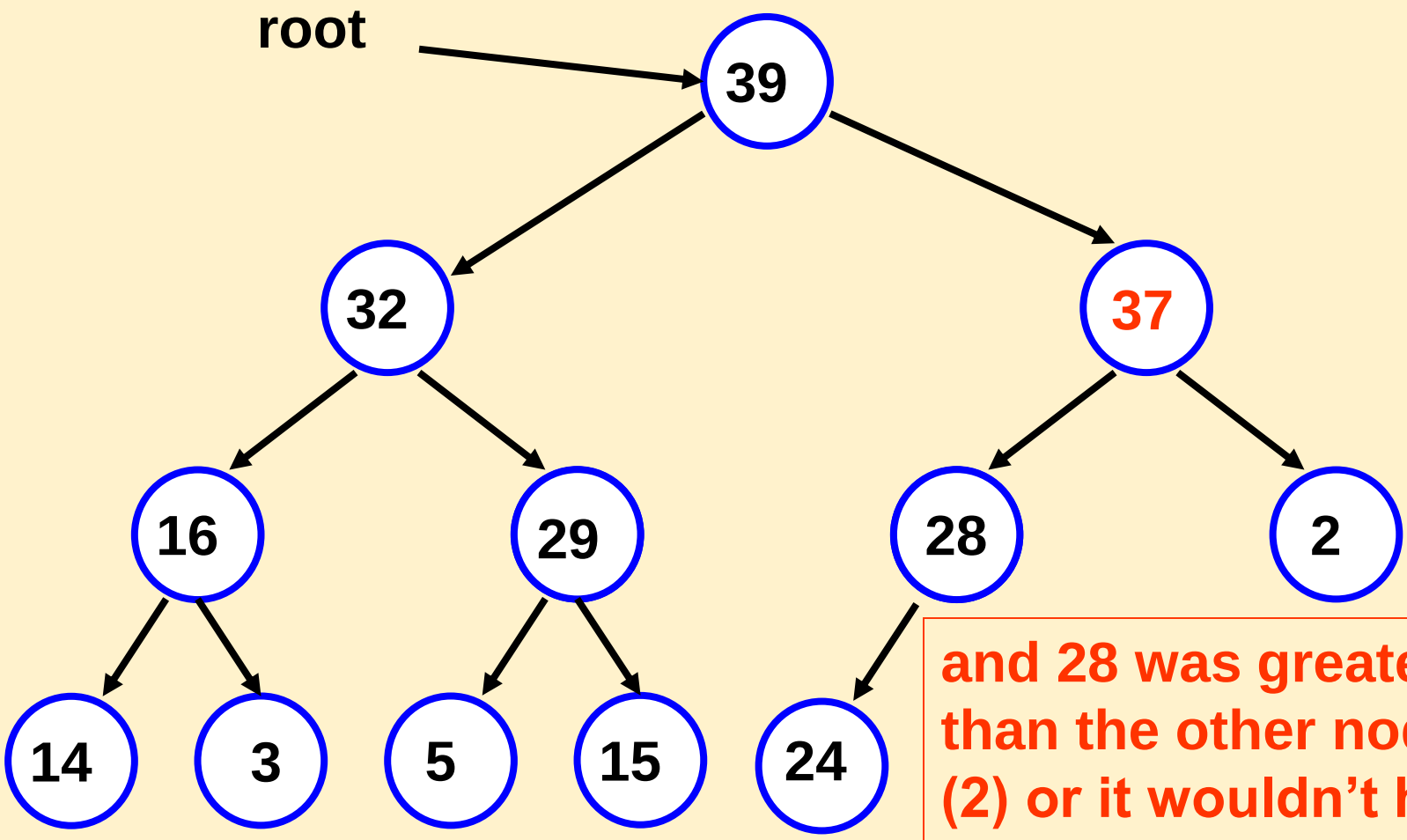
Notice that 37 is guaranteed to be greater than or equal to its children

Enqueue (cont.)



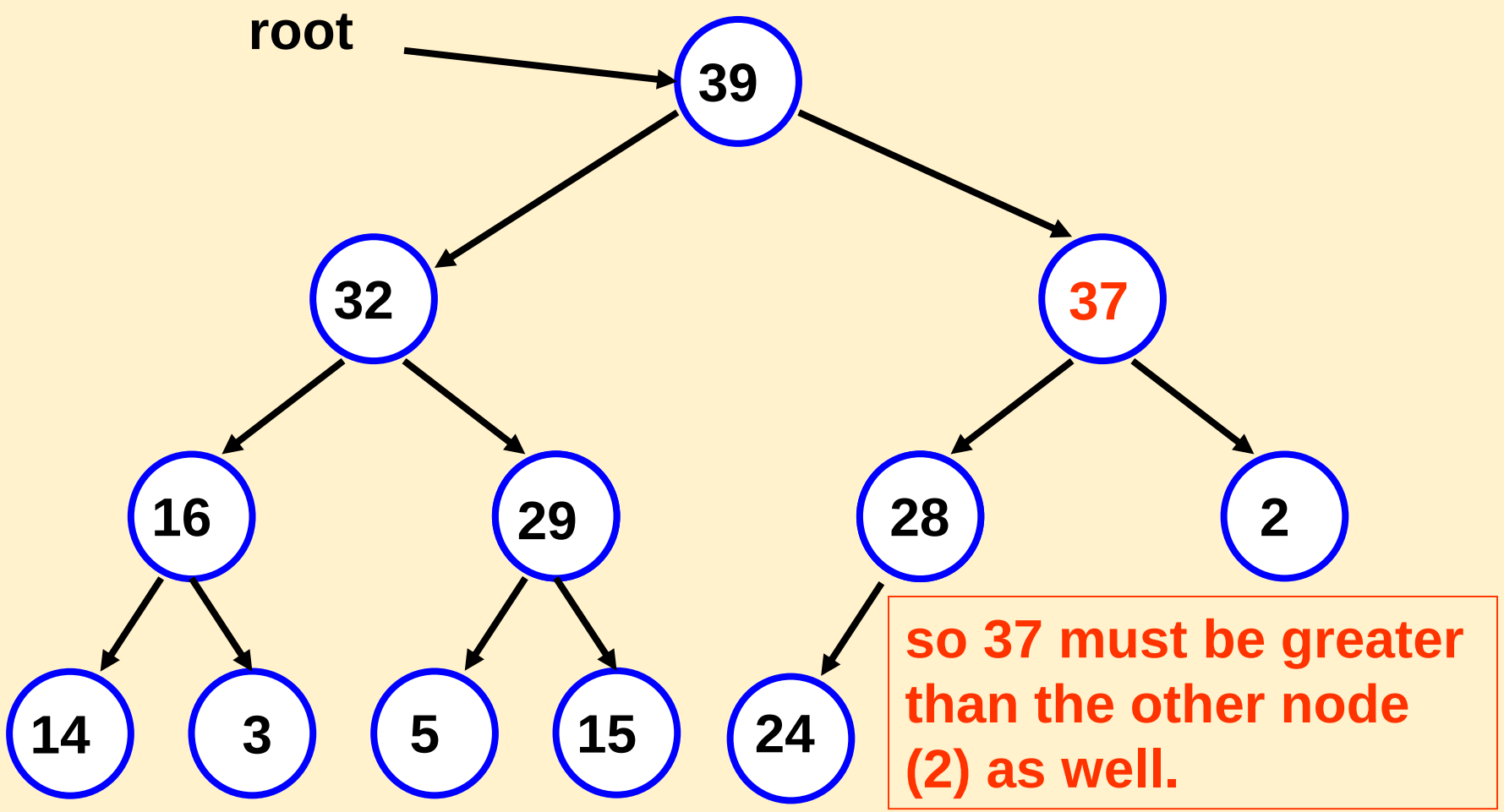
It was greater than the value swapped with (28), or we wouldn't have swapped…

Enqueue (cont.)

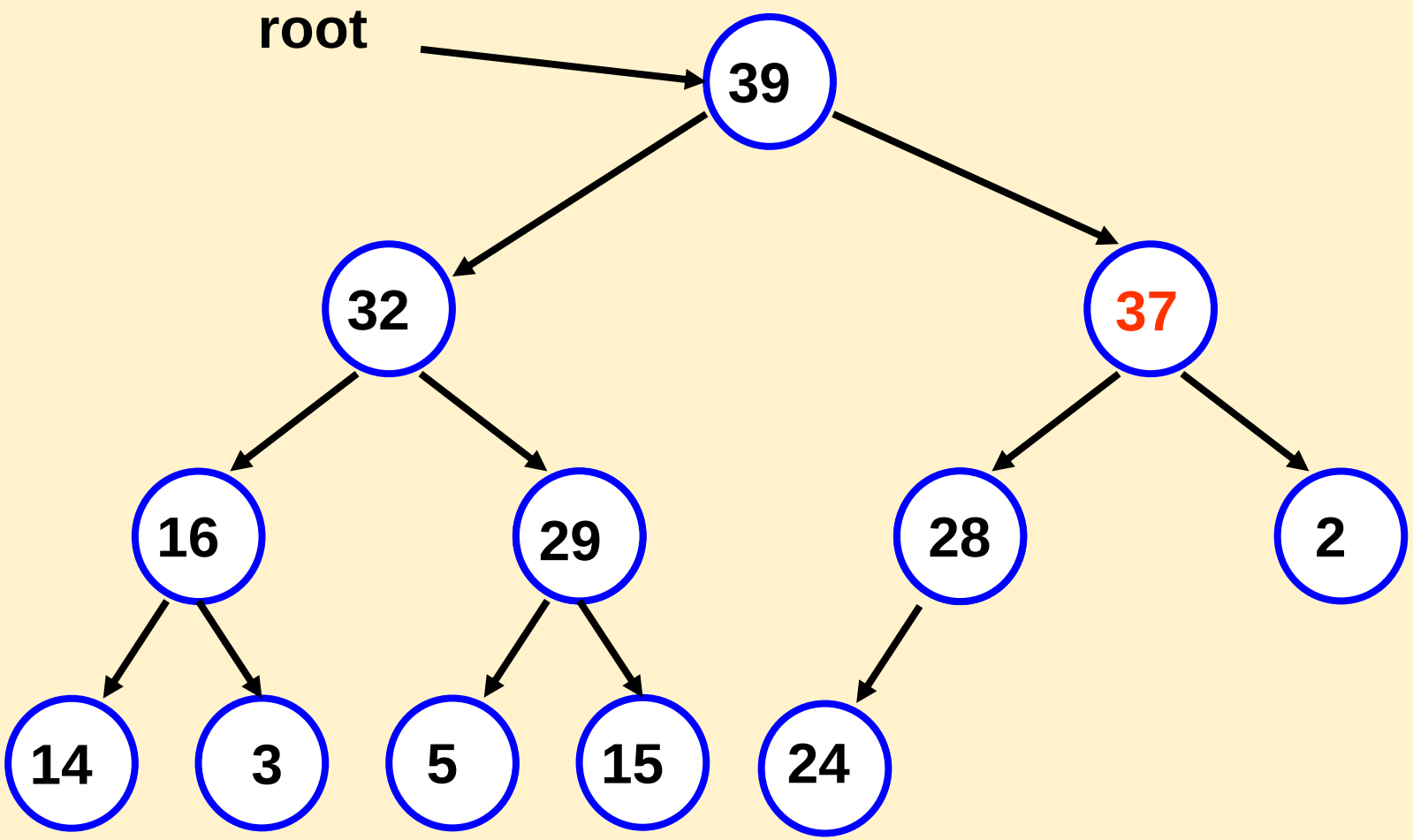


and 28 was greater than the other node (2) or it wouldn't have been a heap…

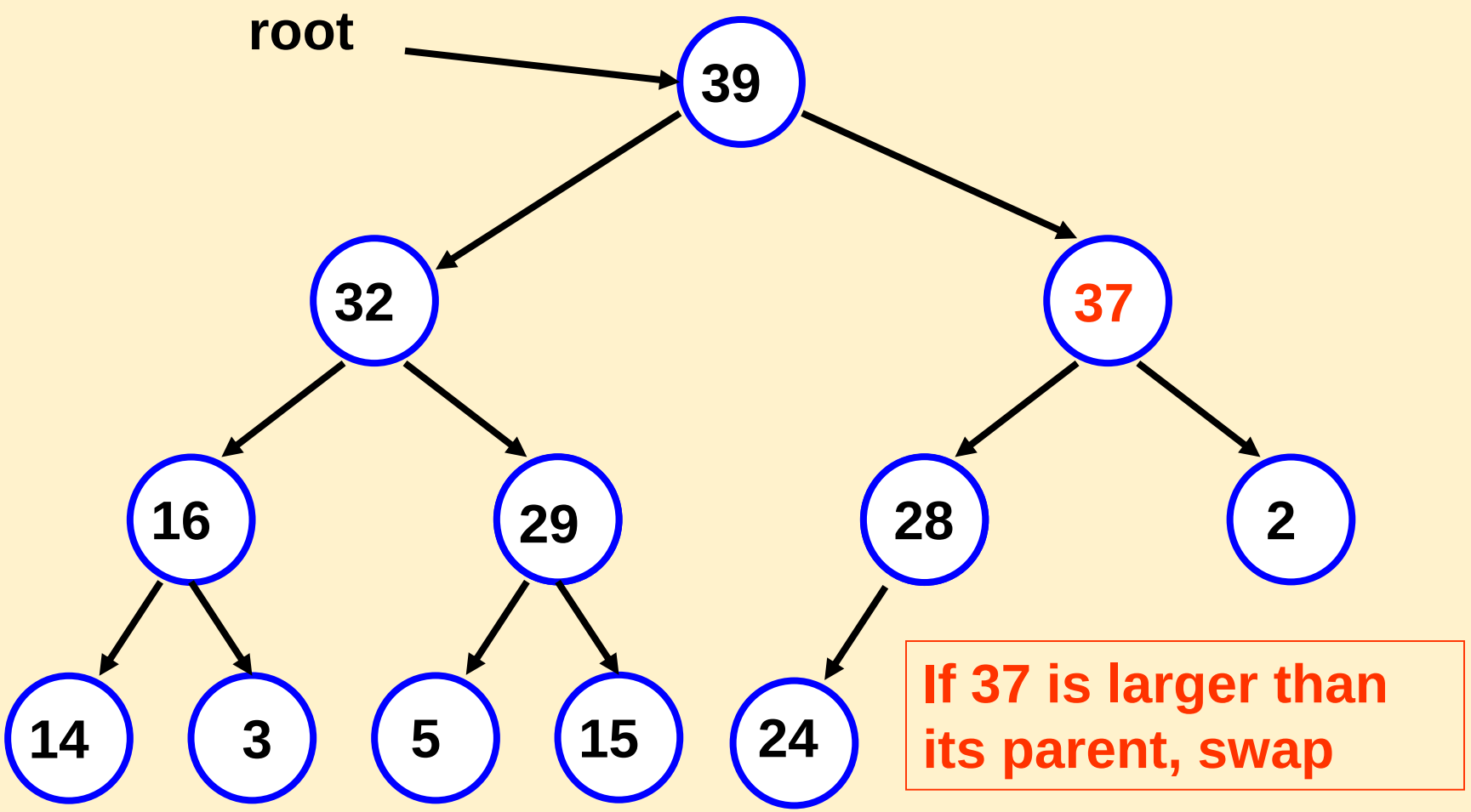
Enqueue (cont.)



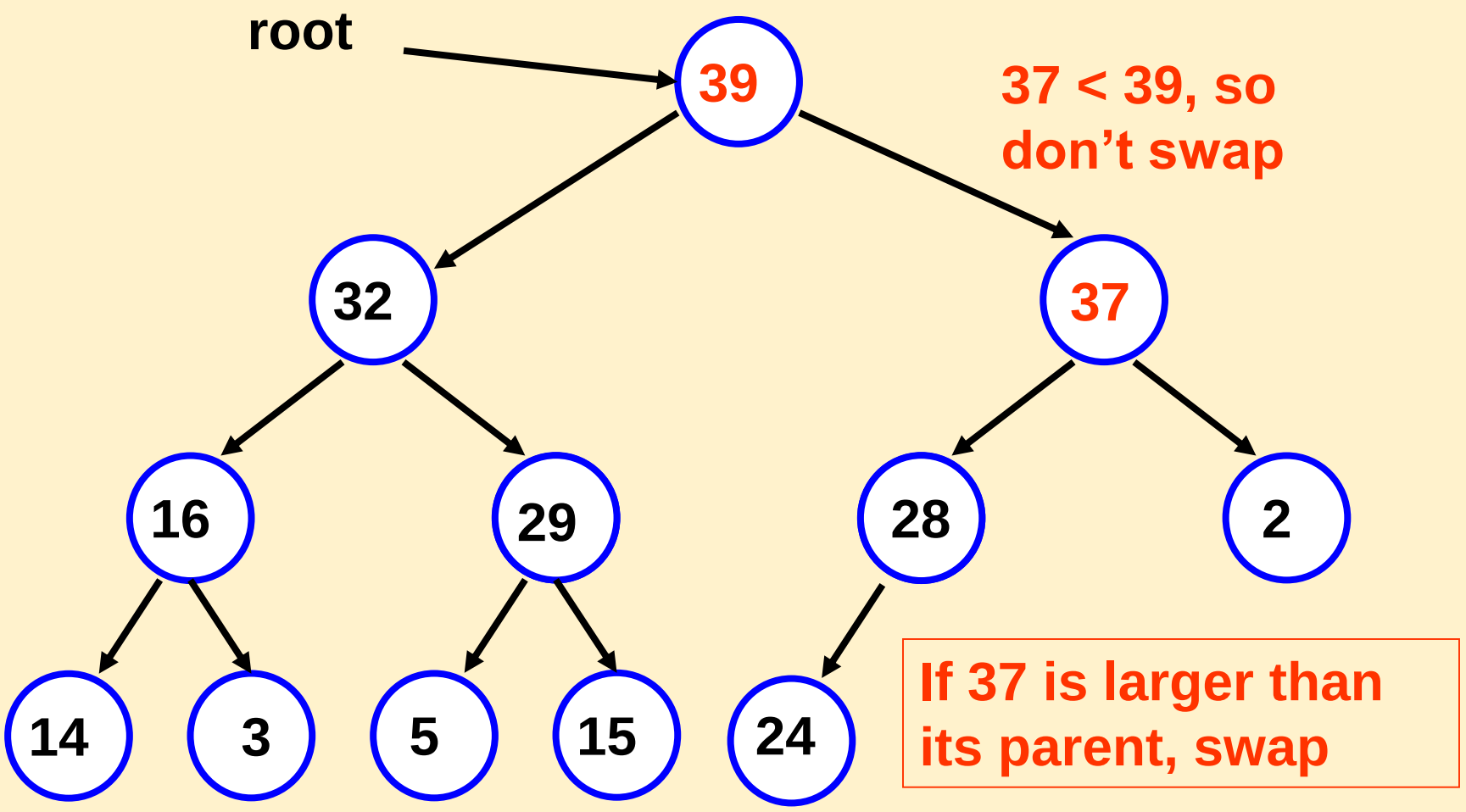
Enqueue (cont.)



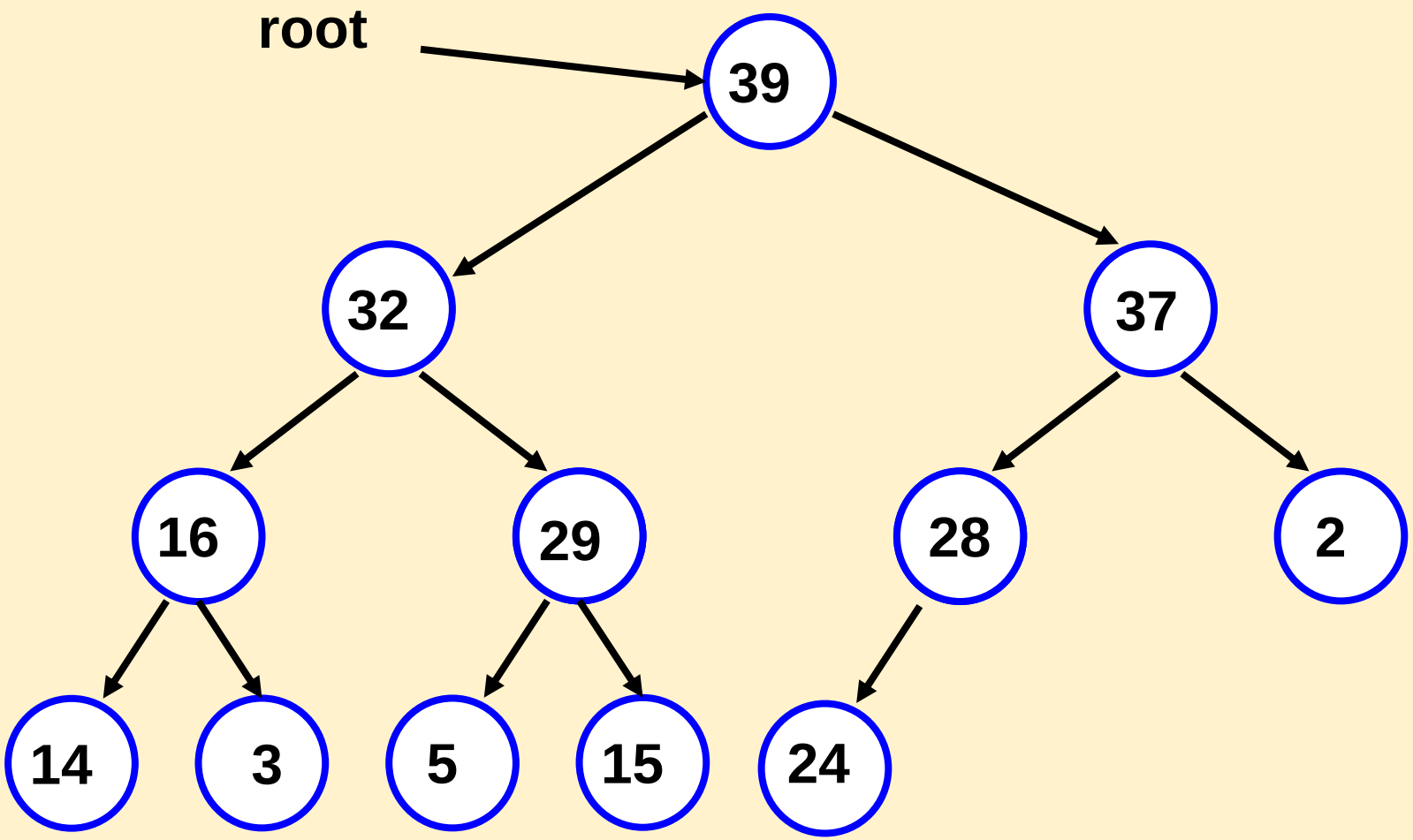
Enqueue (cont.)



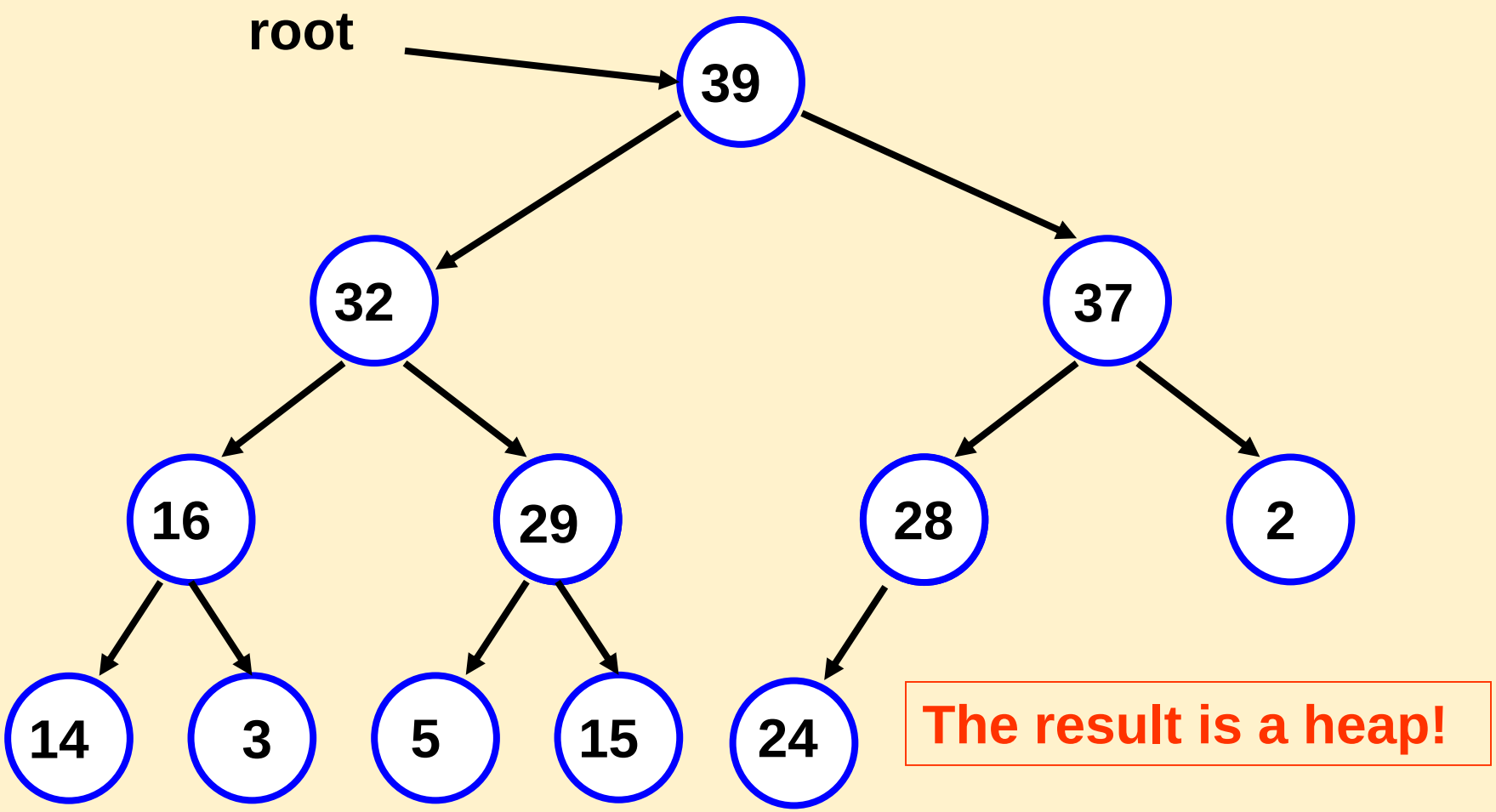
Enqueue (cont.)



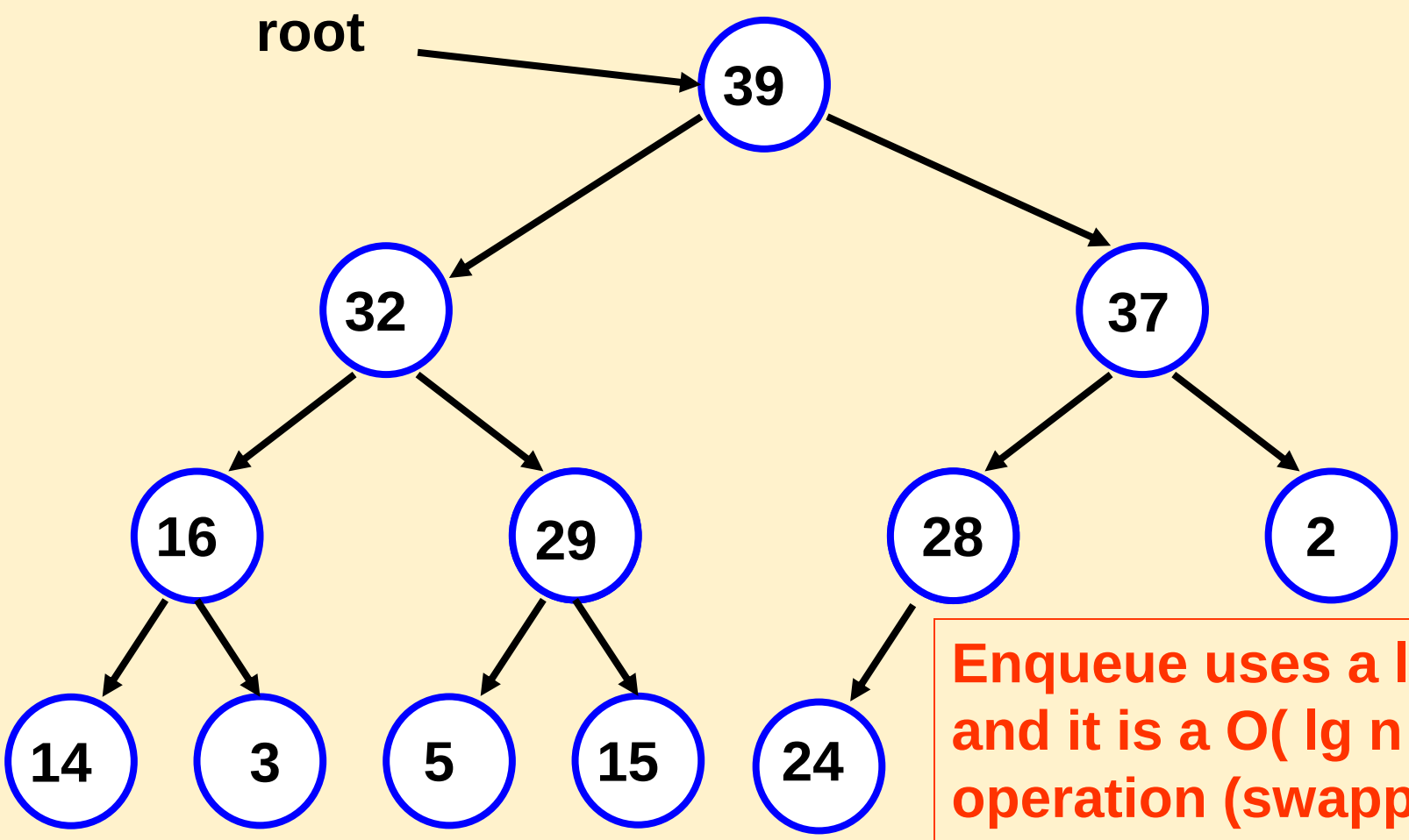
Enqueue (cont.)



Enqueue (cont.)



Enqueue (cont.)



Enqueue uses a loop, and it is a O(lg n) operation (swapping in reverse)

Implementing ReheapUp

- If the node is the root of the heap, do nothing
- If the node is greater than its parent, swap the node and its parent
- Repeat the previous step recursively until the node is the root of the heap or it is less than or equal to its parent
- Index of parent = (index of node - 1) / 2