Algorithm 3.1.49 Optimized insertion sort using the binary search technique. The time complexity for regular insertion sort $\mathcal{O}(n^2)$ reduced to $\mathcal{O}(n \log n)$.

```
procedure BINARY INSERTION SORT($a_1, a_2, \ldots, a_n$: list of comparable entries)
    for $j = 2$ to $n$ do
        $start \leftarrow 1$
        $end \leftarrow j$
        while $start < end$ do                                    ▷ Binary search.
            $midpoint \leftarrow \lfloor \frac{(start+end)}{2} \rfloor$
            if $a_j > a_{midpoint}$ then
                $start \leftarrow (midpoint + 1)$
            else
                $end \leftarrow midpoint$
            end if
        end while                                                 ▷ End binary search.
        $element \leftarrow a_j$                                  ▷ Insert.
        for $k = 0$ to $(j - start - 1)$ do
            $a_{(j-k)} \leftarrow a_{(j-k-1)}$
        end for
        $a_{start} \leftarrow element$                            ▷ End insert.
    end for
    return $a_1, a_2, \ldots, a_n$
end procedure
```