Algorithm 3.1.28 Locate an element from a finite list of increasing integers by recursively splitting the list into four search space partitions.

```
procedure QUATERNARY SEARCH($term$: integer; $a_0$, $a_1$, $\ldots$, $a_n$: finite
  list of increasing integers; $index = 0$: integer)
    $endpoint \leftarrow \lfloor \frac{n}{4} \rfloor$
    if $endpoint = 0$ then                          ▷ Bottom of recursion stack reached
        if $term = a_0$ then
            return $index$
        else if $term = a_1$ then
            return $index + 1$
        else
            return $index + 2$
        end if
    else if $term < a_{endpoint}$ then                    ▷ Recur into search partitions
        return QUATERNARY SEARCH(
                                $term$,
                                $a_0, a_1, \ldots, a_{(endpoint-1)}$,
                                $index$)
    else if $a_{endpoint} \leq term < a_{(endpoint\times 2)}$ then
        return QUATERNARY SEARCH(
                                $term$,
                                $a_{endpoint}, a_{(endpoint+1)}, \ldots, a_{(endpoint\times 2)-1}$,
                                $index + endpoint$)
    else if $a_{(endpoint\times 2)} \leq term < a_{(endpoint\times 3)}$ then
        return QUATERNARY SEARCH(
                                $term$,
                                $a_{(endpoint\times 2)}, a_{(endpoint\times 2)+1)}, \ldots, a_{(endpoint\times 3)-1}$,
                                $index + (endpoint \times 2)$)
    else if $a_{(endpoint\times 3)} \leq term$ then
        return QUATERNARY SEARCH(
                                $term$,
                                $a_{(endpoint\times 3)}, a_{(endpoint\times 3)+1}, \ldots, a_n$,
                                $index + (endpoint \times 3)$))
    else
        return $\perp$                               ▷ The term was never in the list
    end if
end procedure
```