Algorithm 3.1.27 Locate an element from a finite list of increasing integers by splitting the list into three search subspaces.

```
1: procedure TERNARY SEARCH(term: integer; a_0, a_1, ..., a_n: finite list of increasing integers; index = 0: integer)
2:     endpoint ← ⌊n/3⌋
3:     if endpoint = 0 then                          ▷ Bottom of recursion stack reached
4:         if term = a_0 then
5:             return index
6:         else
7:             return index + 1
8:         end if
9:     else if term < a_endpoint then                ▷ Recur into search partitions
10:        return TERNARY SEARCH(
                                  term,
                                  a_0, a_1, ..., a_(endpoint−1),
                                  index)
11:    else if a_endpoint ≤ term < a_(endpoint×2) then
12:        return TERNARY SEARCH(
                                  term,
                                  a_endpoint, a_(endpoint+1), ..., a_(endpoint×2)−1,
                                  index + endpoint)
13:    else if a_(endpoint×2) ≤ term then
14:        return TERNARY SEARCH(
                                  term,
                                  a_(endpoint×2), a_(endpoint×2)+1, ..., a_n,
                                  index + (endpoint × 2))
15:    else
16:        return 0                                  ▷ The term was never in the list
17:    end if
18: end procedure
```