Input: Sorted array, size of array and element to search for
Output: Index of element in array if found, or -1

```
procedure BINARY_SEARCH(A, n, el)
    L ← 0
    R ← n − 1
    while L ≤ R do
        m ← ⌊(L + R)/2⌋
        if A[m] = el then return m
        else if A[m] < el then
            l ← m + 1
        else
            r ← m − 1
    return −1
```

Input: Function, left and right bounds of function, precision criterion
Output: x value of function's maximum (can be adapted to minimum)
▷ Note: Used for unimodal functions.

```
procedure TERNARY_SEARCH(f, left, right, precision)
    if |right − left| < precision then
        return (left + right)/2

    left_third ← (2 ∗ left + right)/3
    right_third ← (left + 2 ∗ right)/3

    if f(left_third) < f(right_third) then
        return Ternary_Search(f, left_third, right, precision)
    else
        return Ternary_Searcch(f, left, right_third, precision)
```