
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)

$L \leftarrow 0$

$R \leftarrow n - 1$

while $L \leq R$ **do**

$m \leftarrow \lfloor (L + R)/2 \rfloor$

if $A[m] = el$ **then return** m

else if $A[m] < el$ **then**

$l \leftarrow m + 1$

else

$r \leftarrow m - 1$

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  $(left + right)/2$   
  
   $left\_third \leftarrow (2 * left + right)/3$   
   $right\_third \leftarrow (left + 2 * right)/3$   
  
  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)
```

Alternative base case:

```
if  $|right - left| < 3$  then  
  return Maximum element in range  $[left, right]$ 
```
