Ejercicio 1

Input → arreglo ordenado de números positivos A[1...n] y un número x

output → Un índice i tal que x = A[i] si x está en A y -1 si no está

#Algoritmo planteado donde A es el arreglo, low el índice menor y high el mayor, además de x que es el número que se busca dentro del arreglo.

```
binary-Search (A, low, high, x)                          cost      times
    if (low == high)                                      c         1
        if (A[low] == x)                                  c         1
            Return low                                    c         1
        else
            Return -1                                     c         1
    else
        mid = L(low + high)/2 ⌋                           c         1
        if (A[mid] == x)                                  c         1
            Return mid                                    c         1
        else
            if (A[mid] > x)                               c         1
            Return binary-Search (A, low, mid, x)         T(n/2)    } 1
            else
                Return binary-Search (A, mid+1, high, x)  T(n/2)    }
                                                          T(n/2) + c
```

El tiempo de ejecución es $T(n/2) + C$ que por teorema maestro podemos decir que es $\Theta(n^0 \log n) \rightarrow \Theta(\log n)$

El número mínimo posibles de operaciones es:

\# caso donde x se encuentra justo en el medio
\# caso en el que el arreglo es de tamaño 1

→ 1 operación

El número máximo de operaciones posibles es:

\# caso en el que x se encuentra en cualquiera de los extremos del arreglo

→ $\log n$