Algo (A, p, r)

```
if (n==1)                                   C1
    return (A[n]==n)                        C2

m = ⌊(p+r)/2⌋                               C3

if A[m] == m                                C4
    return true                             C5

else if (A[m] > m)                          C6
    return Algo(A, p, m-1)                  T(n/2)

else return Algo(A, m+1, r)                 T(n/2)
```

can't add all the costs together! (because of if-else-)

$$T(n) = T\left(\frac{n}{2}\right) + \Theta(1)$$

* $a=1, b=2$

$n^{\log_2 1} = n^0 = c$

$$T(n) = \Theta(c \cdot \lg n) = \Theta(\lg n)$$

Upper Bound:

$T(n) \le d \lg n$ if $d \ge c$

Lower Bound:

$T(n) \ge d \lg n$ if $d \le c$