2.3-5 Write pseudocode for binary search. Argue that the worst-case running time of binary search is $\Theta(\lg n)$.

bsearch(A, $\ell$, r, v)

input: array $A$ indexed from $\ell$ to $r$ with items sorted from smallest to largest and item $v$.
output: bsearch returns a location of item $v$ in array $A$; if $v$ is not found, -1 is returned.

if ($\ell > r$) then return (-1);
$m = \lfloor(\ell + r)/2\rfloor$;
if (A[m] == v) then return(m);
if (A[m] < v) then return(bsearch(a, m+1, r, v));
else return(bsearch(a, $\ell$, m-1, v));

Examining the algorithm yields the following observations:

- The non-recursive work takes constant time $C$.
- The single recursive call has input half the size of the original.
- The algorithm terminates when the array has size no larger than one.

Therefore, we have the following recurrence: $T(n) = T(n/2) + C$, and we conclude that $T(n) = k \cdot (C + 1)$, where $k$ is the number of recursive calls. Also, we note that, if the size of array $A$ is $n$, then $(n/2^k) = 1$, so $k = \lg n$, and $T(n) = (C + 1) \cdot \lg n$. 