Problem: Given a circularily sorted array find element t. There are no duplicates.
Algorithm circularBinarySearch
Input: an array A, a target t, left=1 and right = n where n is length of A.
Output: the index at which t is found.

Start

$m = \lfloor \frac{right+left}{2} \rfloor$
if A[m] = t then
  return m
fi
if A[right] > A[m]
  if A[m] < t ∧ t ≤ A[right] then
    return circularBinarySearch(A, t, m+1, right)
  fi
  else then
    return circularBinarySearch(A, t, left, m-1)
  esle
fi
else then
  if t < A[m] ∧ t ≥ A[left] then
    return circularBinarySearch(A, t, left, m-1)
  fi
  else then
    return circularBinarySearch(A, t, m+1, right)
  esle
esle

End
TCA: $T(n) = T(\frac{n}{2}) + O(1)$; $a=1, b=2, f(n) = O(1)$, watershed function $= n^{\lg 1} = n^0 = 1$. as $1 \in \Theta(1)$ then by case 2 as $k=0$ then $T(n) = \Theta(\lg n)$.

Hyp. Let $P(n)$ be the assertion that circularBinarySearch works for all inputs $right-left+1=n$.

Base case: when $n=1$ $right=left=m$ so the element is found by algorithm.

Inductive Step: We assume circularBinarySearch works as long as $right-left+1\leq k$ and must prove $right-left+1\leq k+1$ in 3 cases $A[m]=t, A[m]<t, A[m]>t$.

Case 1: $A[m]=t$

As the function work this will return m, the index of t

Case 2: $A[m]>t$

We know this half of the array is either sorted or circularily sorted. Thus t must lie to the left of m. If the recursive call works correctly so shall this call. So $n=m-1-left=\left\lfloor\frac{left+right}{2}\right\rfloor-1-left$. If $right+left$ is odd this means $\frac{right+left-1}{2}-left-1=\frac{right+left-2left-1}{2}-1=\frac{right-left-1}{2}-1 \leq right-left+1\leq k$ (II) so by trasitivity $\frac{right-left-1}{2}-1\leq k$ (I) adding $+1$ to both sides $\frac{right-left-1}{2}\leq k+1$. When even $\frac{right+left}{2}-1-left=\frac{right-left}{2}-1\leq right-left+1\leq k$ so by transitivity $\frac{right-left}{2}-1\leq k$ (III) adding 1 to both sides $\frac{right-left}{2}\leq k+1$ (IV) as $\frac{right-left-1}{2}\leq right-left+1 \wedge \frac{right-left}{2}\leq right-left+1$ so by transitivity $right-left+1\leq k+1$ (V) as $k\leq k+1$. So the recursive call must be between 1 and n and is correct by Hyp.

Case 3: $A[m]<t$

This is symmetrical to case 2. Thus $n=right-\left(\left\lfloor\frac{right+left}{2}\right\rfloor+1\right)$ so when $right+left$ is odd then $=right-\frac{right+left-1}{2}-1=\frac{right-left-1}{2}-1$ (VI). When even $=right-\frac{right+left}{2}-1=\frac{right-left}{2}-1$ (VII) so by VI, VII, I, II, III, IV and V $right-left+1\leq k+1$ as $k\leq k+1$. Thus the recursive call must lie between 1 and n and is correct by Hyp.

$\therefore$ as the Inductive step is proven for all cases then circularBinarySearch must be correct