Problem: given a sorted integer array [redacted] check if any element appears at least $\lfloor \frac{n}{2} \rfloor + 1$ times in the array

Algorithm majorityElement

Input: An array A
Output: Yes if an element is the majority of A, no otherwise

```
Start
left:=1
right:= length of A
t m:= ⌊(right+left)/2⌋
leftBinSrch (A, t, left, right) start
    m:= ⌊(right+left)/2⌋
    if (A[m]=t ∧ A[m-1]≠t) [redacted] ∨ (A[m]=t ∧ left=m) then
        return m fi
    if A[m]=t [redacted] ∨ A[m]>t then
        return leftBinSrch(A,t,left,m-1)
    fi
    else then
        return leftBinSrch(A,t,m+1,right)
    esle
end
rightBinSrch (A, t, left, right) Start
    if A[m] = t [redacted] ∨ A[m]<t then
        return rightBinSrch(A,t,m+1,right) fi
    else then
        return rightBinSrch(A,t,left,m-1)
    esle
    if (A[m]=t ∧ A[m+1]≠t) [redacted] ∨ (A[m]=t ∧ m=right) then
        return m fi end
m:= leftBinSrch (A,t,left,right)
k:= rightBinSrch(A,t,left,right)
if (k-m) ≥ (⌊n/2⌋+1) then
    return yes
fi
else then
    return no
esle
End
```

Thm: leftBinSrch finds the leftmost target element. Proof by induction:

Hyp. let $P(n)$ be the assertion that leftBinSrch works for all inputs $right-left+1 \le n$.

Base Case: when $n=1$ $left=right=m$ meaning index $m$ is returned as the algorithm works

Inductive Step: We assume leftBinSrch works as long as $right-left+1 \le k$ and must prove $right-left+1 \le k+1$ in 3 cases the leftmost target is found, $A[m]=t \vee A[m]>t$, $A[m]<t$

Case 1: leftmost target is found
as the function works $m$ is returned

Case 2: $A[m]=t \vee A[m]>t$
$t$ must lie between $A[left]$ and $A[m-1]$ thus $n=(m-1)-left+1=(\lfloor\frac{right+left}{2}\rfloor-1)-left+1$ if $right+left$ is even then $n=\frac{right+left}{2}-\frac{2left}{2}-1+1=\frac{right-left}{2}\le right-left+1$. When odd $n=\frac{right+left-1}{2}-\frac{2left}{2}-1+1=\frac{right-left-1}{2}$ →
→ $\le right-left+1$ and $right-left+1\le k$. Then the recursive call is made an interval 0 to $k$ and is correct by Hyp thus working an interval $k+1$.

Case 3: $A[m]<t$
This is symmetrical to case 2 thus $n=right-(m+1)+1=right-(\lfloor\frac{right+left}{2}\rfloor+1)+1$ if $right+left$ is even then $n=\frac{2right}{2}-\frac{right+left}{2}-1+1=\frac{right-left}{2}\le right-left+1$. When odd $n=\frac{2right}{2}-\frac{right-left+1}{2}-1+1=\frac{right-left+1}{2}\le right-left+1\le k$.
Then the recursive call is made on interval 0 to $k$ and is correct by Hyp thus working on interval $k+1$.

∴ As the inductive step works in all cases leftBinSrch works in all cases.

As well by the same math and nearly similar cases rightBinSrch also works

∴ as both the leftmost and rightmost occurrences of $t$ are captured we return yes if $t$ is a majority element and no if otherwise.