Problem: Given a nearly sorted array s.t. each of the n elements can be misplaced by ± 1 positionally search for a target t if it exists and return index if it does. else return -1.

Algorithm misBin: Input: an array A, a target t, left=1, right=n

Output: Index of t in A if it exists, else -1.

```
Start

if left > right then
   return -1
fi
m := ⌊(left+right)/2⌋
if A[m-1] = t then
   return m-1
fi
elif A[m] = t then
   return m
fide
elif A[m+1] = t then
   return m+1
file
elif A[m] > t then
   return misBin(A, t, left, m-1)
file
else
   return misBin(A, t, m+1, right)
esle
```

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

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

Base Case: when $n=1$ $right = left = m$, if $A[m] = t$ then m is returned. If $A[m] \neq t$ then the else statement happens making $left > right$ and returning -1. Thus the base case works.

Inductive Step: We assume misBin works as long as $right - left + 1 \leq k$ and must prove $right - left + 1 \leq k+1$. to do this we must prove 6 cases, where t does not exist in A, $A[m-1] = t$, $A[m] = t$, $A[m+1] = t$, $A[m] > t$ and $A[m] < t$.

Case 1: t does not exist:
As each element in A is ± 1 away from or is exactly in the correct position if the array A is traversed fully and t is not found at $A[m-1]$, $A[m]$ or $A[m+1]$ the else will trip causing $left > right$ returning -1 correctly.

Case 2, 3, 4: $A[m-1] = t$, $A[m] = t$, $A[m+1] = t$:
As the function works it is obvious that if any equals t the index will be returned.

Case 5: $A[m] > t$:
We know the array is sorted within ± 1 of the correct indicies. As well $A[m-1] \wedge A[m+1] \neq t$. Thus if t exists it must lie to the left of m between $A[left]$ and $A[m-1]$. If the recursive call works correctly so shall this call. So $n = m - 1 - left$. when $right + left$ is odd this means $\frac{left + right - 1}{2} - 1 - left = \frac{left + right - 1 - 2left}{2} - 1 = \frac{right - left - 1}{2} - 1$. when even this means $\frac{right + left}{2} - 1 - left = \frac{right + left - 2left}{2} - 1 = \frac{right - left}{2} - 1$ both of which are $\leq right - left + 1$ so by transitivity of $\leq$ $right - left + 1 = k + 1 > 0$. So the recursive call must be between 1 and n and is correct by Hyp.

Case 6 $A[m] < t$:
This is symmetrical to case 5. Thus we have $right - (m+1) = right - \left\lfloor \frac{right + left}{2} \right\rfloor + 1$ when $right + left$ is even $\frac{2right - right - left}{2} + 1 = \frac{right - left}{2} + 1$. when odd. $\frac{2right - right - left + 1}{2} + 1 = \frac{right - left - 1}{2} + 1$ both of which are $\leq right - left + 1$ so by transitivity of $\leq$ $right - left + 1 = k + 1 > 0$. so the recursive call must be in the array and is correct by Hyp.

$\therefore$ as the Inductive Step works $\forall$ cases we can conclude misBin is correct $\square$