Let $A[1..n]$ be a sorted array of n *distinct* integers. Some of these integers might be positive, negative, or zero.

(a) Describe an $O(\log n)$ algorithm to decide if there exist some index i such that $A[i] = i$.

Hint. *Consider the array* $B[i] = A[i] - i$.

(b) Now, suppose we know that $A[1] > 0$. Describe an $O(1)$ algorithm to decide if there exist some index i such that $A[i] = i$.

Hint. *Again, consider the array* $B[i] = A[i] - i$.

Rubric.

- You should justify why your algorithm is correct and why they run in the allocated time complexities (or faster!).
- This task will form part of the portfolio.
- Ensure that your argument is clear and keep reworking your solutions until your lab demonstrator is happy with your work.

Solution.

(a) Define a new array B, where $B[i] = A[i] - i$ for all $i = 1, \ldots, n$. From here, we can simply perform a binary search starting from $i = \left\lfloor \frac{n}{2} \right\rfloor$, looking for any element that equals zero. In the original array, if $A[i] = i$, then $B[i] = 0$. Since A is sorted and made with distinct integers (always increasing), it means that B is non-decreasing and hence can be searched in $O(\log n)$ time.

(b) Consider the array B from the previous part. If $B[1] = 0$, then we must have $A[1] = 1$ and hence $i = 1$ is a valid index that meets the criteria. Now, A must be increasing, meaning that B is non-decreasing. If $B[1] \neq 0$, then $B[1] > 0$ since $A[1] > 0$, and so no preceeding term can be 0. This means that, unless $B[1] = 0$, there cannot exist such an index.