CS381 Homework 0 Problem 1

Connor Couetil

January 18, 2026

1 Exercise 1.3.1

For a some index i in an array with distinct elements, A[i] is always greater than the previous element (where the previous element of the first element is the last), except at the rotation index, as the array is in ascending order before rotation. So, we can perform a binary search whose match condition for an element is whether the relation A[i] > A[n + i - 1 mod n] holds.

If it does not hold, we have found the rotation index. If it does hold, we must search either the earlier or later elements relative to our current element. Recognize the element at the rotation index must be less than the first element in the array, it denotes the start of the ascending order. Then, if the current element is greater than the first element, the rotation index is to the right of the current element, otherwise it's to the left. We'll continue this search until we've found the rotation index.

There is a special case, when the array has not been rotated. Only in this case is the first element less than the last element, and the index is 1.

Now for the recursive algorithm. We'll assume the special case will not occur and that Midpoint is a routine that finds the middle-most integer between two integers in constant time.

Recursive Algorithm

```
RotationIndex(array, start, end)

1. If $start == end$
    A. Return 1
2. $midpoint = \lceil (end - start + 1)/2 \rceil + start$
3. $neighbor = (midpoint - 1)$ mod Length($array$)
4. If $array[midpoint] < array[neighbor]$
    A. Return $midpoint$
5. If $array[midpoint] > array[1]$
    A. Return Search($array$, $midpoint$, $end$)
6. Return Search($array$, $start$, $midpoint$)
```

Runtime Analysis O(log n) due to recurrence relation $\approx S(n) = S(n/2) + c$, $S(1) = C$