33_Search in Rotated Sorted Array

Since the input array is sorted in ascending order,

if it's rotated at some pivot, we surely would find a gap,

or an inflection point in the rotated array.

The index of this gap (or inflection point) is the pivot position.

Version 1: Traverse the input array to see if the target exites.

Time Complexity: O(n); n == # of elements in array, too slow.

Version 2: 2 times binary search

① find the rotated position;

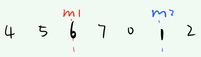
② find the target according to two ascending parts of the array.

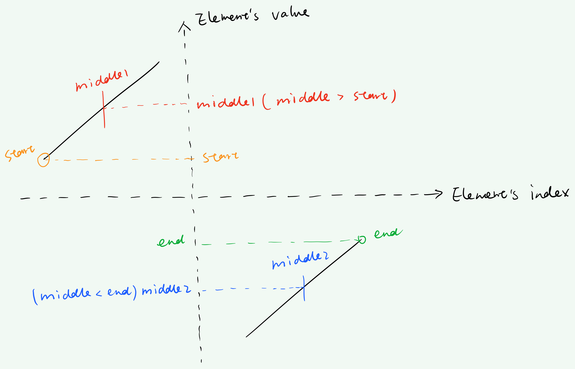
Time Complexity: O(log n + log n) ~ O(log n), n == # of elements in array

Version 3: One time binary search.

There're two possibilities of the middle point / middle > start ⇒ middle 1

\ middle < end ⇒ middle 2





1. For middle 1

① if start <= target <= middle 1

if ↕ else

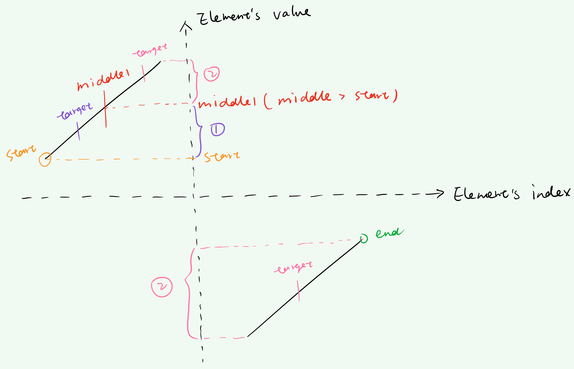
Let binary search's end = middle 1

② if target > middle 1

Or

target < start

Let binary search's start = middle 1



2. For middle 2

① if middle 2 <= target <= end

if ↕ else

Let binary search's start = middle 2

② if target < middle 2

Or

target >= start

Let binary search's end = middle 2