

# Divide & Conquer <sup>(potential)</sup> Test Questions.

1. Number of rotations of a sorted list.

Eg Input:  $[13, 14, 1, 8, 9, 11]$

Output: 2

Why? The list is result of 2 rotations of the sorted list:

$1, 8, 9, 11, 13, 14$   
 $14, 1, 8, 9, 11, 13$   
 $13, 14, 1, 8, 9, 11$

← rotations are in this direction only

Devise a Divide & Conquer algorithm that determines the number of rotations away from sorted a list is.

... And have it work in  $O(\lg n)$

2. Search for an element in a circularly sorted array

Eg find 4 in [6, 9, 10, 1, 3, 4]

... in  $O(\lg n)$

[do it directly - don't use the result of Q1]

3. Find number of occurrences of a value in a sorted array.

Eg. 5 in  $[3, 3, 5, 5, 5, 5, 5, 5, 5, 7, 8]$

... in  $O(\lg n)$

4. Max subarray Sum

[ 1 -3 4 2 -3 6 1 -1 2 ]

numRots(A[1..n])

if  $n \leq 1$  return 0

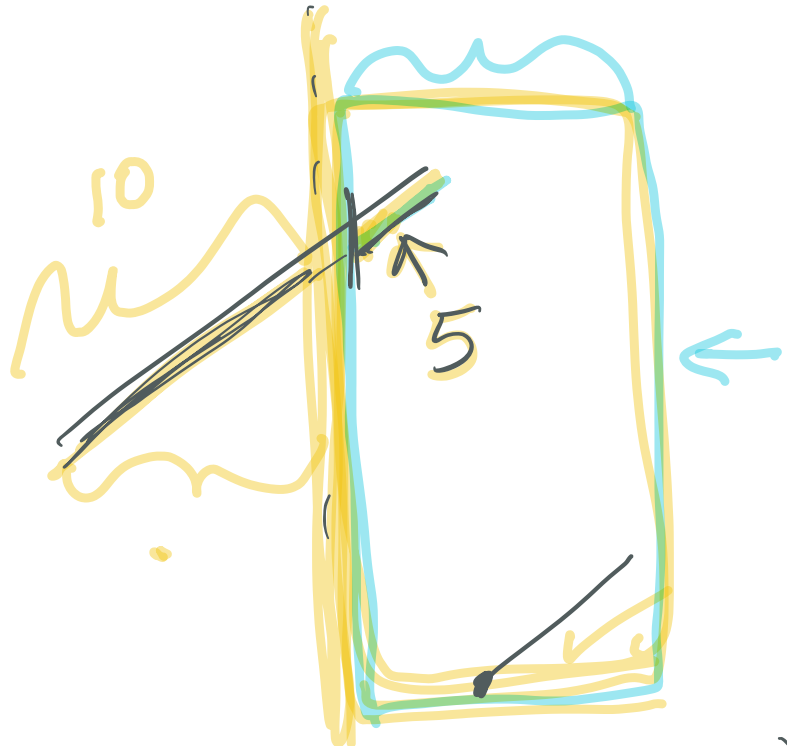


mid =  $\frac{n}{2}$

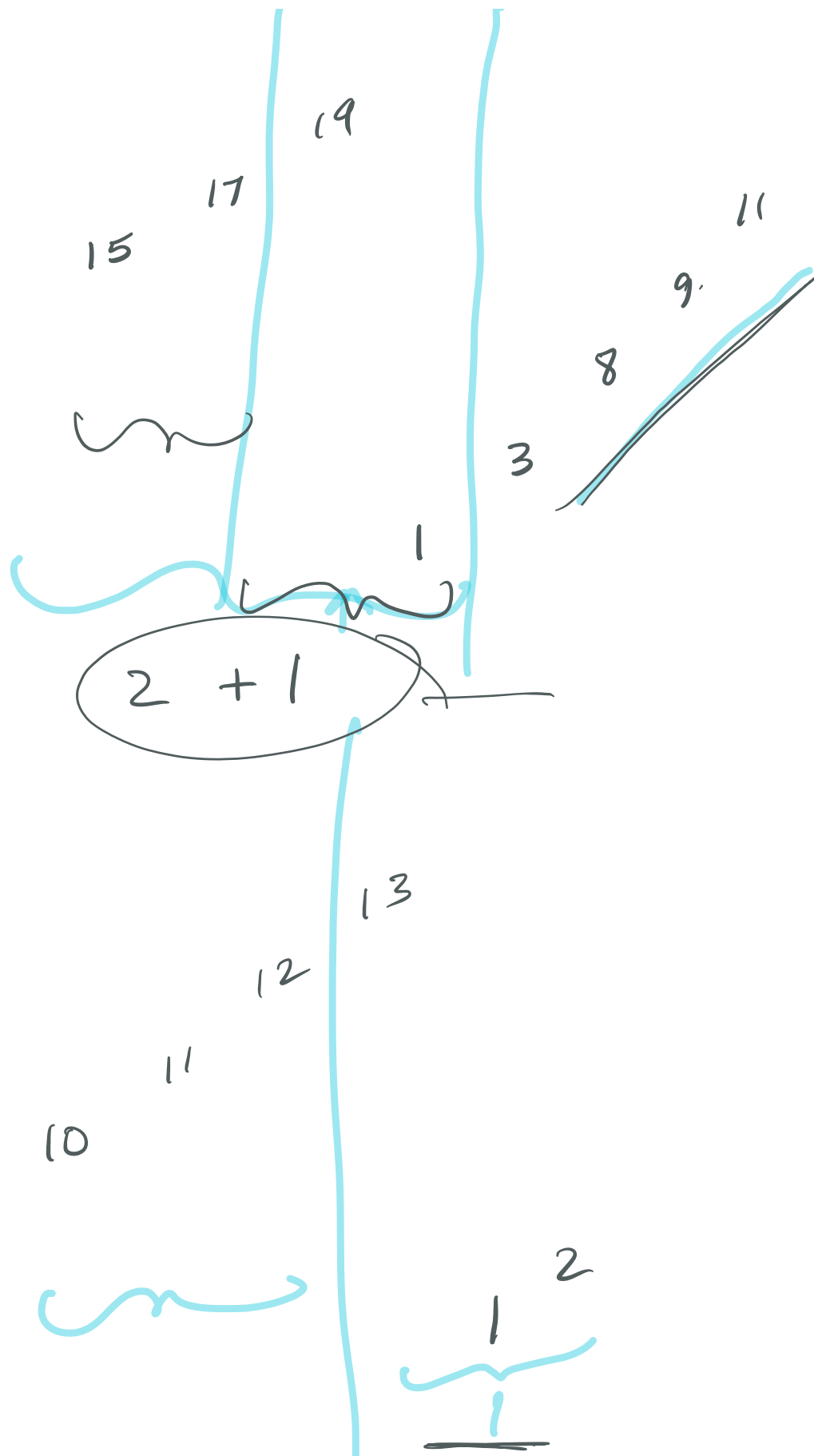
$l = A[1]$        $m = A[\text{mid}]$        $r = A[n]$

if  $l < r$  return 0

if  $m < r$   
return numRots(A[1..mid])



else return mid + numRots(A[mid+1..n])



$$\# \text{rotations} = 1 + 3 = 4.$$