222. Count Complete Tree Nodes
We will count all nodes by recursion, if the most left side is equal to right side height, we will return pow(2, height). If not, we will return 1 + count(root.left) + count(root.right). TC is O(logn * logn)

```
import math
class Solution:
    def countNodes(self, root: TreeNode) -> int:
        if not root:
            return 0
        l, r = root, root
        h_l, h_r = 0, 0
        while l:
            h_l += 1
            l = l.left
        while r:
            h_r += 1
            r = r.right
        if h_l == h_r:
            return int(math.pow(2, h_l)) - 1
        return 1 + self.countNodes(root.left) + self.countNodes(root.right)
```

1119. Remove Vowels from a String
We will iterate all chars in string and add letters which not belong to vowels to result. TC is O(n)

```
class Solution:
    def removeVowels(self, S: str) -> str:
        result = ''
        for c in S:
          if c not in 'aeiou':
            result += c
        return result
```

1165. Single-Row Keyboard
We will calculate all differences among neighbored letters' indexes and accumulate them. TC is O(n)

```
class Solution:
    def calculateTime(self, keyboard: str, word: str) -> int:
        cur_index = 0
        time = 0
        for i in word:
          next_index = keyboard.index(i)
          time += abs(next_index - cur_index)
          cur_index = next_index
```

```
        return time
```

1108. Defanging an IP Address

We will split our string by '.' and join by '[.]', TC is O(n)

```
class Solution:
    def defangIPaddr(self, address: str) -> str:
        return '[.]'.join(address.split('.'))
```

760. Find Anagram Mappings

We will use a dict to memorize all indexes of the second list, Then iterate all elements in A and append all associated element's index in B to result. TC is O(n)

```
from collections import defaultdict
class Solution:
    def anagramMappings(self, A: List[int], B: List[int]) -> List[int]:
        memo = defaultdict(int)
        result = []
        for idx, c in enumerate(B):
          memo[c] = idx
        for i in A:
          result.append(memo[i])
        return result
```