

15th Oct

LEETCODE

23 Saizam.

Day 10

1929. Concatenation of Array

- return nums + nums
- return 2 * nums
- return nums.extend(nums)
- result = [0] * 2 * len(nums)
n = len(nums)

```
for i in range(n):  
    res[i] = nums[i]  
    res[i+n] = nums[i]
```

return res.

Space Complexity $O(n)$

As, we iterate through the list once

Space $O(n)$

As, requires additional space.

All create a new list, but (extend) ~~first~~ modifies the same list in place

So $O(1)$

16th Oct

Day 11

217 Contains Duplicate

Time limit exceed

```
li = []  
for i in nums:  
    if i in li:  
        li.append(i) return True  
    else:  
        li.append(i)
```

} $O(n^2)$

return false

nums = sorted(nums) $\rightarrow O(n \log n)$

for i in range(len(nums)-1):

if nums[i] == nums[i+1]:

return True

return false

return len(set(nums)) != len(nums)

$\rightarrow O(n)$

When u do "set", python has to look at every element in nums exactly once.

So for "n" elements $\rightarrow O(n)$

19th Oct

Day 14

Python

Valid Anagram 242

```
dic1 = {}  
dic2 = {}
```

Time
 $O(n)$

```
for i in range(len(s):  
    if s[i] not in dic1:  
        dic1[s[i]] = 1  
    else:  
        dic1[s[i]] += 1
```

```
    t = t  
    return dic1 == dic2
```

OR $\text{Counter}(s) == \text{Counter}(t)$ $O(n)$

Two Sum

Brute force

```
for i in range(len(nums)):  
    for j in range(i+1, len(nums)):  
        if nums[i] + nums[j] == target:  
            return [i, j]
```

$O(n^2)$

Why dictionary \rightarrow Fast lookup $O(1)$ [number: index]
But list \rightarrow $O(n)$

\rightarrow we want to remember, what we have already seen quickly

```
dic = {}  
for i in range(len(nums)):  
    diff = target - nums[i]  
    if diff not in dic:  
        dic[nums[i]] = i  
    else:  
        return [dic[diff], i]
```

22 Oct

Day 17

Python

14 Longest Common Prefix

$n \rightarrow$ strings
 $m \rightarrow$ characters length

```
if len(strs) == 1:  
    return strs[0]  
strs = sorted(strs)  $\rightarrow O(n \cdot m \log(n))$   
first = strs[0]  
last = strs[-1]  
for i in range(len(first)):  
    if first[i] != last[i]:  
        return first[:i]  
  
return first
```