EASY MISC

PART2

PROBLEM SET

- No. 459 Repeated Substring Pattern
- No. 496 Next Greater Element I
- No. 532 K-diff Pairs in an Array
- No. 268 Missing Number
- No. 504 Base 7

NO. 459

REPEATED SUBSTRING PATTERN

PROBLEM DESCRIPTION

- Given a non-empty string check if it can be constructed by taking a substring of it and appending multiple copies of the substring together. You may assume the given string consists of lowercase English letters only and its length will not exceed 10000.

- Example 1:

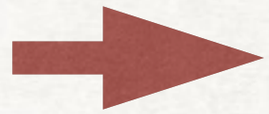
  - Input: “abab”, Output: True

  - Explanation: It's the substring "ab" twice.

IDEA

- Regex

```
for i in range(1, len(s)//2 + 1):
        pattern = "^(%s){2,}$" % (s[0:i])
        if re.search(pattern, s): return True
    return False
```



TLE

- Trick
  - Double the input string. E.g. “abab” to “abababab”
  - Remove the first and the last char. E.g. “bababa”
  - If “abab” exists in the new string, return True

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/459_Repeated_Substring_Pattern

NO. 496

NEXT GREATER ELEMENT I

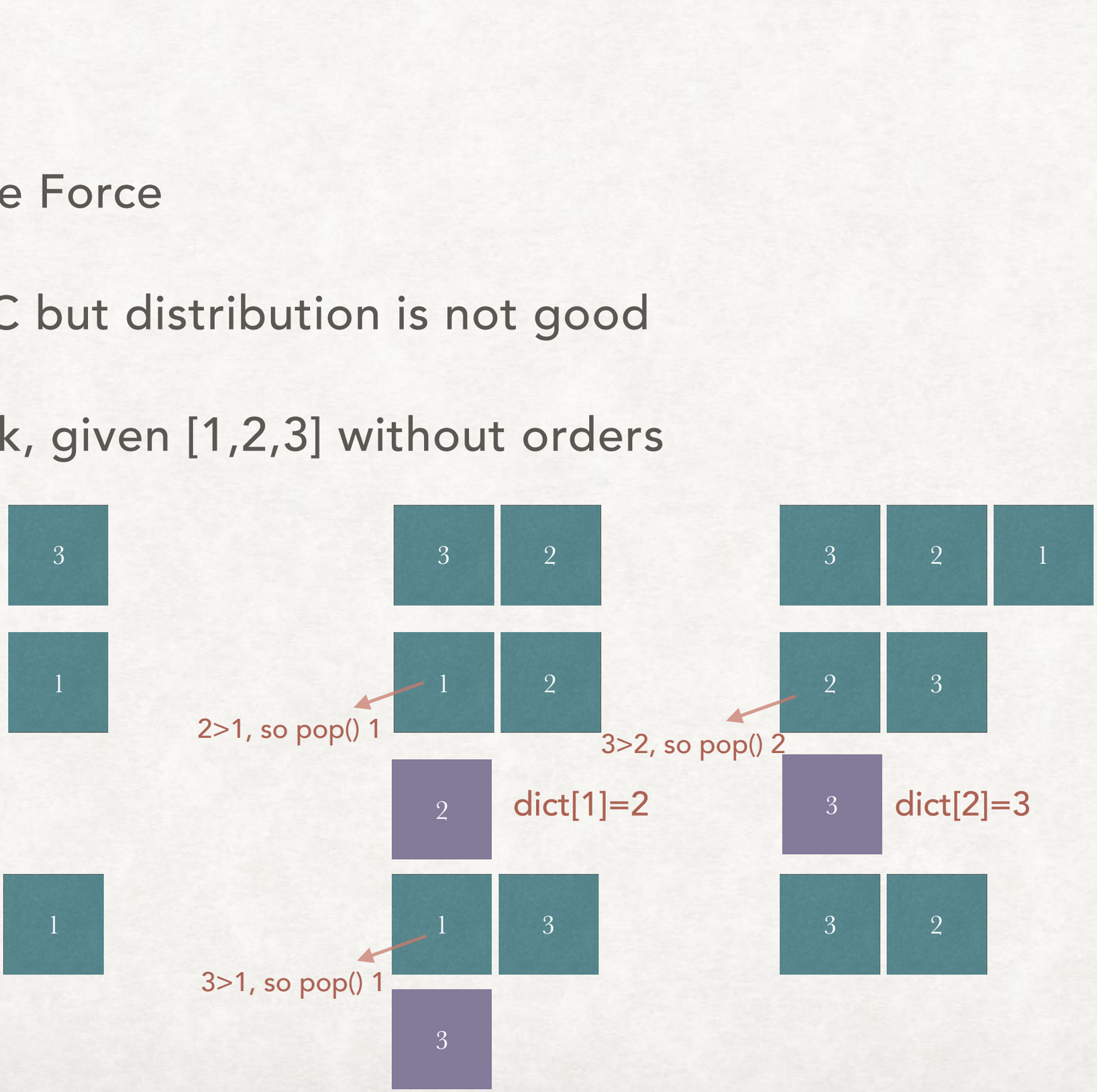
PROBLEM DESCRIPTION

- You are given two arrays (without duplicates) nums1 and nums2 where nums1’s elements are subset of nums2. Find all the next greater numbers for nums1's elements in the corresponding places of nums2.

- The Next Greater Number of a number x in nums1 is the first greater number to its right in nums2. If it does not exist, output -1 for this number.

IDEA

- Brute Force
  - AC but distribution is not good
- Stack, given [1,2,3] without orders
  - 1. 

 # [3,2,1]
  - 2. 

 # [1,2,3]
  - 3.

 # [1,3,2]

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/496_Next_Greater_Element_I

NO. 532

K-DIFF PAIRS IN AN ARRAY

PROBLEM DESCRIPTION

- Given an array of integers and an integer k, you need to find the number of unique k-diff pairs in the array. Here a k-diff pair is defined as an integer pair (i, j), where i and j are both numbers in the array and their absolute difference is k.

- Example 1:

  - Input: [3, 1, 4, 1, 5], k = 2, Output: 2

  - Explanation: There are two 2-diff pairs in the array, (1, 3) and (3, 5). Although we have two 1s in the input, we should only return the number of unique pairs.

IDEA

- If k > 0, then calculate whether n + k exists in the array
  - To boost up, use set() instead of list()
- if k == 0, then calculate how many pairs with the same value in the array?
  - Apply collections.Counter(iter).values()
- if k < 0, return 0

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/532_K-diff_Pairs_in_an_Array

NO. 268

MISSING NUMBER

PROBLEM DESCRIPTION

- Given an array containing n distinct numbers taken from 0, 1, 2, ..., n, find the one that is missing from the array.
- For example, given nums = [0, 1, 3] return 2.

IDEA

- Linear scan
  - AC but distribution sucks

```
for i in range(len(nums)):
        if nums[i] - i != 0: return i
```

- Sum up
  - ideal case = len(nums) * (len(nums) + 1) / 2
  - actual case = sum(nums)
  - missing number = ideal case - actual case

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/268_Missing_Number

NO. 504

BASE 7

PROBLEM DESCRIPTION

- Given an integer, return its base 7 string representation.
- Example 1: Input: 100, Output: “202"
- Example 2: Input: -7, Output: "-10"

IDEA

- Base conversion template
  - Convert to 10 base if necessary
  - Convert to new base

```
while num > 0:
    digit = num % new_base
    num /= new_base
    ans += digit
return ans[::-1]
```

- Notice if it is negative number

```
if negative: ans = '-'+ans
```

SOLUTION

- https://github.com/Brady31027/leetcode/tree/master/504_Base_7