# Assignment 4

1.Odd string difference:

def find\_odd\_string(words):

n = len(words[0])

for i in range(n - 1):

diff = ord(words[0][i + 1]) - ord(words[0][i])

for word in words:

temp\_diff = [ord(word[i + 1]) - ord(word[i]) for i in range(n - 1)]

if temp\_diff != diff:

return word

words1 = ["adc", "wzy", "abc"]

print(find\_odd\_string(words1))

words2 = ["aaa", "bob", "ccc", "ddd"]

print(find\_odd\_string(words2))

2.Words with in two digits of dictionary:

def words\_within\_two\_edits(queries, dictionary):

def is\_edit\_distance\_one(word1, word2):

if len(word1) != len(word2):

return False

count = 0

for i in range(len(word1)):

if word1[i] != word2[i]:

count += 1

if count > 1:

return False

return count == 1

def is\_edit\_distance\_two(word1, word2):

if len(word1) != len(word2):

return False

count = 0

for i in range(len(word1)):

if word1[i] != word2[i]:

count += 1

if count > 2:

return False

return count == 2

result = []

for query in queries:

for word in dictionary:

if query == word:

result.append(query)

break

elif is\_edit\_distance\_one(query, word):

result.append(query)

break

elif is\_edit\_distance\_two(query, word):

result.append(query)

break

return result

queries = ["word", "note", "ants", "wood"]

dictionary = ["wood", "joke", "moat"]

print(words\_within\_two\_edits(queries, dictionary))

3.Next greater element:

def nextGreaterElement(nums):

stack, result = [], [-1] \* len(nums)

for i in range(len(nums) \* 2):

while stack and (nums[i % len(nums)] > nums[stack[-1]]):

result[stack.pop()] = nums[i % len(nums)]

if i < len(nums):

stack.append(i)

return result

nums1 = [2, 4, 0, 9, 6]

output1 = nextGreaterElement(nums1)

print(output1)

4. Minimum Addition to Make Integer Beautiful:

def min\_addition\_to\_make\_beautiful(n, target):

digit\_sum = sum(int(digit) for digit in str(n))

if digit\_sum <= target:

return 0

else:

return (target - digit\_sum % (target + 1))

n = 16

target = 6

output = min\_addition\_to\_make\_beautiful(n, target)

print(output)

5. Sort Array by Moving Items to Empty Space:

def min\_operations\_to\_sort(nums):

n = len(nums)

count = 0

for i in range(n):

if nums[i] != 0 and nums[i] != i:

nums[nums[i]], nums[i] = nums[i], nums[nums[i]]

count += 1

return count

nums = [4, 2, 0, 3, 1]

print(min\_operations\_to\_sort(nums))