**Problem #1636: Sort Array by Increasing Frequency**

<https://leetcode.com/problems/sort-array-by-increasing-frequency/description/>

**My Solution:**

1. From python library collections import Counter.
2. Create freqDict to be a dictionary with key as the element in nums and value as the frequency of the element.
3. Initialize a dictionary called mydict as an empty dictionary.

This dictionary will have the key to the frequency from freqDict ie. value, and for the value of mydict will be a list of elements of num having the frequency given by the key in mydict.

1. Iterate through the key value pairs of freqDict items. If the value val is not in mydict keys, then make val the key in mydict and the value of mydict will be key as a list.

If value is already in mydict as the key, then append the key as the value.

1. Initialize res to be an empty list to store the result.
2. Iterate through the sorted keys in mydict (sorted in ascending order). Get the corresponding value list for the key called valList. Sort this valLis in descending order.

For each val in valList, add a list of val repeated key times to res.

1. Return res.

from collections import Counter

class Solution:

def frequencySort(self, nums: List[int]) -> List[int]:

freqDict = Counter(nums)

print("freqDict = ", freqDict)

mydict = {}

for key, val in freqDict.items():

if val not in mydict.keys():

mydict[val] = [key]

else:

mydict[val].append(key)

res = []

for key in sorted(mydict.keys()):

valList = sorted(mydict[key], reverse = True)

for val in valList:

res += [val] \* key

return res