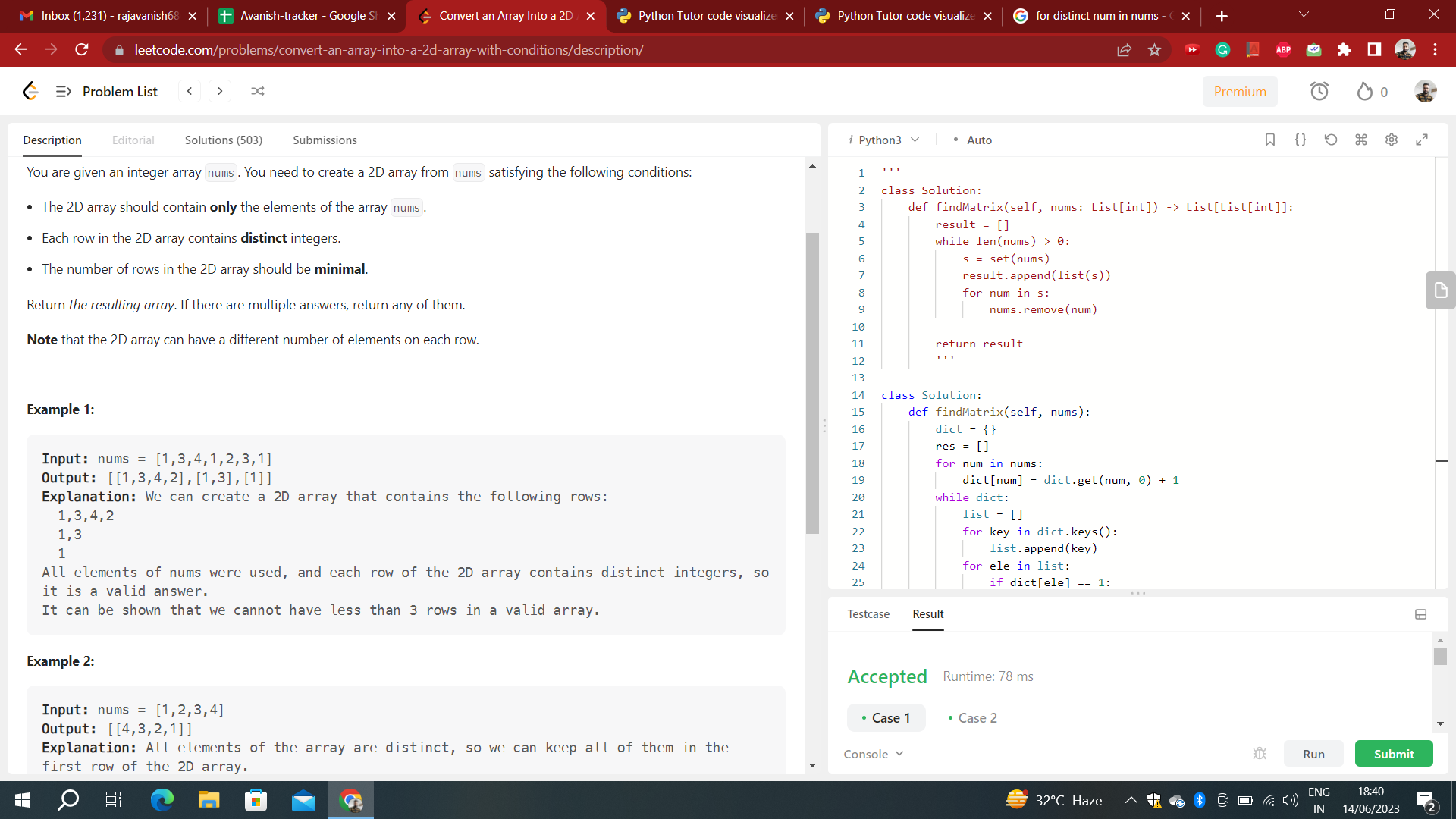
**Convert an Array Into a 2D Array With Conditions**

<https://leetcode.com/problems/convert-an-array-into-a-2d-array-with-conditions/>



**Time complexity: O()**

'''

class Solution:

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

        result = []

        while len(nums) > 0:

            s = set(nums)

            result.append(list(s))

            for num in s:

                nums.remove(num)

        return result

        '''

**Time complexity: O(n)**

class Solution:

    def findMatrix(self, nums):

        dict = {}

        res = []

        for num in nums:

            dict[num] = dict.get(num, 0) + 1

        while dict:

            list = []

            for key in dict.keys():

                list.append(key)

            for ele in list:

                if dict[ele] == 1:

                    dict.pop(ele)

                else:

                    dict[ele] -= 1

            res.append(list)

        return res

'''

from collections import Counter

class Solution:

    def findMatrix(self, nums):

        dict = Counter(nums)

        ans = []

        for key in dict:

            for val in range(dict[key]):

                print(len(ans))

                if len(ans) <= val:

                    ans.append([])

                ans[val].append(key)

        return ans

        '''

'''

from collections import Counter

class Solution:

    def findMatrix(self, nums):

        result = []

        dict = Counter(nums)

        for i in range(max(dict.values())):

            list = []

            for key, value in dict.items():

                if value != 0:

                    list.append(key)

                    dict[key] -= 1

            result.append(list)

        return result

        '''