Question: 4

Given an array nums of n integers, return an array of all the unique quadruplets [nums[a], nums[b], nums[c], nums[d]] such that:

a) 0 <= a, b, c, d < n

b) a, b, c, and d are distinct.

c) nums[a] + nums[b] + nums[c] + nums[d] == target

Ans:

class Solution:

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

n = len(nums)

ans = []

if n < 4:

return ans

nums.sort()

for i in range(n - 3):

if i and nums[i] == nums[i - 1]:

continue

for j in range(i + 1, n - 2):

if j > i + 1 and nums[j] == nums[j - 1]:

continue

k, l = j + 1, n - 1

while k < l:

x = nums[i] + nums[j] + nums[k] + nums[l]

if x < target:

k += 1

elif x > target:

l -= 1

else:

ans.append([nums[i], nums[j], nums[k], nums[l]])

k, l = k + 1, l - 1

while k < l and nums[k] == nums[k - 1]:

k += 1

while k < l and nums[l] == nums[l + 1]:

l -= 1

return ans