Brute Force

class Solution {

public int threeSumSmaller(int[] nums, int target) {

}

}

//

-2 + 0 + 1 = -1 <2

-2 + 0 + 3 = 1 < 2

0 + 1+ 3 = 4 !< 2

[-2,0,1,3],

i goes from -2 to 3

j goes from 0 to 3

k goes to 1 to 3

sum = -2 + 0 + 1

sum = -2 + 0 + 3

- -2 + 1 + 3

O(n ^ 3)

int result =0

i from 0 to n -1

j to i+1 to n-1

k to j+1 to n-1

sum = a[i] + a[j] + a[k]

if( sum < target)

result ++;

two sum problem - sort

-2,0,1,3],

sum of two less > greater,

0 3

ith j

1,2,3,5 - sorted

i j

0,1,2,3

1,2,3,5 (largest) -

1 + 5 < 7

1 + 2 < 7

1 + 3 < 7

Count of element for which sum would less that given target keeping left constnt = j(right) - i(left) = 3-0 =3

[1, 2, 3, 5, 8]

left

2,3,5 (right)

2,5 <= 7

3-1 = 2

j-i

1+ 5 = 6. < 7 one pair

1,2

1,3

1,5

no of combinations is equal to 3

// -2

//

//

[1, 2, 3, 5, 8] looking for 7

i goes 1 to 3

i is taken as 1

selection

[2 3, 5, 8] looking for 6

2+ 8 = 10

2+ 5 = 7

2+3 = 5 < 6

sum += 2 - 1; = 1

public int threeSumSmaller(int[] nums, int target) {

Arrays.sort(nums);

int sum = 0;

// select the first element

for (int i = 0; i < nums.length - 2; i++) {

sum += twoSumSmaller(nums, i + 1, target - nums[i]);

}

return sum;

}

// remaining two elements

private int twoSumSmaller(int[] nums, int startIndex, int target) {

int sum = 0;

int left = startIndex;

int right = nums.length - 1;

while (left < right) {

if (nums[left] + nums[right] < target) {

sum += right - left;

left++;

} else {

right--;

}

}

return sum;

}

}

—---

Optimize solution

public int threeSumSmaller(int[] nums, int target) {

Arrays.sort(nums);

int sum = 0;

for (int i = 0; i < nums.length - 2; i++) {

sum += twoSumSmaller(nums, i + 1, target - nums[i]);

}

return sum;

}

private int twoSumSmaller(int[] nums, int startIndex, int target) {

int sum = 0;

int left = startIndex;

int right = nums.length - 1;

while (left < right) {

if (nums[left] + nums[right] < target) {

sum += right - left;

left++;

} else {

right--;

}

}

return sum;

}

—--------------------