- Input: list<int>
- Sum the absolute difference w/ all other elements in list

Ex) L = [ 1 3 5 ]

Output[0] = |1-1| + |1-3| + |1-5| = 6 (0, 2, 4)

Output[1] = |3-1| + |3-3| + |3-5| = 4 (2, 0, 2)

Output[2] = |5-1| + |5-3| + |5-5| = 6 (4, 2, 0)

Naive:

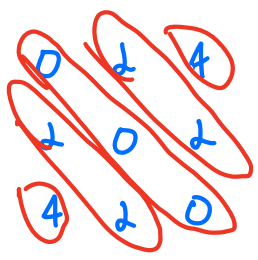
- Iterate over every element in nums.
- Foreach element, enumerate over each value, summing the result

```
result: [int]
foreach num in nums:
    sum = 0
    for each value in nums
        sum += |value - num|
    result.push_back(sum)
```

Improvement:

- Create output array of size n called result, 0 filled

If the above summation were in a matrix




* seems to be a pattern on the diagonal ↘

input = [ 1 3 7 16 24 25 ]

Output[0] = |1-1| + |1-3| + |1-7| + |1-16| + |1-24| + |1-25|
0 2 6 15 23 24

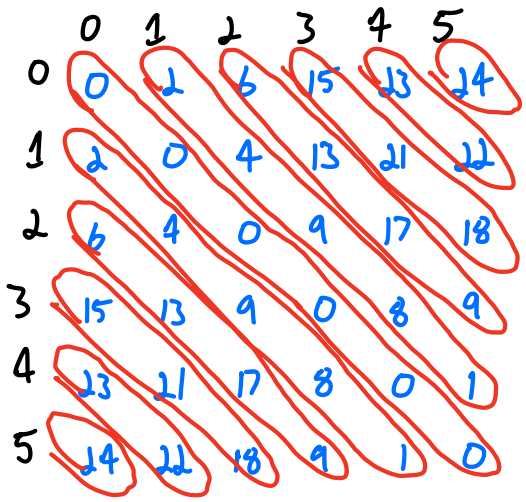
Output[1] = |3-1| + |3-3| + |3-7| + |3-16| + |3-24| + |3-25|
2 0 4 13 21 22

Output[2] = |7-1| + |7-3| + |7-7| + |7-16| + |7-24| + |7-25|
6 4 0 9 17 18

Output[3] = |16-1| + |16-3| + |16-7| + |16-16| + |16-24| + |16-25|
15 13 9 0 8 9

Output[4] = |24-1| + |24-3| + |24-7| + |24-16| + |24-24| + |24-25|
23 21 17 8 0 1

Output[5] = |25-1| + |25-3| + |25-7| + |25-16| + |25-24| + |25-25|
24 22 18 9 1 0




* Conclusion:

$|nums[i] - nums[j]| == |nums[j] - nums[i]|$

is True

We could pre-calculate a value at nums[j]

```
map[int,int]
result[int]
for i = 0 → n-1
    map[i+1] += |nums[i] - nums[i+1]
    result = map[i+1]
    for j = i+2 → n
        result += |nums[i] - nums[j]
    v.push_back(result)
return v
```

Close!

Algorithm correct, ordering of code wrong.

Problem: Complexity O(n²) !!

nums[int] = [2 3 5]

- Sum the array nums  sum(nums) = 10
- Iterate over all num in nums.
- Track the current sum, for each num,
  subtract total sum from the current.
  This is the r-sum ( r-sum = total - current)
- Find the l-sum (l-sum = current ; depends on iteration)

```
nums[int] = [ 2  3  5]
lsum = 0
rsum = sum(nums) = 10
result[int]
 for i = 0 → n
      ld = nums[i] * (len(nums)-i) + lsum
      rd = nums[i] * i + rsum
      result[i] = rd - ld
      rd -= nums[i]
      ld += nums[i]
→ result.
```

1.]

i = 0

nums[i] = 2

lsum = 0

rsum = 10

ld = 2 * (3 - 0) + 0 = 6

rd = 2 * 0 + 10 = 10

res = 10 - 6 = 4

2.]

i = 1

nums[i] = 3

lsum = 2

rsum = 8

ld = 3 * (3 - 1) + 2 = 8

rd = 3 * 1 + 8 = 11

res = 11 - 8 = 3

3.]

i = 2

nums[i] = 5

lsum = 5

rsum = 5

ld = 5 * (3 - 2) + 5 = 10

rd = 5 * 2 + 5 = 15

res = 15 - 10 = 5