

{% note info %} **摘要** Title: 1945. 奶牛棒球 Tag: 二分、双指针 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by: NEFU AB-IN

[Link](#)

[@TOC](#)

1945. 奶牛棒球

- 题意

农夫约翰的 N 头奶牛排成一排，每头奶牛都位于数轴中的不同位置上。它们正在练习投掷棒球。农夫约翰观看时，观察到一组三头牛 (X,Y,Z) 完成了两次成功的投掷。牛 X 把球扔给她右边的牛 Y ，然后牛 Y 把球扔给她右边的牛 Z 。约翰指出，第二次投掷的距离不少于第一次投掷的距离，也不超过第一次投掷的距离的两倍。请计算共有多少组牛 (X,Y,Z) 可能是约翰所看到的。

- 思路

- **二分** 先枚举其中两个点，然后二分第三个点， N 为 $1e6$ ， $O(n^2 \log n)$ 是可以过的
- **双指针** 找端点 l 和 r ， l 求的是 **大于等于** $2y-x$ 的最小值， r 求的是 **大于** $3y-2x$ 的最小值，每次枚举这两个端点即可

- 代码

有两版二分

- **手写二分**

- 找第一个大于等于 x 的下标
- 找最后一个小于等于 x 的下标

这两个二分属于是非常经典的例子了 所以范围是**两端闭合**的，故是 $(k2 - k1 + 1)$

```
'''
Author: NEFU AB-IN
Date: 2022-01-16 12:07:46
FilePath: \ACM\Acwing\1945.py
LastEditTime: 2022-01-16 22:34:58
'''

from bisect import bisect_left, bisect_right

n = int(input())
lst = []

def erfen1(j, x):
```

```

l = j
r = n - 1
while l < r:
    mid = l + r >> 1
    if lst[mid] >= x: #二分的位置在mid左边或mid的位置, 由于可能有多个,
    所以r = mid把所有符合的包含进去
        r = mid
    else:
        l = mid + 1
return r

def erfen2(j, x):
    l = j
    r = n - 1
    while l < r:
        mid = l + r + 1 >> 1 #因l = mid, 需要上取整
        if lst[mid] <= x:
            l = mid
        else:
            r = mid - 1
    return r

if __name__ == '__main__':
    for i in range(n):
        x = int(input())
        lst.append(x)
    lst.sort()
    res = 0
    for i in range(n - 2):
        for j in range(i + 1, n - 1):
            x = lst[j] - lst[i]
            # k1 = bisect_left(lst, lst[j] + x)
            # k2 = bisect_right(lst, lst[j] + 2 * x)
            # res += (k2 - k1)
            k1 = erfen1(j + 1, lst[j] + x)
            k2 = erfen2(j + 1, lst[j] + 2 * x)
            # 由于可能出现端点就不满足的情况, 所以需要判断
            if lst[k1] >= lst[j] + x and lst[k2] <= lst[j] + 2 * x:
                res += (k2 - k1 + 1)
    print(res)

```

调用函数

- 找第一个大于等于x的下标
- 找第一个大于x的下标, 其实就相当于上面求的, 只不过这里边界取不到

所以范围是左端闭合, 右端不闭合的, 故是 (k2 - k1)

```

'''
Author: NEFU AB-IN
Date: 2022-01-16 12:07:46
FilePath: \ACM\Acwing\1945.py
LastEditTime: 2022-01-16 22:34:58
'''

from bisect import bisect_left, bisect_right

n = int(input())
lst = []

def erfen1(j, x):
    l = j
    r = n - 1
    while l < r:
        mid = l + r >> 1
        if lst[mid] >= x: #二分的位置在mid左边或mid的位置, 由于可能有多个,
            所以r = mid把所有符合的包含进去
            r = mid
        else:
            l = mid + 1
    return r

def erfen2(j, x):
    l = j
    r = n - 1
    while l < r:
        mid = l + r + 1 >> 1 #因l = mid, 需要上取整
        if lst[mid] <= x:
            l = mid
        else:
            r = mid - 1
    return r

if __name__ == '__main__':
    for i in range(n):
        x = int(input())
        lst.append(x)
    lst.sort()
    res = 0
    for i in range(n - 2):
        for j in range(i + 1, n - 1):
            x = lst[j] - lst[i]
            k1 = bisect_left(lst, lst[j] + x)
            k2 = bisect_right(lst, lst[j] + 2 * x)
            res += (k2 - k1)
            # k1 = erfen1(j + 1, lst[j] + x)
            # k2 = erfen2(j + 1, lst[j] + 2 * x)
            # if lst[k1] >= lst[j] + x and lst[k2] <= lst[j] + 2 * x:

```

```
        #      res += (k2 - k1 + 1)
print(res)
```

◦ 双指针

```
'''
Author: NEFU AB-IN
Date: 2022-02-09 16:59:54
FilePath: \ACM\Acwing\1945.1.py
LastEditTime: 2022-02-09 17:06:01
'''

lst = []

if __name__ == '__main__':
    n = int(input())
    for i in range(n):
        x = int(input())
        lst.append(x)
    lst.sort()
    res = 0
    for i in range(n - 2):
        for j in range(i + 1, n - 1):
            x = lst[j] - lst[i]
            l = j + 1 #两个端点从j+1开始往后取
            r = j + 1
            while l < n and lst[l] < lst[j] + x:
                l += 1
            while r < n and lst[r] <= lst[j] + 2 * x: #小于等于就往后走
                r += 1
            res += r - l
    print(res)
```