

## 1502. Can Make Arithmetic Progression From Sequence

21 June 2022 09:21 AM

A sequence of numbers is called an **arithmetic progression** if the difference between any two consecutive elements is the same.

Given an array of numbers `arr`, return `true` if the array can be rearranged to form an **arithmetic progression**. Otherwise, return `false`.

### Example 1:

Input: `arr = [3,5,1]`

Output: `true`

Explanation: We can reorder the elements as `[1,3,5]` or `[5,3,1]` with differences 2 and -2 respectively, between each consecutive elements.

Arithmetic Progression :

$$a_n = a_1 + (n-1)d$$

$a_n$  = the  $n^{\text{th}}$  term in the sequence

$a_1$  = the 1<sup>st</sup> term in the sequence

$d$  = the common difference between terms.

$$a_1, a_1 + d, a_1 + 2d, \dots, a_1 + (n-1)d$$

1<sup>st</sup> Approach

`arr`  $\rightarrow$  sort

$$d = arr[1] - arr[0]$$

for ( $i=2 \rightarrow i+n$ )

if (`arr[i] - arr[i-1] != d`)

return false

return true

```
// 1st Approach
class Solution {
    public boolean canMakeArithmeticProgression(int[] arr) {
        Arrays.sort(arr);

        int d = arr[1] - arr[0];
        for(int i = 2; i < arr.length; i++){
            if(arr[i] - arr[i - 1] != d)
                return false;
        }
        return true;
    }
}
```

T.C  $\Rightarrow O(n \log n)$

2<sup>nd</sup> Approach

$$\begin{array}{ccccccc}
 a, & a+d, & a+2d, & \dots & a+(n-1)d \\
 \downarrow & & & & \uparrow \\
 \text{mini} & & & & \text{max}
 \end{array}$$

$$\text{max} \leftarrow T_n = a + (n-1)d$$

$$\text{min} \leftarrow T_1 = a$$

$$T_n - T_1 = (n-1)d \Rightarrow d = \frac{T_n - T_1}{n-1}$$

1<sup>st</sup> term  $a = \text{min of array}$

for (  $i \rightarrow$  )  
Set

$$\begin{array}{c}
 a_1 \\
 a_2 \\
 a_3 \\
 \vdots \\
 a_n
 \end{array}$$
 Set

min

$$\begin{array}{c}
 a + (i-1)d \\
 \downarrow \\
 \frac{\text{max} - \text{min}}{n-1}
 \end{array}$$

if we get any one

if we get any one of the value in the set, that means we found.

1. Find the `max` and `min` of `arr` and compute the average difference;
2. Put all numbers into a `HashSet`;
3. Start from the `min`, add the average difference to make the next number in the arithmetic sequence, check one by one if it is in the `HashSet`; if any one not in, return `false`; otherwise, return `true`.

**Note:**

1. There are `n - 1` slots between `n` element of the array;
2. `diff = max - min` must be divisible by `n - 1` for `arr` to be an arithmetic sequence;
3. After sorting `arr`, the adjacent elements difference must be `diff / (n - 1)`, if it is an arithmetic sequence.

```
class Solution {
    public boolean canMakeArithmeticProgression(int[] arr) {
        Set<Integer> seen = new HashSet<>();
        int mi = Integer.MAX_VALUE, mx = Integer.MIN_VALUE, n = arr.length;
        for (int a : arr) {
            mi = Math.min(mi, a);
            mx = Math.max(mx, a);
            seen.add(a);
        }
        int diff = mx - mi;
        if (diff % (n - 1) != 0) {
            return false;
        }
        diff /= n - 1;
        while (--n > 0) {
            if (!seen.contains(mi)) {
                return false;
            }
            mi += diff;
        }
        return true;
    }
}
```

1. why do this?

```
if (diff % (n - 1) != 0) {
    return false;
}
```

2. how does it become average difference of each element?

```
diff /= n - 1;
```

**A1:**

If you have 2 elements, there is only 1 difference;

If you have 3 elements, there are 2 differences;

If you have 4 elements, there are 3 differences;

...

If you have `n` elements, there are `n - 1` differences;

1. if `n - 1` can not divide the total difference `diff`, then at least 1 difference is not same as others;
2. `n` elements correspond to `n - 1` differences.

$$T.C \Rightarrow O(n) \quad S.C \Rightarrow O(1)$$