

## 检查整数及其两倍数是否存在

给你一个整数数组 `arr`，请你检查是否存在两个整数 `N` 和 `M`，满足 `N` 是 `M` 的两倍（即， $N = 2 * M$ ）。

更正式地，检查是否存在两个下标 `i` 和 `j` 满足：

- `i != j`
- `0 <= i, j < arr.length`
- `arr[i] == 2 * arr[j]`

**示例 1：**

输入：`arr = [10,2,5,3]`

输出：`true`

解释：`N = 10` 是 `M = 5` 的两倍，即  $10 = 2 * 5$ 。

**示例 2：**

输入：`arr = [7,1,14,11]`

输出：`true`

解释：`N = 14` 是 `M = 7` 的两倍，即  $14 = 2 * 7$ 。

**示例 3：**

输入：`arr = [3,1,7,11]`

输出：`false`

解释：在该情况下不存在 `N` 和 `M` 满足  $N = 2 * M$ 。

```
class Solution {
public:
    bool checkIfExist(vector<int>& arr) {
        int len=arr.size();
        int i;
        for(i=0;i<len;i++)
        {
            if(abs(arr[i])%2==0)
            {
                int key=arr[i]/2;

                for(int j=0;j<len;j++)
```

```
        {
            if(key==arr[j]&&i!=j) //i!=j 避免出现 0 的情况
                return true;
        }
    /*
        //二分查找
        sort(arr.begin(),arr.end());
        int left=0;
        int right=len-1;
        while(left<=right)
        {
            int mid=left+(right-left)/2;
            if(arr[mid]>key)
            {
                right=mid-1;
            }
            else if(arr[mid]<key)
            {
                left=mid+1;
            }
            else
                return true;
        }
    */
    }
    return false;
}
};
```