

将数组分成和相等的三个部分

给你一个整数数组 **A**，只有可以将其划分为三个和相等的非空部分时才返回 **true**，否则返回 **false**。

形式上，如果可以找出索引 $i+1 < j$ 且满足 $A[0] + A[1] + \dots + A[i] = A[i+1] + A[i+2] + \dots + A[j-1] = A[j] + A[j+1] + \dots + A[A.length - 1]$ 就可以将数组三等分。

示例 1:

输入: `[0,2,1,-6,6,-7,9,1,2,0,1]`

输出: `true`

解释: $0 + 2 + 1 = -6 + 6 - 7 + 9 + 1 = 2 + 0 + 1$

示例 2:

输入: `[0,2,1,-6,6,7,9,-1,2,0,1]`

输出: `false`

示例 3:

输入: `[3,3,6,5,-2,2,5,1,-9,4]`

输出: `true`

解释: $3 + 3 = 6 = 5 - 2 + 2 + 5 + 1 - 9 + 4$

解题思路:

先求和，除以三，除不开就 **false**，能除开就记录出现出现和为 **1/3** 的次数，大于等于三就对，大于三是用来看 **0** 的情况。

54 / 55 个通过测试用例

状态: 解答错误
提交时间: 5 分钟前

输入: `[10,-10,10,-10,10,-10,10,-10]`

输出: `false`

预期: `true`

```
bool canThreePartsEqualSum(int* arr, int arrSize){
    int sum=0;
    int cout=0;
    int div;
    for(int i=0;i<arrSize;i++)
    {
        sum+=arr[i];//求和
    }
    if(sum%3!=0)
        return false;
    div=sum/3;
    sum=0;
    for(int j=0;j<arrSize;j++)
    {
        sum+=arr[j];
        if(sum==div)
        {
            sum=0;
            cout++;//统计分割次数
        }
    }
    if(cout>=3)
        return true;
    else
        return false;
}
```