```
/*
给定一个整数数组 A，只有我们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，如果我们可以找出索引 i+1 < j 且满足 (A[0] + A[1] + ... + A[i] == A[i+1] + A[i+2] + ... + A[j-1] == A[j] + A[j-1] + ... + A[A.length - 1]) 就可以将数组三等分。



示例 1：

输出：[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



提示：

    3 <= A.length <= 50000
    -10000 <= A[i] <= 10000

来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/partition-array-into-three-parts-with-equal-sum
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/
```

分析:

- 首先求整体和,若整体和都不能被3整除，直接返回$false$;
- 分别从头尾两边开始求和，若求和满足三分量就跳出，并保存边界索引位置.
- 检测索引位置合法性.

方法一:C++

```
class Solution
{
```

```
    public:
        bool canThreePartsEqualSum(vector<int>& A)
        {
            int     sum_all     =   0       ;
            int     i           =   0       ;
            int     target      =   0       ;
            int     left        =   0       ;
            int     right       =   0       ;
            int     left_sum    =   0       ;
            int     right_sum   =   0       ;

            for(i = 0 ; i < A.size();i++)
            {
                sum_all +=  A[i];
            }

            if(sum_all%3)
            {
                return false;
            }
            else
            {
                target = sum_all / 3 ;

                left    = 0;
                right   = A.size()-1;

                while(left < A.size() )
                {
                    left_sum += A[left];
                    if(left_sum == target)
                    {
                        break;
                    }
                    left++;
                }

                while(right >= 0)
                {
                    right_sum += A[right];
                    if(right_sum == target)
                    {
                        break;
                    }
                    right--;
                }
                return ((right - left) > 0);
            }
        }
};

/*
执行结果:
通过
显示详情
执行用时 :60 ms, 在所有 cpp 提交中击败了75.85% 的用户
内存消耗 :12.6 MB, 在所有 cpp 提交中击败了49.40%的用户
*/
```

AlimyBreak
2019.11.21