

三角形的最大周长

给定由一些正数（代表长度）组成的数组 a ，返回由其中三个长度组成的、面积不为零的三角形的最大周长。

如果不能形成任何面积不为零的三角形，返回 0。

示例 1：

输入：[2,1,2]

输出：5

示例 2：

输入：[1,2,1]

输出：0

示例 3：

输入：[3,2,3,4]

输出：10

示例 4：

输入：[3,6,2,3]

输出：8

解题思路：排序+贪心算法

- (1) 第一步，数组排序 从小到大
- (2) 第二步，逆序查找，我们可以选择枚举三角形的最长边 c ，而从贪心的角度考虑，我们一定是选「小于 c 的最大的两个数」作为边长 a 和 b ，此时最有可能满足 $a+b>c$ ，使得三条边能够组成一个三角形，且此时的三角形的周长是最大的。
- (3) 因此，我们先对整个数组排序，倒序枚举第 i 个数作为最长边，那么我们只要看其前两个数 $A[i-2]$ 和 $A[i-1]$ ，判断 $A[i-2]+A[i-1]>A[i]$ 是否大于 $A[i]$ 即可，如果能组成三角形我们就找到了最大周长的三角形，返回答案 $A[i-2]+A[i-1]+A[i]$ 即可。如果对于任何数作为最长边都不存在面积不为零的三角形，则返回答案 0

```
class Solution {
public:
    int largestPerimeter(vector<int>& A) {
        sort(A.begin(),A.end());
        for(int i=A.size()-1;i>=2;i--)
        {
            if(A[i-1]+A[i-2]>A[i])
            {
                return A[i-1]+A[i-2]+A[i];
            }
        }
        return 0;
    }
};
```