# 题目

给定由一些正数（代表长度）组成的数组nums，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：nums = [2,1,2]

输出：5

解释：你可以用三个边长组成一个三角形:1 2 2。

示例 2：

输入：nums = [1,2,1,10]

输出：0

解释：

你不能用边长 1,1,2 来组成三角形。

不能用边长 1,1,10 来构成三角形。

不能用边长 1、2 和 10 来构成三角形。

因为我们不能用任何三条边长来构成一个非零面积的三角形，所以我们返回 0。

提示：

3 <= nums.length <= 10^4

1 <= nums[i] <= 10^6

# 分析

## 方法一：排序+贪心

思路：

代码：

class Solution {

public:

int largestPerimeter(vector<int>& nums) {

sort(nums.begin(), nums.end(), greater<int>());

for (int i = 0; i < nums.size() - 2; ++i) {

if (nums[i] < nums[i + 1] + nums[i + 2]) {

return nums[i] + nums[i + 1] + nums[i + 2];

}

}

return 0;

}

};