

976. Largest Perimeter Triangle

Solution:-

```
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;  
    }  
};
```

The screenshot shows a code editor interface with a dark theme. On the left, the problem description for '976. Largest Perimeter Triangle' is visible, including the input/output examples and the explanation. The main area displays the C++ code solution, which sorts the array in descending order and checks for the largest valid triangle perimeter. The 'Test Result' section at the bottom shows 'Accepted' with a runtime of 0 ms for the test case [2,1,2].

976. Largest Perimeter Triangle Solved

Given an integer array `nums`, return the largest perimeter of a triangle with a non-zero area, formed from three of these lengths. If it is impossible to form any triangle of a non-zero area, return 0.

Example 1:
Input: `nums = [2,1,2]`
Output: 5
Explanation: You can form a triangle with three side lengths: 1, 2, and 2.

Example 2:
Input: `nums = [1,2,1,10]`
Output: 0
Explanation: You cannot use the side lengths 1, 1, and 2 to form a triangle. You cannot use the side lengths 1, 1, and 10 to form a triangle. You cannot use the side lengths 1, 2, and 10 to form a triangle. As we cannot use any three side lengths to form a triangle of non-zero area, we return 0.

```
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;  
    }  
};
```

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input
`nums = [2,1,2]`