

## [561 Array Partition \(link\)](#)

### Description

Given an integer array `nums` of  $2n$  integers, group these integers into  $n$  pairs  $(a_1, b_1)$ ,  $(a_2, b_2)$ , ...,  $(a_n, b_n)$  such that the sum of  $\min(a_i, b_i)$  for all  $i$  is **maximized**. Return *the maximized sum*.

#### Example 1:

**Input:** `nums = [1,4,3,2]`

**Output:** 4

**Explanation:** All possible pairings (ignoring the ordering of elements) are:

1.  $(1, 4), (2, 3) \rightarrow \min(1, 4) + \min(2, 3) = 1 + 2 = 3$
2.  $(1, 3), (2, 4) \rightarrow \min(1, 3) + \min(2, 4) = 1 + 2 = 3$
3.  $(1, 2), (3, 4) \rightarrow \min(1, 2) + \min(3, 4) = 1 + 3 = 4$

So the maximum possible sum is 4.

#### Example 2:

**Input:** `nums = [6,2,6,5,1,2]`

**Output:** 9

**Explanation:** The optimal pairing is  $(2, 1), (2, 5), (6, 6)$ .  $\min(2, 1) + \min(2, 5) + \min(6, 6) = 1 + 2 + 6 = 9$ .

#### Constraints:

- $1 \leq n \leq 10^4$
- `nums.length == 2 * n`
- $-10^4 \leq \text{nums}[i] \leq 10^4$

(scroll down for solution)

# Solution

Language: *cpp*

Status: Accepted

```
#include <vector>
#include <algorithm>

using namespace std;

class Solution {
public:
    int arrayPairSum(vector<int>& nums) {
        sort(nums.begin(), nums.end()); // Сортируем массив

        int sum = 0;
        int n = nums.size() / 2;

        for (int i = 0; i < n; ++i) {
            sum += nums[2*i]; // Берем только каждый второй элемент (минимальные из пар)
        }

        return sum;
    }
};
```