

## 数组拆分 I

给定长度为  $2n$  的整数数组 `nums`，你的任务是把这些数分成  $n$  对，例如  $(a_1, b_1)$ ， $(a_2, b_2)$ ， $\dots$ ， $(a_n, b_n)$ ，使得从 1 到  $n$  的  $\min(a_i, b_i)$  总和最大。

返回该 **最大总和**。

**示例 1:**

输入: `nums = [1,4,3,2]`

输出: 4

解释: 所有可能的分法（忽略元素顺序）为:

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

所以最大总和为 4

**示例 2:**

输入: `nums = [6,2,6,5,1,2]`

输出: 9

解释: 最优的分法为  $(2, 1), (2, 5), (6, 6)$ .  $\min(2, 1) + \min(2, 5) + \min(6, 6) = 1 + 2 + 6 = 9$

```
class Solution {
public:
    int arrayPairSum(vector<int>& nums) {
        sort(nums.begin(),nums.end());
        int sum=0;
        for(int i=0;i<nums.size();i+=2)
        {
            sum+=nums[i];
        }
        return sum;
    }
};
```