

给定长度为 $2n$ 的数组, 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。

示例 1:

输入: `[1, 4, 3, 2]`

输出: `4`

解释: n 等于 2, 最大总和为 $4 = \min(1, 2) + \min(3, 4)$.

解题思路, 其实就是把数组排序, 然后从0位开始相加, 自增数为2, 像

$a[0], a[1], a[2], a[3]$, 几个排序相加, 能组合的情况为 $a[0], a[1]$ 和 $a[0], a[2]$ 因为右边和左边调换顺序的话, 结果就必为 $a[0], a[1]$ 的组合, 比另外一组小, 将这个思路推及到整个数组, 得到自增2的相加。

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