## 题目描述：

Given an array of **2n** integers, your task is to group these integers into **n** pairs of integer, say ($a_1$, $b_1$), ($a_2$, $b_2$), ..., ($a_n$, $b_n$) which makes sum of min($a_i$, $b_i$) for all i from 1 to n as large as possible.

**Example 1:**

```
Input: [1,4,3,2]

Output: 4
Explanation: n is 2, and the maximum sum of pairs is 4 = min(1, 2) + min(3, 4).
```

**Note:**

3. **n** is a positive integer, which is in the range of [1, 10000].
4. All the integers in the array will be in the range of [-10000, 10000].

## 自己的解题思路：

先对数组排序，然后从零开始，连续的2个分为一组，取第一个小的为求和元素，这样求出来的（min(a,b)+min(c,d)）必定是最大的。思想：让较大的和较大在一起，才能保留较大的，所以先排序后分组

此处卡顿的地方：快排 （排序还要多练习）

## 题解代码：

```
class Solution {
    public int arrayPairSum(int[] nums) {
        quickSort(0,nums.length-1,nums);
        int minSum = 0;
        for(int i = 0;i< nums.length;i++){
            if((i+1)%2 == 1) minSum += nums[i];
        }
        return minSum;
    }

    private void quickSort(int low,int high,int[] nums){
        if(low>=high) return;
        int target = nums[low];
        int index = partition(low,high,nums);
        nums[index] = target;
```

```
        quickSort(low,index-1,nums);
        quickSort(index+1,high,nums);
    }

    private int partition(int low,int high,int nums[]){
        int target = nums[low];
        while(low<high){
            while(nums[high] >= target && low<high) high--;
            nums[low] = nums[high];
            while(nums[low] <= target && low<high) low++;
            nums[high] = nums[low];
        }
        return low;
    }
}
```