5755. Minimize Maximum Pair Sum in Array

My Submissions (/contest/biweekly-contest-53/problems/minimize-maximum-pair-sum-in-array/submissions/)

Back to Contest (/contest/biweekly-contest-53/)

The **pair sum** of a pair `(a,b)` is equal to `a + b`. The **maximum pair sum** is the largest **pair sum** in a list of pairs.

- For example, if we have pairs `(1,5)`, `(2,3)`, and `(4,4)`, the **maximum pair sum** would be `max(1+5, 2+3, 4+4) = max(6, 5, 8) = 8`.

Given an array `nums` of **even** length `n`, pair up the elements of `nums` into `n / 2` pairs such that:

- Each element of `nums` is in **exactly one** pair, and
- The **maximum pair sum** is **minimized**.

Return *the minimized **maximum pair sum** after optimally pairing up the elements*.

User Accepted:	9
User Tried:	12
Total Accepted:	9
Total Submissions:	12
Difficulty:	Medium

Example 1:

```
Input: nums = [3,5,2,3]
Output: 7
Explanation: The elements can be paired up into pairs (3,3) and (5,2).
The maximum pair sum is max(3+3, 5+2) = max(6, 7) = 7.
```

Example 2:

```
Input: nums = [3,5,4,2,4,6]
Output: 8
Explanation: The elements can be paired up into pairs (3,5), (4,4), and (6,2).
The maximum pair sum is max(3+5, 4+4, 6+2) = max(8, 8, 8) = 8.
```

Constraints:

- `n == nums.length`
- `2 <= n <= 10^5`
- `n` is **even**.
- `1 <= nums[i] <= 10^5`

JavaScript

```
const mx = Math.max;
const stin = (a) => a.sort((x, y) => x - y);
const minPairSum = (a) => {
    stin(a);
    let res = 0;
    let n = a.length;
    for (let i = 0; i < n; i++) {
        let sum = a[i] + a[n - 1 - i];
        res = mx(res, sum);
    }
    return res;
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/499897050/) More Details (/submissions/detail/499897050/)

Share your acceptance!