5998. Maximum Split of Positive Even Integers

My Submissions (/contest/biweekly-contest-72/problems/maximum-split-of-positive-even-integers/submissions/)

Back to Contest (/contest/biweekly-contest-72/)

You are given an integer `finalSum`. Split it into a sum of a **maximum** number of **unique** positive even integers.

- For example, given `finalSum = 12`, the following splits are **valid** (unique positive even integers summing up to `finalSum`): `(2 + 10)`, `(2 + 4 + 6)`, and `(4 + 8)`. Among them, `(2 + 4 + 6)` contains the maximum number of integers. Note that `finalSum` cannot be split into `(2 + 2 + 4 + 4)` as all the numbers should be unique.

Return *a list of integers that represent a valid split containing a* ***maximum*** *number of integers*. If no valid split exists for `finalSum`, return *an* ***empty*** *list*. You may return the integers in **any** order.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:

```
Input: finalSum = 12
Output: [2,4,6]
Explanation: The following are some valid splits: (2 + 10), (2 + 4 + 6), and (4 + 8).
(2 + 4 + 6) has the maximum number of integers, which is 3. Thus, we return [2,4,6].
Note that [2,6,4], [6,2,4], etc. are also accepted.
```

Example 2:

```
Input: finalSum = 7
Output: []
Explanation: There are no valid splits for the given finalSum.
Thus, we return an empty array.
```

Example 3:

```
Input: finalSum = 28
Output: [6,8,2,12]
Explanation: The following are some valid splits: (2 + 26), (6 + 8 + 2 + 12), and (4 + 24).

(6 + 8 + 2 + 12) has the maximum number of integers, which is 4. Thus, we return [6,8,2,12].
Note that [10,2,4,12], [6,2,4,16], etc. are also accepted.
```

Constraints:

- $1 <= finalSum <= 10^{10}$

JavaScript

```
const maximumEvenSplit = (x) => {
    if (x & 1) return [];
    if (x == 2) return [2];
    let se = new Set(), sum;
    for (let i = 2; i < x; i += 2) {
        sum = sumOfRange(2, i);
        se.add(i);
        if (sum == x) {
            return [...se];
        } else if (sum > x) {
            se.delete(i);
            sum -= i;
            break;
        }
    }
    let rest = x - sum;
    for (const e of se) {
```

```
        let expect = e + rest;
        if (!se.has(expect)) {
            se.add(expect)
            se.delete(e);
            return [...se];
        }
    }
    return [];
};

const sumOfRange = (l, r) => {
    let cnt = (r - l) / 2 + 1;
    return (l + r) * cnt / 2;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/644590519/) | More Details > (/submissions/detail/644590519/)

Share your acceptance!