Question #1 (Very Hard)

Description:

Implement a JavaScript function named findUniqueIndexPairs that, given an array of integer numbers and a target sum targetSum, identifies all pairs of indices whose values sum up to the target sum. Each pair of indices should be unique and returned as an array of pairs, with each pair represented as a two-element array. The function must ensure that:

- The sum of the values at the two indices in each pair equals the target sum.
- Each index from the array is used at most once across all pairs.
- The pairs are unique; there are no duplicate pairs in the output.
- The order of pairs or indices in the output does not matter.

Constraints:

- The array may contain both positive and negative integers.
- The function should correctly handle cases with duplicate values in the input array.
- If no pairs match the criteria, the function should return an empty array.
- Indices in a pair should be listed in ascending order (i.e., [i, j] where i < j).

Note: It is possible to have different valid answers depending on the set of unique pairs that was chosen by your solution.

Starting Code

```
function findUniqueIndexPairs(numbers, targetSum){
     // Write your code here
}
```

Sample Input 1

```
const numbers = [2, 4, -2, 1, 0, 5, -4];
const targetSum = 3;
findUniqueIndexPairs(numbers, targetSum)
```

Sample Output 1

```
[
  [0, 3], // 2 + 1 = 6
  [2, 5], // -2 + 5 = 3
  // Note: No more pairs can be formed without reusing indices.
]
```

Sample Input 2

```
const numbers = [1, 2, 3, 4, 5];
const targetSum = 9;
findUniqueIndexPairs(numbers, targetSum)
```

Sample Output 2

```
[
  [3, 4] // 4 + 5 = 9
  // only one pair meets the criteria without reusing indices.
]
```