Given an array of integers and a target, return the indices of two numbers such that they add up to the target number.

Example Input: nums = [3,2,4], target = 6

Example Output: [1, 2] since, nums[1] + nums[2] = target

javascript

```javascript
function targetSum (arr, t) {
    const arrWithIdx = arr.map((val, idx) => ({val, idx}))

    arrWithIdx.sort((a, b) => a.val - b.val)
    let i = 0;
    let j = arr.length - 1

    while (i < j) {
        let sum = arrWithIdx[i].val + arrWithIdx[j].val
        if (sum < t) i++
        else if (sum > t) j--
        else return [arrWithIdx[i].idx, arrWithIdx[j].idx]
    }
    return -1
}

console.log(targetSum([1,4,1,2], 3))
```

Short way:

javascript

```javascript
function targetSum (arr, t) {
    let complementSet = {}
    let complement
    for (let [idx, val] of arr.entries()) {
        complement = t - val
        if (complementSet[complement]) {
            return [complementSet[complement], idx]
        }
        complementSet[val] = idx
    }
}

console.log(targetSum([1,4,1,2], 3))
```