

## Algorithms

recf15d1pqKhxYBcH

*Описание задания.*

*1 уровень сложности: 1) Проверьте, равны ли два массива или нет*

*Учитывая два массива, arr1 и arr2 одинаковой длины N, задача состоит в том, чтобы определить, равны ли данные массивы или нет.*

*Два массива называются равными, если:*

*оба они содержат один и тот же набор элементов,*

*расположение (или перестановки) элементов может/не может быть одинаковым.*

*Если есть повторения, то количество повторяющихся элементов также должно быть одинаковым, чтобы два массива были равны.*

*Ввод: arr1[] = {1, 2, 5, 4, 0}, arr2[] = {2, 4, 5, 0, 1}*

*Вывод: Да*

*Временная сложность -  $O(n)$*

```
function isEqualArrays (arr1, arr2) {  
    let arr1Sort = arr1.sort((a, b) => a - b)  
    let arr2Sort = arr2.sort((a, b) => a - b)  
  
    for (let i = 0; i < arr1.length; i++) {  
        if (arr1Sort[i] !== arr2Sort[i]) {  
            return false;  
        }  
    }  
    return true  
}  
  
console.log(isEqualArrays (arr1, arr2))
```

### Оптимальный вариант:

```
function isEqualArrays (arr1, arr2) {  
    const frequencies1 = new Map()  
    for (let i = 0; i < arr1.length; i++) {  
        let key = arr1[i]  
        const counter = frequencies1.has(key) ? frequencies1.get(key) + 1 : 1  
        frequencies1.set(key, counter)  
    }  
    const frequencies2 = new Map()  
    for (let i = 0; i < arr2.length; i++) {  
        let key = arr2[i]  
        const counter = frequencies2.has(key) ? frequencies2.get(key) + 1 : 1  
        frequencies2.set(key, counter)  
    }  
    for (const [key, counter1] of frequencies1) {  
        const counter2 = frequencies2.get (key)  
        if (counter1 !==counter2){  
            return false  
        }  
    }  
}
```

2)

*Сгруппируйте слова с одинаковым набором символов*

*Дан список слов со строчными буквами. Реализуйте функцию поиска всех слов с одинаковым уникальным набором символов.*

*Input: String words[] = {"student", "students", "dog", "god", "cat", "act", "flow", "wolf"};*

*Output :*

*student, students,*

*cat, act,*

*dog, god,*

*flow, wolf,*

*Идея состоит в том, чтобы использовать хеширование. Генерируем ключ для всех слов. Ключ содержит все уникальные символы (размер ключа не более 26 для строчных букв). Мы храним индексы слов как значения для ключа. После того, как мы заполнили все ключи и значения в хеш-таблице, мы можем распечатать результат, пройдя по таблице.*

```
function GroupSameWords(words) {  
    const map = new Map()  
    for (let i = 0; i < words.length; i++){  
        const key = [ ... new Set (words[i].split ( ' '))].sort ().join ( ' ' )  
        const wordsByKey = map.has (key) ? map.get (key) : []  
        wordsByKey.push (words[i])  
        map.set (key, wordsByKey)  
    }  
    return [...map.values()]  
}  
  
console.log (GroupSameWords(words) )
```