|  |  |  |
| --- | --- | --- |
| **Встречаемость 1-10** | **Задача** | **Файл** |
| **10** | RLE. Дана строка (возможно, пустая), состоящая из букв A-Z: AAAABBBCCXYZDDDDEEEFFFAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBB  Нужно написать функцию RLE, которая на выходе даст строку вида: A4B3C2XYZD4E3F3A6B28 | compress-repeat-chars.js |
| **10** | Вернуть длину наибольшей непрерывной цепочки повторяющегося символа. Может быть задано любого символа или “1”. | longest-repetition.js |
| **9** | Вернуть K наиболее частых слов. | most-frequent-words.js |
| **9** | Построить по шаблону объект с вложенными полями-объектами. | nested-objects.js |
| **8** | Реализовать глубокое клонирование объекта. | deep-clone.js |
| **8** | Нечёткий поиск.  Подстрока str должна быть в dataStr в нужном порядке, но не обязательно слитно. | fuzzy-search.-ordered.js |
| **8** | Дан список интов, повторяющихся элементов в списке нет. Нужно преобразовать это множество в строку, сворачивая соседние по числовому ряду числа в диапазоны. Примеры:  [1,4,5,2,3,9,8,11,0] => "0-5,8-9,11"  [1,4,3,2] => "1-4"  [1,4] => "1,4" | make-intervals.js |
| **8** | Объединить два отсортированных по возрастанию массивов так, чтобы получился массив, упорядоченный по возрастанию. | merge-two-arrays.js |
| **7** | Реализовать удаление дубликатов в отсортированном массиве чисел. | dedup-numbers.js |
| **7** | Реализовать генерацию первых N чисел Фибоначчи. | fibo.js |
| **7** | Sample Input ["eat", "tea", "tan", "ate", "nat", "bat"]  Sample Output [ ["ate", "eat", "tea"], ["nat", "tan"], ["bat"] ]  Т.е. сгруппировать слова по "общим буквам". | group-words-by-chars.js |
| **7** | Вернуть длину наибольшей непрерывной цепочки повторяющегося символа. | longest-repetition.js |
| **7** | Слияние отрезков: (дубликат)  Вход: [1, 3] [100, 200] [2, 4]  Выход: [1, 4] [100, 200] | merge-overlapped-intervals.js |
| **7** | В массиве целых чисел, где числа идут подряд, возрастая на '1', отсутствует одно число. Найти его. | one-absent-number.js |
| **7** | Проверить, является ли слово палиндромом. | palindrome-words.js |
| **7** | Вернуть число символов S, которые встречаются в J. | warmup.js |
| **6** | Нечёткий поиск.  Подстрока str должна быть в dataStr слитно, но в произвольном порядке символов. | fuzzy-search.js |
| **6** | Проверить, является ли число палиндромом. | palindrome-number.js |
| **6** | Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.  An input string is valid if:  Open brackets must be closed by the same type of brackets.  Open brackets must be closed in the correct order. | valid-brackets.js |
| **5** | Реализовать парсинг целых чисел. На вход могут приходить числа со знаком и с примесью нечисловых знаков. Нечисловые хвосты надо отсекать, а префиксы обрабатывать правильно.  Возвращать ошибку, а пробелы пропускать. Знак должен быть указан слитно с числом. | atoi.js |
| **5** | Даны два массива: [1, 2, 3, 2, 0] и [5, 1, 2, 7, 3, 2]  Надо вернуть [1, 2, 2, 3] (порядок неважен)  Фактически нам нужно вернуть пересечение множеств, но с повторением элементов. | intersect-arrays.js |
| **5** | Проверить, являются переданные строки S и T анаграммами. | valid-anagram.js |
| **4** | Даны даты заезда и отъезда каждого гостя.  Для каждого гостя дата заезда строго раньше даты отъезда (то есть каждый гость останавливается хотя бы на одну ночь).  В пределах одного дня считается, что сначала старые гости выезжают, а затем въезжают новые.  Найти максимальное число постояльцев, которые одновременно проживали в гостинице (считаем, что измерение количества постояльцев происходит в конце дня).  sample = [ (1, 2), (1, 3), (2, 4), (2, 3), ] | max-hotel-visitors.js |
| **4** | Вернуть длину наибольшей непрерывной цепочки '1', которую можно получить удалением только одного '0'. | max-unit-seq-removing-zero.js |
| **4** | Объединить N отсортированных по возрастанию смассивов так, чтобы получился массив, упорядоченный по возрастанию. | merge-n-arrays.js |
| **4** | Обменять два числа местами не используя третью переменную. | swap-numbers.js |
| **3** | Дан список интов и число-цель. Нужно найти такой range, чтобы сумма его элементов давала число-цель.  elements = [1, -3, 4, 5]  target = 9 | find-target-interval-summ.js |
| **2** | Дано N - число скобок. Нужно сгенерировать все комбинации правильных скобочных структур длины 2\*N. | all-valid-brackets.js |
| **1** | Даны два неотрицательных целых числа. Надо их сложить в виде строк (как в столбик). | add-numbers.js |
| **1** | Надо проверить является ли число N простым, а потом оптимизировать алгоритм. | prime-numbers.js |
|  |  |  |