



(つ・_・)つ ♥ Поступашки - ШАД, Стажировки и Магистратура ♥

t.me/postypashki-old

Введение-содержание

Помимо благотворительной деятельности, Поступашки также проводят курсы и индивидуальные занятия по подготовке к ШАД, к олимпиадам, к собеседованиям, подготовке абитуриентов к ВУЗовской программе, подготовке к экзаменам, контрольным и прочим студентческим работам по основным математическим дисциплинам: анализ, линейная алгебра, теория вероятностей, теория групп и тд. А также по алгоритмам и структурам данных, ML&DL.

[Подробнее об индивидуальных занятиях](#)

[Подробнее о курсах](#)

Начинающий архивариус

Люба, начинающий разработчик, получила первый большой проект: нужно цифровизировать архив документов и сделать удобный поиск. С первой частью она уже справилась: все документы разбиты на секции и хранятся в специальном образом организованном хранилище json-ов. Теперь ей предстоит реализовать функцию, которая будет осуществлять поиск по этим документам.

Условия:

Каждый документ имеет:

- id,
- список версий этого документа,
- список документов, от которых зависит.

Каждая версия имеет:

- дату начала периода действия в UTC,
- дату окончания периода действия в UTC,
- текст документа.

https://t.me/postypashki_oldhttps://t.me/postypashki_old

На вход функции будет передаваться 3 аргумента:

- дата начала периода в таймзоне пользователя в формате YYYY-MM-DD
- дата окончания периода в таймзоне пользователя в формате YYYY-MM-DD
- массив документов.

Функция должна вернуть тексты всех документов, которые были активными в этот период, в порядке появления в исходном массиве. Версия считается активной, только если у всех документов из массива `deps` документа есть актуальные на данный период версии. Если в течение запрошенного периода версии менялись - нужно вернуть последнюю актуальную. Если какой-то документ закончил своё действие в этот период - вернуть последнюю актуальную версию. Если действие версии не ограничено справа или слева, там будет `null`.

Пример использования:

Вызов `search('2025-01-01', '2025-02-01', {...})` должен вернуть массив строк типа `['string13', 'string24']`:

Входные данные:

```
'2025-02-01'
'2025-02-28'
[
  {
    id: '1',
    deps: ['2'],
    versions: [
      {
        from: '2025-01-01T00:00:00Z',
        to: '2025-02-01T00:00:00Z',
        text: 'String 11',
      },
      {
        from: '2025-02-01T00:00:00Z',
        to: '2025-03-01T00:00:00Z',
        text: 'String 12',
      },
      {
        from: '2025-03-01T00:00:00Z',
        to: '2025-04-01T00:00:00Z',
        text: 'String 13',
      },
    ],
  },
  {
    id: '2',
    deps: [],
    versions: [
      {
        from: '2025-01-01T00:00:00Z',
        to: '2025-02-01T00:00:00Z',
        text: 'String 21',
      },
      {
        from: '2025-02-01T00:00:00Z',
        to: '2025-03-01T00:00:00Z',
        text: 'String 22',
      },
      {
        from: '2025-03-01T00:00:00Z',

```

```
    from: '2025-03-01T00:00:00Z',  
    to: '2025-04-01T00:00:00Z',  
    text: 'string 23',  
  },  
],
```

Ответ: ['String 12', 'String 22']

Ваше решение должно быть экспортировано в виде модуля: `module.exports = { search };`

Энергетические уровни

Вы работаете ученым на атомной электростанции. Известно, что количество энергии, вырабатываемое реактором в каждый момент времени, равно какому-либо значению из конечного множества возможных значений. У вас для исследования есть столбчатая диаграмма с результатом замеров уровня энергии реактора в течение дня в различные моменты времени. Тщательно изучив данные, вы обнаруживаете аномалию: между двумя соседними по значению уровнями энергии слишком большой разрыв. Необходимо разобраться в чем может быть дело.

Формат ввода

Входными данными является ссылка на DOM-узел, непосредственными дочерними элементами которого являются div-элементы, представляющие столбцы на диаграмме. Значение уровня энергии хранится в атрибуте `data-value` каждого такого элемента.

Примечания

https://t.me/postypashki_oldhttps://t.me/postypashki_oldhttps://t.me/postypashki_old

Пример вёрстки страницы можно скачать из приложенного [файла с песочницей](#), в нем вы так же сможете отладить свое решение.

Ваше решение должно быть оформлено в виде функции, которая принимает единственный аргумент — ссылку на DOM-элемент, являющийся контейнером для элементов-столбцов. Функция должна возвращать число — максимальный разрыв между смежными с точки зрения значений уровнями энергии. Столбцы на диаграмме отображают уровень энергии в различные моменты времени, поэтому они не упорядочены по значению.

Шаблон решения

```
function solution(container) {  
    // ваше решение  
}
```

Решение должно быть написано на JavaScript и предназначено для выполнения в контексте браузера.

Пиксельный отпечаток

Разработайте алгоритм, который принимает на вход информацию о разработчике в виде объекта данных и визуализирует её в форме пиксельной графики, где каждый параметр представлен определённым цветом.

Информация, содержащаяся в объекте `window.content`, который состоит из строк, должна быть закодирована в отдельные цветные блоки на изображении.

Данные преобразуются в массив байтов: каждому символу строки присваивается его ASCII-код (число от 0 до 255), далее каждое число преобразуется в двоичную систему (состоящую из символов 0 и 1) и отображается на рисунке в виде последовательности из восьми квадратов (где 0 - белый квадрат, а 1 - цветной).

Пример преобразования:

- Исходная строка: `abc`
- ASCII-коды: `a = 97, b = 98, c = 99`
- Двоичная форма: `97 = 1100001, 98 = 1100010, 99 = 1100011`

Контрольная сумма:

Для проверки правильности изображения используется контрольная сумма:

- Прочитайте биты (0 и 1) каждого столбца изображения и переведите их в числа.
- Если полученное число четное, закрасьте соответствующий квадрат в черный цвет.

Пример для матрицы 3x3:

```
1 0 1
1 1 0
1 0 0
```

- Контрольная сумма (по столбцам):
 - Первый столбец: `111` (в двоичной) = `7` (в десятичной), `7 % 2 = 1` (нечетное)
 - Второй столбец: `010` = `2`, `2 % 2 = 0` (четное, закрасьте в черный)
 - Третий столбец: `100` = `4`, `4 % 2 = 0` (четное, закрасьте в черный)

Размер изображения:

- Пиксельная графика будет иметь размер 256x256 пикселей.
- Каждый пиксель изображения будет размером 8x8 пикселей.

Ваше решение должно быть в виде HTML-файла с кодом, который выполняет описанные действия.

Формат входных данных:

https://t.me/postypashki_old https://t.me/postypashki_old https://t.me/postypashki_old
window.content - объект со следующими атрибутами:

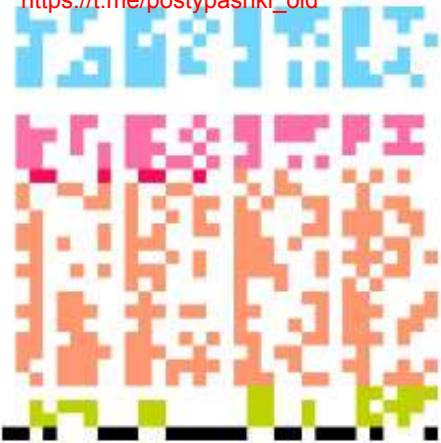
- type: 32 символа
- name: 16 символов
- age: 2 символа
- skills: 64 символа
- date: 10 символов

Пример:

```
const window.content = {
  type: "Frontend React Developer", // #70d6ff
  name: "Frontender Name",         // #ff70a6
  age: "12",                        // #ff0a54
  skills: "HTML, CSS, JavaScript, JQuery, PHP, Canvas, Effector, Node.js.", // #ff9770
  date: "07.02.2025"                // #bfd200
}

const checksum = "..."; // #000000
```

Пример результата:



По ссылке можно скачать архив с шаблоном. В качестве решения ожидается html файл с вашим кодом.