Free Code Camp Dieciséis problemas básicos del algoritmo JS

Registre, el algoritmo básico durante el proceso de despacho de aduanas de la FCC, un total de 15 preguntas. Está familiarizado con algunas API y algunas con la estructura del script JS.

**1.Invertir una cuerda: invertir una cuerda**

**Reclamación:**  
primero convierta la cadena en una matriz, luego use el método inverso de la matriz para invertir el orden de la matriz y, finalmente, convierta la matriz en una cadena.  
El resultado debe ser una cadena

**Ideas:**

1. Split () directamente en una matriz
2. Use reverse () para voltear la matriz
3. Combine la matriz en una cadena con join ().
4. **function reverseString(str) {**
5. **var newstr = str.split("");**
6. **newstr.reverse();**
7. **str = newstr.join("");**
8. **return str;**
9. **}**
10. **reverseString("hello");**

Vale la pena señalar que join () debe agregar un carácter nulo al parámetro para indicar una conexión directa. Úselo directamente, usará el predeterminado, use una coma**“，”**Apartado.

La escritura fue un poco más complicada, y luego la optimicé refiriéndome a los métodos de escritura y la estructura de la cadena de otras personas.

1. **function reverseString(str) {**
2. **return str.split("").reverse().join("");**
3. **}**
4. **reverseString("hello");**

**2.Factorializar un número: calcular el factorial de un entero**

**Reclamación:**  
Si la letra n se usa para representar un número entero, el factorial representa el producto de todos los números enteros menores o iguales an.

**Ideas:**

1. Calcule el factorial, comience desde 1, declare el valor inicial de la variable como 1 y el valor total de la variable factorial
2. Utilice un bucle for para recorrer la longitud del valor pasado por el parámetro y realizar una multiplicación acumulativa.
3. **function factorialize(num) {**
4. **var i=1;**
5. **var all=1;**
6. **for( i=1; i<=num; i++){**
7. **all \*=i;**
8. **}**
9. **return all;**
10. **}**
11. **factorialize(5);**

**3.Compruebe si hay palíndromos - Compruebe si hay palíndromos**

**Reclamación:**  
Si la cadena dada es un palíndromo, devuelve verdadero; de lo contrario, devuelve falso.

Si una cadena ignora la puntuación, las mayúsculas y minúsculas, y los espacios, y se lee exactamente igual que al revés, entonces la cadena es palíndromo.

**Ideas:**

1. Compruebe el palíndromo, la cuerda debe procesarse. Convierta minúsculas (toLowerCase ()), expresiones regulares que procesan caracteres especiales, y luego divídalo en matrices (split ()).
2. Después de dividir en una matriz, voltee la matriz (reverse ()) y combínela en una cadena (jion ()).
3. Compare y juzgue la cadena combinada con la cadena de parámetros pasada.
4. **function palindrome(str) {**
6. **var muevetr = str.replace (/ \ W + / g, ''); *// La expresión regular filtra todos los caracteres que no son palabras y los reemplaza con una cadena vacía***
7. **movetr = movetr.replace (/ \ \_ + / g, ''); *// La expresión regular filtra todos los caracteres de subrayado y los reemplaza con una cadena vacía***
8. **movestr = movestr.toLowerCase();**
9. **var spstr = movestr.split('');**
10. **var restr = spstr.reverse();**
11. **var newstr = restr.join('');**
12. **if(newstr===movestr){**
13. **return true;**
14. **}**
15. **else{**
16. **return false;**
17. **}**
18. **}**
19. **palindrome("eye");**

Es la primera vez que lo implementamos según la idea. Aunque se realiza, es feo. Intenta optimizarlo. Dos ideas de optimización:

Uno:

1. **function palindrome(str) {**
3. **str = str.replace (/ [^ A-Za-z0-9] / g, ''). toLowerCase (); *// La coincidencia de índice directo de expresión regular comienza con AZ, az, 0-9 cadenas y números, la cadena se convierte a minúsculas***
4. **for (var i = 0; i <str.length / 2; i ++) {*// Toma la mitad de la longitud de la cadena y haz un bucle***
5. **if (str [i]! == str [str.length-i-1]) {*// Determinar si la posición correspondiente de la cadena es igual según la regla del palíndromo (el primer dígito es el último dígito, el segundo dígito es el segundo lugar recíproco, y así sucesivamente)***
6. **return false;**
7. **}**
8. **}**
9. **return true;**
10. **}**
11. **palindrome("eye");**

Segundo:

1. **function palindrome(str) {**
3. **var re = / [\ W \ s \_] / gi; *// carácter de escape \ W, caracteres que no son palabras y caracteres de espacio en blanco y no distinguen entre mayúsculas y minúsculas***
4. **str = str.replace(re,"");**
5. **return str.toLowerCase() === str.split("").reverse().join("").toLowerCase();**
6. **}**
7. **palindrome("eye");**

Fue principalmente el procesamiento de expresiones regulares, lo que llevó mucho tiempo. La regularidad sigue siendo débil y debe reforzarse.

**4.Encuentre la palabra más larga de una cadena: encuentre la palabra más larga**

**Reclamación:**

Encuentra la palabra más larga de la oración y devuelve su longitud.  
El valor de retorno de la función debe ser un número.

**Ideas:**

1. Divida la oración, divida la cadena en un segmento de cadenas, divida las oraciones con espacios (split ('')) y guárdelas en una matriz.
2. Declare un valor inicial, recorra la matriz, compare el valor inicial con la longitud de cada cadena y asigne el mayor al valor inicial. Después de atravesar el valor inicial, se puede obtener el valor máximo.
3. **function findLongestWord(str) {**
4. **var newstr = str.split (""); *// Divide la cadena en una matriz, separada por espacios***
5. **var longWord = 0; *// Establecer el mínimo inicial***
6. **for (var i = 0; i <newstr.length; i ++) {*// for loop, compare la palabra más larga***
7. **if(newstr[i].length>longWord){**
8. **longWord=newstr[i].length;**
9. **}**
10. **}**
11. **return longWord;**
12. **}**
13. **findLongestWord("The quick brown fox jumped over the lazy dog");**

Según la idea, no siempre es lo suficientemente bueno, pero aún debe optimizarse:

1. **function findLongestWord(str) {**
2. **var arr = str.split (""); *// La cadena se divide en una matriz, separada por espacios***
3. **arr = arr.map (function (item) {*// El método map coincide con la longitud de cada cadena en la matriz y devuelve la longitud del carácter***
4. **return item.length;**
5. **}). sort (function (a, b) {*// El método de la función sort () ordena los números de mayor a menor,***
6. **return b-a;**
7. **});**
8. **}**

**5.Título de una oración: escriba en mayúscula la primera letra de la palabra en la oración**

**Reclamación:**  
escribe en mayúscula la primera letra de la palabra en la oración  
Asegúrese de que la primera letra de cada palabra de la cadena esté en mayúscula y el resto en minúsculas.  
Lo mismo se aplica a conectores como 'el' y 'de'.

**Ideas:**

1. Primero divida la oración y modifique individualmente la primera letra de cada cadena a mayúsculas y el resto a minúsculas.
2. Convierta la cadena después de la conversión de mayúsculas y minúsculas en una oración.
3. **function titleCase(str) {**
4. **var str1 = str.split (""); *// Divide la cadena en una matriz***
5. **var str2;**
6. **var newstr;**
7. **for(var i=0; i < str1.length;i++){**
8. **str2 = str1 [i] .split (""); *// Repetir las palabras de la matriz str1 y dividirlas en caracteres individuales en la matriz str2***
9. **str2 [0] = str2 [0] .toUpperCase (); *// Capitaliza el primer carácter de un solo carácter en la matriz str2***
10. **for (var j = 1; j <str2.length; j ++) {*// array str2 carácter único y los caracteres restantes están en minúsculas***
11. **str2[j] = str2[j].toLowerCase();**
12. **}**
13. **str1 [i] = str2.join (""); *// La matriz de un solo carácter se fusiona en palabras;***
14. **}**
15. **newstr = str1.join (""); *// La matriz de palabras se fusiona en una cadena;***
16. **return newstr;**
17. **}**
18. **titleCase("I'm a little tea pot");**

Hmm ... todavía tengo que optimizar, dos bucles for, tengo ganas de escribir una lista larga.

1. **function titleCase(str) {**
2. ***// toLowerCase () se convierte todo a minúsculas, split () divide la cadena en una matriz y el método map () devuelve una nueva matriz;***
3. **var newstr=str.toLowerCase().split(" ").map(function(item){**
4. **return item[0].toString().toUpperCase()+item.slice(1);**
5. ***// Obtenga la primera letra del arreglo y conviértala a mayúsculas***
6. ***// Luego agregue el siguiente contenido en minúsculas (el método slice () obtiene el contenido del rango, 1 significa comenzar desde la segunda letra)***
7. **}). join (""); *// el método join () se combina en una cadena***
8. **return newstr;**
9. **}**
10. **titleCase("I'm a little tea pot");**

**6. Devuelva los números más grandes en matrices: encuentre el número más grande en varias matrices**

**Reclamación:**  
Encuentra el número más grande en varias matrices  
La matriz grande de la derecha contiene 4 matrices pequeñas, encuentre el valor máximo en cada matriz pequeña y luego concatenarlas para formar una nueva matriz.

**Ideas:**

1. Para múltiples arreglos, primero procese los subarreglos. Atraviesa la matriz grande y ordena primero cada submatriz.
2. Agregue el más grande de cada submatriz ordenada a la nueva matriz.
3. **function largestOfFour(arr) {**
4. **var newArr=[];**
5. **for(var i=0;i<arr.length;i++){**
6. **arr[i].sort(function(a,b){**
7. **return b-a;**
8. **});**
9. **newArr.push(arr[i][0]);**
10. **}**
11. **return newArr;**
12. **}**
13. **largestOfFour([[4, 5, 1, 3], [13, 27, 18, 26], [32, 35, 37, 39], [1000, 1001, 857, 1]]);**

**7. Confirme el final: compruebe el final de la cadena**

**Reclamación:**

Determine si una cadena de caracteres (str) termina con una cadena de caracteres especificada (destino).  
Si es así, devuelve verdadero; si no, devuelve falso.  
Consejo: String.substr ()

**Ideas:**

1. La idea es muy simple, obtenga directamente la longitud de la cadena final a comparar, y luego encuentre el final de la longitud correspondiente en la cadena larga y compárelo.
2. **function confirmEnding(str, target) {**
3. **var starLen =str.length - target.length;**
4. **var strArr=str.substr(starLen);**
5. **if(strArr===target){**
6. **return true;**
7. **}**
8. **else{**
9. **return false;**
10. **}**
11. **}**
12. **confirmEnding("Bastian", "ian");**

También puede usar el método slice () usado antes,  
slice (star, end), start número negativo significa comenzar a contar desde el final de la cadena, end es opcional y el final  
Por ejemplo:  
var str=“abcd”；  
var newstr=str.slice（-2）  
newstr

Lo optimicé y se siente bien.

1. **function confirmEnding(str, target) {**
2. **if(str.substr(-target.length) === target){**
3. **return true;**
4. **}**
5. **else{ return false; }**
6. **}**
7. **confirmEnding("Bastian", "n");**

**8. Repetir una cadena repetir una cadena - Repetir cadena de salida**

**Reclamación:**  
Repite la cadena de salida (lo importante se dice 3 veces)  
Repite una cadena especificada num veces y devuelve una cadena vacía si num es un número negativo.

--  
**Ideas:**  
1. La idea es bastante clara, repita la cadena de salida n veces y devuelva una cadena vacía si n es negativo;

1. **function repeat(str, num) {**
2. **var newstr =str;**
3. **if(num >= 0){**
4. **for(var i=0;i<num-1; i++){**
5. **str+= newstr;**
6. **}**
7. **return str;**
8. **}**
9. **else{ return ""}**
10. **}**
11. **repeat("abc", 3);**

Después de pensarlo, uno más si se puede omitir el juicio.

1. **function repeat(str, num) {**
2. **var newStr='';**
3. **for(var i=1;i<=num;i++){**
4. **newStr+=str;**
5. **}**
6. **return newStr;**
7. **}**

**9. Truncar una cadena: truncar una cadena**

**Reclamación:**  
Truncar cadena  
(use Ritz para cortar el retiro opuesto)

Si la longitud de la cadena es más larga que el número de parámetro especificado, la parte extra está representada por ...  
Recuerde, los tres puntos insertados al final de la cadena también se incluyen en la longitud de la cadena.  
Sin embargo, si el número de parámetro especificado es menor o igual a 3, los tres puntos agregados no se incluirán en la longitud de la cadena.

Consejo: slice ()

**Ideas:**

1. Realice un juicio de dos pasos para determinar si la cadena es mayor que el valor dado, si es mayor que el valor, continúe con el siguiente paso y, si es menor, regrese directamente.
2. Cuando se juzga que la cadena de caracteres es mayor que el valor dado, se juzga si es mayor que el valor mínimo 3.
3. **function truncate(str, num) {**
4. **var newStr= "";**
5. **if(str.length>num){**
6. **if(num<=3){**
7. **newStr =str.slice(0,num)+"...";**
8. **}**
9. **else{**
10. **newStr =str.slice(0,num-3)+"...";**
11. **}**
12. **return newStr;**
13. **}**
14. **else{**
15. **return str;**
16. **}**
17. **}**
18. **truncate("A-tisket a-tasket A green and yellow basket", 11);**

Simplificado un poco

1. **function truncate(str, num) {**
2. **var result='';**
3. **if(str.length<=num){**
4. **result=str;**
5. **}else{**
6. **result=num>3?str.slice(0,num-3)+'...':str.slice(0,num)+'...';**
7. **}**
8. **return result;**
9. **}**
10. **truncate("A-tisket a-tasket A green and yellow basket", 11);**

**10.Mono grueso: divide la matriz, el mono come plátano**

**Reclamación:**  
El mono come un plátano y divide la matriz  
(El mono se come el plátano pero se rompe en varios trozos para comer)  
Divide un arreglo de arreglo en varios bloques de arreglo de acuerdo con el tamaño de arreglo especificado.

Por ejemplo: chunk ([1,2,3,4], 2) = [[1,2], [3,4]];  
chunk([1,2,3,4,5],2)=[[1,2],[3,4],[5]];

Consejos: Array.push (), Array.slice ()

**Ideas:**

1. Para dividir el número en bloques, primero divida la matriz en una matriz especificada y luego agréguela a una nueva matriz.
2. **function chunk(arr, size) {**
3. **var newArr=[];**
4. **for(var i=0; i<arr.length;i+=size){**
5. **newArr.push(arr.slice(i,i+size));**
6. **}**
7. **return newArr;**
8. **}**
9. **chunk(["a", "b", "c", "d"], 2);**

**11.Slasher Flick - matriz truncada**

**Reclamación:**  
devuelve los elementos restantes después de truncar n elementos en una matriz, el truncamiento comienza en el índice 0.

Consejos: Array.slice (), Array.splice ()

**Ideas:**

1. Es bastante simple, principalmente porque está familiarizado con el uso de api, y puede usar ambas apis aquí.

Utilice slice ()

1. **function slasher(arr, howMany) {**
2. **return arr.slice(howMany);**
3. **}**
4. **slasher([1, 2, 3], 2);**

Usar empalme ()

1. **function slasher(arr, howMany) {**
2. **arr.splice(0,howMany);**
3. **return arr;**
4. **}**
5. **slasher([1, 2, 3], 2);**

**12.Mutaciones - Comparar cadenas**

**Reclamación:**

Comparar cadenas  
(Toad puede comerse a compañeros y oponentes)

Si el primer elemento de cadena de la matriz contiene todos los caracteres del segundo elemento de cadena, la función devuelve verdadero.

Por ejemplo, ["hola", "Hola"] debe devolver verdadero, porque todos los caracteres de la segunda cadena se pueden encontrar en la primera cadena ignorando las mayúsculas y minúsculas.

["hola", "hey"] debería devolver falso porque la cadena "hola" no contiene el carácter "y".  
["Alien", "line"] debe devolver verdadero, porque todos los caracteres en "line" se pueden encontrar en "Alien".

Sugerencia: indexOf ()

**Ideas:**

1. Parece que no hay idea, anótelo paso a paso según sea necesario.

**indexOf()** El método devuelve el índice de la primera aparición del valor especificado en el objeto String que llama y comienza a buscar en fromIndex.  
Si no se encuentra el valor, devuelve -1

1. **function mutation(arr) {**
2. **arr[0]=arr[0].toLowerCase();**
3. **arr[1]=arr[1].toLowerCase();**
4. **for(var i=0;i<arr[1].length; i++){**
5. **if(arr[0].indexOf(arr[1][i])==-1){**
6. **return false;**
7. **}**
8. **}**
9. **return true;**
10. **}**
11. **mutation(["hello", "hey"]);**

**13.Falsy Bouncer - filtrar los valores falsos de la matriz**

**Reclamación:**

Filtrar valores falsos de matriz  
(Rey mono verdadero y falso)  
Elimina todos los valores falsos en la matriz.

En JavaScript, los valores falsos son false, null, 0, "", undefined y NaN.

Sugerencias: Objetos booleanos, Array.filter ()

**Ideas:**

1. Determine si el valor es verdadero por booleano y regrese a la nueva matriz si coincide. Se puede lograr Array.fillter ().
2. **arr = arr.filter(filterFalsy);**
3. **function filterFalsy(item){**
4. **return Boolean(item) === true;**
5. **}**
6. **return arr;**
7. **bouncer([7, "ate", "", false, 9]);**

**14.Busque y destruya: destruya la matriz**

**Reclamación:**  
destruye la matriz  
¡El mortero de Jinks!  
implementa una función de destrucción, el primer parámetro es la matriz que se va a destruir y los parámetros restantes son los valores que se van a destruir.

Sugerencias: objeto Arguments, Array.filter () Array.filter ()

**Ideas:**

1. Juzgue el número de parámetros a través de argumentos y luego utilícelo como condición de ciclo
2. Array.filter () filtra los parámetros pasados ​​y devuelve el resto.
3. **function destroyer(arr) {**
4. **var arg =arguments;**
5. **for(var i =1;i<arg.length;i++){**
6. **arr = arr.filter(function(x){**
7. **return x!==arg[i];**
8. **});**
9. **}**
10. **return arr;**
11. **}**
12. **destroyer([1, 2, 3, 1, 2, 3], 2, 3);**

**15 ¿A dónde pertenezco? Ordenación de matriz para encontrar índice**

**Reclamación:**  
Ordena la matriz y busca el índice del elemento  
¿Dónde estoy?  
Ordene la matriz primero, luego busque la posición del valor especificado en la matriz y finalmente devuelva el índice correspondiente a la posición.  
Ejemplo: donde ([1,2,3,4], 1.5) debería devolver 1. Porque 1,5 se inserta en la matriz [1,2,3,4] y se convierte en [1,1.5,2,3,4], y el valor de índice correspondiente a 1,5 es 1.  
De manera similar, donde ([20,3,5], 19) debería devolver 2. Debido a que la matriz se ordenará primero en [3,5,20], 19 se inserta en la matriz [3,5,20] y se convierte en [3,5,19,20], y el valor del índice correspondiente a 19 es 2 .

Consejo: sort ()

**Ideas:**

1. Primero presione el número que se agregará a la matriz, luego use sort () para ordenar, y luego use indexOf () para encontrar el índice.
2. **function where(arr, num) {**
3. **arr.push(num);**
4. **arr.sort(function(a,b){return a-b;});**
5. **return arr.indexOf(num);**
6. **}**
7. **where([40, 60], 50);**

**16 Cifrado César - Cifrado César**

Reclamación:  
Contraseña de César

(Que Dios pertenezca a Dios, César pertenezca a César)  
A continuación, presentaremos el cifrado Caesar, que es popular en todo el mundo, también llamado cifrado por desplazamiento.  
La contraseña de cambio significa que las letras de la contraseña se cambiarán en la cantidad especificada.  
Un caso común es la contraseña ROT13, donde las letras cambiarán 13 posiciones. From'A '↔'N', 'B' ↔'O ', y así sucesivamente.  
Escriba una función ROT13 para ingresar la cadena cifrada y generar la cadena descifrada.  
Todas las letras están en mayúsculas. No convierta ningún carácter que no sea una letra (por ejemplo: espacios, signos de puntuación). Si encuentra estos caracteres especiales, omítalos.

**Consejos: charCodeAt (), String.fromCharCode ()**

Ideas:

1. Comience con la idea de ROT13, primero divida cada carácter del parámetro de cadena entrante en Unicode y guárdelo en una matriz.
2. Procese esta matriz, el rango de código Unicode de letras mayúsculas es 65 ~ 90, recorra la matriz y regrese directamente si no cumple con este rango.
3. Si cumple con el rango, se procesa la regla ROT13 y se agrega 13 a cada valor de código Unicode, cuando excede el rango, comienza desde el principio.

Primera edición escrita:

1. **function rot13(str) {**
2. **var arr=[];**
3. **var arr2=[];**
4. **for(var i =0;i<str.length;i++){**
5. **arr.push(str.charCodeAt(i));**
6. **}**
7. **arr = arr.map(function(x){**
9. **if(x<65 || x>90){**
10. **return x;**
11. **}**
12. **else{**
13. **if(x+13>90){**
14. **return x+13-26;**
15. **}**
16. **else{**
17. **return x+13;**
18. **}**
19. **}**
20. **});**
22. **for(var j = 0; j<arr.length;j++){**
23. **arr2.push(String.fromCharCode(arr[j]));**
24. **}**
25. **str =arr2.join('');**
26. **return str;**
27. **}**
28. **rot13("SERR PBQR PNZC");**

Después de la optimización:

1. **function rot13(str) {**
3. **var arr =[];**
5. **for(var i=0;i<str.length;i++){**
6. **if(str.charCodeAt(i)<65 ||str.charCodeAt(i)>90){**
7. **arr.push(String.fromCharCode(str.charCodeAt(i)));**
8. **}**
9. **else if(str.charCodeAt(i)>77){**
10. **arr.push(String.fromCharCode(str.charCodeAt(i)-13));**
11. **}**
12. **else{**
13. **arr.push(String.fromCharCode(str.charCodeAt(i)+13));**
14. **}**
15. **}**
16. **return arr.join('');**
17. **}**
18. **rot13("SERR PBQR PNZC");**

Está hecho. Después de completar estas dieciséis preguntas muy básicas, he aprendido mucho.