

1, 判断字符串是否是这样组成的, 第一个必须是字母, 后面可以是字母、数字、下划线, 总长度为 5-20

```
var reg = /^[a-zA-Z][a-zA-Z_0-9]{4,19}$/;  
reg.test("a1a__a1a__a1a__a1a__");
```

2, 截取字符串 abcdefg 的 efg

```
var str = "abcdefg";  
  
if (/efg/.test(str))  
{  
  
    var efg = str.substr(str.indexOf("efg"), 3);  
  
    alert(efg);  
}
```

3, 判断一个字符串中出现次数最多的字符, 统计这个次数

//将字符串的字符保存在一个 hash table 中, key 是字符, value 是这个字符出现的次数

```
var str = "abcdefgaddda";  
  
var obj = {};  
  
for (var i = 0, l = str.length; i < l; i++)  
{  
  
    var key = str[i];  
  
    if (!obj[key])
```

```
{
    obj[key] = 1;
}

else
{
    obj[key]++;
}
}

/*遍历这个 hash table , 获取 value 最大的 key 和 value*/

var max = -1;

var max_key = "";

var key;

for (key in obj)
{
    if (max < obj[key])
    {
        max = obj[key];
        max_key = key;
    }
}

alert("max:"+max+" max_key:"+max_key);
```

4 . IE 与 FF 脚本兼容性问题

(1) window.event :

表示当前的事件对象，IE 有这个对象，FF 没有，FF 通过给事件处理函数传递事件对象

(2) 获取事件源

IE 用 srcElement 获取事件源，而 FF 用 target 获取事件源

(3) 添加，去除事件

IE : element.attachEvent(" onclick " , function)
element.detachEvent("onclick" , function)

FF : element.addEventListener(" click " , function, true)
element.removeEventListener("click" , function, true)

(4) 获取标签的自定义属性

IE : div1.value 或 div1["value"]

FF : 可用 div1.getAttribute("value")

(5) document.getElementById()和 document.all[name]

IE;document.getElementById()和 document.all[name]均不能获取
div 元素

FF : 可以

(6) input.type 的属性

IE : input.type 只读

FF : input.type 可读写

(7) innerText textContent outerHTML

IE : 支持 innerText, outerHTML

FF : 支持 textContent

(8) 是否可用 id 代替 HTML 元素

IE : 可以用 id 来代替 HTML 元素

FF : 不可以

这里只列出了常见的，还有不少，更多的介绍可以参看 JavaScript 在 IE 浏览器和 Firefox 浏览器中的差异总结

5, 规避 javascript 多人开发函数重名问题

(1) 可以开发前规定命名规范，根据不同开发人员开发的功能在函数前加前缀

(2) 将每个开发人员的函数封装到类中，调用的时候就调用类的函数，即使函数重名只要类名不重复就 ok

6, javascript 面向对象中继承实现

javascript 面向对象中的继承实现一般都使用到了构造函数和 Prototype 原型链，简单的代码如下：

```
function Animal(name)
```

```
{
```

```
    this.name = name;
```

```
}
```

```
Animal.prototype.getName = function() {alert(this.name)}
```

```
function Dog() {};  
  
Dog.prototype = new Animal("Buddy");  
  
Dog.prototype.constructor = Dog;  
  
var dog = new Dog();
```

7 . FF 下面实现 outerHTML

FF 不支持 outerHTML , 要实现 outerHTML 还需要特殊处理

思路如下 :

在页面中添加一个新的元素 A , 克隆一份需要获取 outerHTML 的元素 , 将这个元素 append 到新的 A 中 , 然后获取 A 的 innerHTML 就可以了。

SPANDIV

SPAN

P

8 , 编写一个方法 求一个字符串的字节长度

假设 :

一个英文字符占用一个字节 , 一个中文字符占用两个字节

```
function GetBytes(str)  
{  
  
    var len = str.length;  
  
    var bytes = len;  
  
    for(var i=0; i<len;i++)
```

```

    {
        if (str.charCodeAt(i) > 255) bytes++;
    }

    return bytes;
}

alert(GetBytes("你好,as"));

```

9 . 编写一个方法 去掉一个数组的重复元素

```

var arr = [1,1,2,3,3,2,1];

Array.prototype.unique = function()
{
    var ret = [];

    var o = {};

    var len = this.length;

    for (var i=0; i<len;i++)

    {
        var v = this[i];

        if (!o[v])

        {
            o[v] = 1;

            ret.push(v);
        }
    }
}

```

```
    }  
  
    return ret;  
  
};  
  
alert(arr.unique());
```

10 , 写出 3 个使用 this 的典型应用

(1)在 html 元素事件属性中使用 , 如

(2)构造函数

```
function Animal(name, color) {  
  
    this.name = name;  
  
    this.color = color;  
  
}
```

(3)CSS expression 表达式中使用 this 关键字

12 , 如何显示/隐藏一个 DOM 元素?

```
el.style.display = "";
```

```
el.style.display = "none";
```

el 是要操作的 DOM 元素

13 , JavaScript 中如何检测一个变量是一个 String 类型?请写出函数实现

String 类型有两种生成方式 :

(1)

```
Var str = "hello world" ;
```

(2)

```
Var str2 = new String( "hello world" );
```

```
function IsString(str){return (typeof str == "string" || str.constructor  
== String);}
```

```
var str = "";
```

```
alert(IsString(1));
```

```
alert(IsString(str));
```

```
alert(IsString(new String(str)));
```

14，网页中实现一个计算当年还剩多少时间的倒数计时程序，要求网页上实时动态显示 “××年还剩××天××时××分××秒”

15，补充代码，鼠标单击 Button1 后将 Button1 移动到 Button2 的后面

16，JavaScript 有哪几种数据类型

简单：Number，Boolean，String，Null，Undefined

复合：Object，Array，Function

17，下面 css 标签在 JavaScript 中调用应如何拼写，border-left-color，
-moz-viewport

borderLeftColor

mozViewport

18 , JavaScript 中如何对一个对象进行深度 clone

```
function cloneObject(o)
{
    if( !o || 'object' !== typeof o)
    {
        return o;
    }
    var c = 'function' === typeof o.pop ? [] : {};
    var p, v;
    for(p in o)
    {
        if(o.hasOwnProperty(p))
        {
            {
                v = o[p];
                if(v && 'object' === typeof v)
                {
                    c[p] = Ext.ux.clone(v);
                }
            }
            else
            {
                {
```

```

        c[p] = v;
    }
}
}

return c;
};

```

19, 如何控制 alert 中的换行

```
\n alert( "p\np" );
```

20, 请实现, 鼠标点击页面中的任意标签, alert 该标签的名称.(注意兼容性)

```
SPANDIV
```

```
SPAN
```

```
P
```

21, 请编写一个 JavaScript 函数 parseQueryString, 它的用途是把 URL 参数解析为一个对象, 如:

```

var url = "http://witmax.cn/index.php?key0=0&key1=1&key2=2";

function parseQueryString(url)
{
    var params = {};
    var arr = url.split("?");

```

```

if (arr.length <= 1)

    return params;

arr = arr[1].split("&");

for(var i=0, l=arr.length; i<l;i++)

    {

        var a = arr[i].split("=");

        params[a[0]] = a[1];

    }

    return params;

}

var url = "http://witmax.cn/index.php?key0=0&key1=1&key2=2";

var ps = parseQueryString(url);

alert(ps["key1"]);

```

22 , ajax 是什么? ajax 的交互模型? 同步和异步的区别? 如何解决跨域问题?

Ajax 是多种技术组合起来的一种浏览器和服务端交互技术, 基本思想是允许一个互联网浏览器向一个远程页面/服务做异步的 http 调用, 并且用收到的数据来更新一个当前 web 页面而不必

刷新整个页面。该技术能够改进客户端的体验。包含的技术:

XHTML : 对应 W3C 的 XHTML 规范, 目前是 XHTML1.0。

CSS : 对应 W3C 的 CSS 规范, 目前是 CSS2.0

DOM : 这里的 DOM 主要是指 HTML DOM , XML DOM 包括在下面的

XML 中

JavaScript : 对应于 ECMA 的 ECMAScript 规范

XML : 对应 W3C 的 XML DOM、XSLT、XPath 等等规范

XMLHttpRequest : 对应 WhatWG 的 Web Applications1.0 规范
(<http://whatwg.org/specs/web-apps/current-work/>)

同步 : 脚本会停留并等待服务器发送回复然后再继续

异步 : 脚本允许页面继续其进程并处理可能的回复

跨域问题简单的理解就是因为 JS 同源策略的限制 , a.com 域名下的 JS 无法操作 b.com 或 c.a.com 下的对象。

PS : (1)如果是端口或者协议造成的跨域问题前端是无能为力的

(2) 在跨域问题上 , 域仅仅通过 URL 的首部来识别而不会尝试判断相同的 IP 地址对应的域或者两个域是否对应一个 IP

前端对于跨域的解决办法 :

(1) document.domain+iframe

(2) 动态创建 script 标签

23 , 什么是闭包?下面这个 ul , 如何点击每一列的时候 alert 其 index?

这是第一条

这是第二条

这是第三条

内部函数被定义它的函数的外部区域调用的时候就产生了闭包。

```
(function A() {
```

```
var index = 0;

var ul = document.getElementById("test");

var obj = {};

for (var i = 0, l = ul.childNodes.length; i < l; i++)

{

    if (ul.childNodes[i].nodeName.toLowerCase() == "li")

    {

        var li = ul.childNodes[i];

        li.onclick = function()

        {

            index++;

            alert(index);

        }

    }

}

})();
```

24, 请给出异步加载 js 方案, 不少于两种

默认情况 javascript 是同步加载的, 也就是 javascript 的加载时阻塞的, 后面的元素要等待 javascript 加载完毕后才能进行再加载, 对于一些意义不是很大的 javascript, 如果放在页头

会导致加载很慢的话，是会严重影响用户体验的。

异步加载方式：

(1) defer，只支持 IE

(2) async：

(3) 创建 script，插入到 DOM 中，加载完毕后 callBack，见代码：

```
function loadScript(url, callback)
{
    var script = document.createElement("script")
    script.type = "text/javascript";
    if (script.readyState)
    { //IE
        script.onreadystatechange = function()
        {
            if (script.readyState == "loaded" ||script.readyState ==
"complete")
            {
                script.onreadystatechange = null;
                callback();
            }
        };
    }
    else
```

```

    {

        //Others: Firefox, Safari, Chrome, and Opera

        script.onload = function()

            {

                callback();

            };

        }

        script.src = url;

        document.body.appendChild(script);

    }

```

25 , 请设计一套方案 , 用于确保页面中 JS 加载完全。

```

var n = document.createElement("script");

n.type = "text/javascript";

//以上省略部分代码

//ie 支持 script 的 readystatechange 属性 (IE support the
readystatechange event for script and css nodes)

if(ua.ie)

{

    n.onreadystatechange = function()

    {

        var rs = this.readyState;

```

```

        if('loaded' === rs || 'complete'===rs)

            {

                n.onreadystatechange = null;

                f(id,url); //回调函数

            }

        };

        //省略部分代码

        //safari 3.x supports the load event for script nodes(DOM2)

        n.addEventListener('load',function(){

            f(id,url);

        });

        //firefox and opera support onload(but not dom2 in ff) handlers
for
        //script nodes. opera, but no ff, support the onload event for link
        //nodes.

    }

    else

    {

        n.onload = function(){f(id,url);};

    }

```

26 . js 中如何定义 class,如何扩展 prototype?

`Ele.className = "****" ; //****在 css 中定义 , 形式如下 : .**** {...}`

`A.prototype.B = C;`

A 是某个构造函数的名字

B 是这个构造函数的属性

C 是想要定义的属性的值

27 , 如何添加 html 元素的事件,有几种方法.

(1) 为 HTML 元素的事件属性赋值

(2) 在 JS 中使用 `ele.on*** = function() {...}`

(3) 使用 DOM2 的添加事件的方法 `addEventListener` 或 `attachEvent`

28 , `document.write` 和 `innerHTML` 的区别

`document.write` 只能重绘整个页面

`innerHTML` 可以重绘页面的一部分

29 , 多浏览器检测通过什么?

(1) `navigator.userAgent`

(2) 不同浏览器的特性 , 如 `addEventListener`

30 , js 的基础对象有那些, `window` 和 `document` 的常用的方法和属性列出来

`String, Number, Boolean`

`Window:`

方 法 :

setInterval,setTimeout,clearInterval,clearTimeout,alert,confirm,open

属性：name,parent,screenLeft,screenTop,self,top,status

Document 方法：

createElement,execCommand,getElementById,getElementsByTagName,
getElementsByTagName,write,writeln

属性：cookie,doctype,documentElement,readyState,URL,

31 , 前端开发的优化问题

(1) 减少 http 请求次数：css spirit,data uri

(2) JS , CSS 源码压缩

(3) 前端模板 JS+数据 ,减少由于 HTML 标签导致的带宽浪费 ,前端用
变量保存 AJAX 请求结果 ,每次操作本地变量 ,不用请求 ,减少请求次数

(4) 用 innerHTML 代替 DOM 操作 ,减少 DOM 操作次数 ,优化
javascript 性能

(5) 用 setTimeout 来避免页面失去响应

(6) 用 hash-table 来优化查找

(7) 当需要设置的样式很多时设置 className 而不是直接操作 style

(8) 少用全局变量

(9) 缓存 DOM 节点查找的结果

(10) 避免使用 CSS Expression

(11) 图片预载

(12) 避免在页面的主体布局中使用 table , table 要等其中的内容完全

下载之后才会显示出来，显示比 div+css 布局慢

32，如何控制网页在网络传输过程中的数据量

启用 GZIP 压缩

保持良好的编程习惯，避免重复的 CSS，JavaScript 代码，多余的 HTML

标签和属性

33，Flash、Ajax 各自的优缺点，在使用中如何取舍？

Ajax 的优势

- (1) 可搜索型
- (2) 开放性
- (3) 费用
- (4) 易用性
- (5) 易于开发

Flash 的优势

- (1) 多媒体处理
- (2) 兼容性
- (3) 矢量图形 比 SVG，Canvas 优势大很多
- (4) 客户端资源调度，比如麦克风，摄像头

亚信 web 前端面试题

1.判断字符串以字母开头，后面可以是数字，下划线，字母，长度为 6-30

2.请举例使用 callee 属性实现函数的递归使用

3.Ajax 同步和异步的区别，如何解决跨域问题？

4.apply 和 call 方法有形似之处，请说出二者的区别

5.请解读一下 javascript 代码，并之处问题所在

```
var Obj=function(msg){  
    this.msg=msg;  
    this.shout=function(){  
        alert(this.msg);  
    }  
    this.waitAndShout=function(){  
        setTimeout(this.shout, 2000);  
    }  
}  
  
var aa=new Obj("abc");  
  
aa.waitAndShout();
```

6.请给 Array 本地对象增加一个原型方法，他的用途是删除数组中重复的条目，并将新的数组返回

7.以下代码中 end 字符串什么时候输出

[查看源代码](#)

[打印](#)

[帮助](#)

```
var t=true;

setTimeout(function(){
    console.log(123);

    t=false;

},1000);

while(t)

{}

console.log('end');
```

8.你对前端开发工程师这个职位怎么理解的？前景如何？

9.前端开发的优化问题？

其他面试题集

题一

CSS

1、css 选择器优先级顺序

2、将图片 bg.jpg 作为按钮的背景图片，并把文字隐藏

```
<input type="button" value="我是按钮" />
```

4、写出你所了解的目前主流浏览器(ie6,ie7,ie8,firefox,chrome)的 css hack

JAVASCRIPT

1、表单验证

A、写一段正则表达式验证字符串匹配邮箱格式，邮箱如

admin2514@topsem.com.cn、google@gmail.com。

B、验证字符串是否为数字字符串。如 '125' , '2313.12' 等为数字字符串。

2、写出下面代码的输出结果

A、

```
Var obj={};
```

```
Var ref=obj;
```

```
obj.name=" objectA" ;
```

```
obj=["one","two","three"];
```

```
alert(ref.name);
```

```
alert(ref.length);
```

B、

```
Var outter=[];
```

```
Function clouseTest(){
```

```
Var array=["one","two","three","four"];
```

```
for(vari=0;i<array.length;i++){
```

```
Var x={};
```

```
x.no=i;
```

```
x.text=array[i];
```

```
x.invoke=function(){
```

```
alert(i);
```

```
}
```

```
outter.push(x);
```

```
}
```

```
}
```

```
//调用这个函数
```

```
clouseTest();
```

```
outter[0].invoke();
```

```
outter[3].invoke();
```

C、

```
$(function){
```

```
Var con=$( "div#panel" );
```

```
this.id=" content" ;
```

```
con.click(function(){  
  
alert(this.id);  
  
});  
  
});
```

3、操作表格 (可用 javascript 框架实现)

```
<table id=" mytable" >  
  
<tr><td>1</td><td>2</td><td>3</td></tr>  
  
<tr><td>4</td><td>5</td><td>6</td></tr>  
  
<tr><td>7</td><td>8</td><td>9</td></tr>  
  
</table>
```

将第二行，第二列的内容改为 55，背景色设置为红色。

4、算法题

A、快速排序算法

B、写一个函数实现字符串翻转

5、网站有一个广告，显示规则为：

- a、第一次进入页面，广告显示 10 秒后隐藏
- b、一个小时内第二次进入不显示广告简单描述下，实现的方法

6、列举你使用过的 javascript 操作字符串函数。

7、描述下你对 js 闭包、面向对象、继承的理解

8、你熟悉的 Javascript 框架，使用此框架实现过的最复杂的功能是？

题二

HTML+CSS

1.对 WEB 标准以及 W3C 的理解与认识

- 标签闭合、标签小写、不乱嵌套、提高搜索机器人搜索几率；
- 使用外链 css 和 js 脚本、结构行为表现的分离、文件下载与页面速度更快；
- 内容能被更多的用户所访问、内容能被更广泛的设备所访问、更少的代码和组件；
- 容易维护、改版方便，不需要变动页面内容；
- 提供打印版本而不需要复制内容、提高网站易用性；

2.xhtml 和 html 有什么区别

HTML 是一种基本的 WEB 网页设计语言 ,XHTML 是一个基于 XML 的置标语言

最主要的不同：

- XHTML 元素必须被正确地嵌套。
- XHTML 元素必须被关闭。
- 标签名必须用小写字母。
- XHTML 文档必须拥有根元素。

3.Doctype? 严格模式与混杂模式-如何触发这两种模式，区分它们有何意义?

用于声明文档使用那种规范 (html/Xhtml) 一般为 严格 过度 基于框架的
html 文档

加入 XML 声明可触发，解析方式更改为 IE5.5 拥有 IE5.5 的 bug

4.行内元素有哪些？块级元素有哪些？CSS 的盒模型？

块级元素：div p h1 h2 h3 h4 form ul

行内元素: a b br i span input select

Css 盒模型:内容，border ,margin , padding

5.CSS 引入的方式有哪些? link 和@import 的区别是?

内联 内嵌 外链 导入

区别：同时加载；前者无兼容性，后者 CSS2.1 以下浏览器不支持；Link 支持使用 javascript 改变样式，后者不可

6.CSS 选择符有哪些？哪些属性可以继承？优先级算法如何计算？内联和 important 哪个优先级高？

- 标签选择符 类选择符 id 选择符
- 继承不如指定 Id>class>标签选择
- 内联和 important 优先级高是后者优先级高

7.前端页面有哪三层构成，分别是什么？作用是什么？

结构层 Html 表示层 CSS 行为层 js

8.css 的基本语句构成是？

选择器{属性 1:值 1;属性 2:值 2;.....}

9.你做的页面在哪些浏览器测试过？这些浏览器的内核分别是什么？

Ie(Ie 内核) 火狐 (Gecko) 谷歌 (webkit) opear(Presto)

10.写出几种 IE6 BUG 的解决方法

- 双边距 BUG float 引起的 使用 display
- 像素问题 使用 float 引起的 使用 display:inline -3px
- 超链接 hover 点击后失效 使用正确的书写顺序 link visited hover active
- Ie z-index 问题 给父级添加 position:relative
- Png 透明 使用 js 代码改
- Min-height 最小高度 !Important 解决'
- select 在 ie6 下遮盖 使用 iframe 嵌套

- 为什么没有办法定义 1px 左右的宽度容器(IE6 默认的行高造成的，使用 `overflow:hidden;zoom:0.08; line-height:1px`)

11.标签上 title 与 alt 属性的区别是什么？

- Alt 当图片不显示是 用文字代表。
- Title 为该属性提供信息

12.描述 css reset 的作用和用途。

Reset 重置浏览器的 css 默认属性浏览器的品种不同 样式不同 然后重置，让他们统一

13.解释 css sprites，如何使用。

Css 精灵 把一堆小的图片整合到一张大的图片上，减轻服务器对图片的请求数量

14.浏览器标准模式和怪异模式之间的区别是什么？

- 盒子模型 渲染模式的不同
- 使用 `window.top.document.compatMode` 可显示为什么模式

15.你如何对网站的文件和资源进行优化？期待的解决方案包括：

- 文件合并
- 文件最小化/文件压缩

- 使用 CDN 托管
- 缓存的使用

16.什么是语义化的 HTML ?

直观的认识标签 对于搜索引擎的抓取有好处

17.清除浮动的几种方式，各自的优缺点

- 使用空标签清除浮动 clear:both (理论上能清楚任何标签 , , , 增加无意义的标签)
- 使用 overflow:auto (空标签元素清除浮动而不得不增加无意代码的弊端,使用 zoom:1 用于兼容 IE)
- 是用 afert 伪元素清除浮动(用于非 IE 浏览器)

Javascript

1.javascript 的 typeof 返回哪些数据类型

Object number function boolean underfind

2.例举 3 种强制类型转换和 2 种隐式类型转换？

强制 (parseInt,parseFloat,number)

隐式 (== - ===)

3.split() join() 的区别

前者是切割成数组的形式，后者是将数组转换成字符串

4.数组方法 pop() push() unshift() shift()

Push()尾部添加 pop()尾部删除

Unshift()头部添加 shift()头部删除

5.事件绑定和普通事件有什么区别

6.IE 和 DOM 事件流的区别

- 执行顺序不一样、
- 参数不一样
- 事件加不加 on
- this 指向问题

7.IE 和标准下有哪些兼容性的写法

- `var ev = ev || window.event`
- `document.documentElement.clientWidth` ||
`document.body.clientWidth`
- `var target = ev.srcElement||ev.target`

8.ajax 请求的时候 get 和 post 方式的区别

一个在 url 后面一个放在虚拟载体里面

有大小限制

安全问题

应用不同一个是论坛等只需要请求的，一个是类似修改密码的

9.call 和 apply 的区别

`Object.call(this,obj1,obj2,obj3)`

`Object.apply(this,arguments)`

10.ajax 请求时，如何解释 json 数据

使用 eval 、 parse 鉴于安全性考虑 使用 parse 更靠谱 ;eval 不建议使用，它需要讲字符串解释成代码，然后再运行此代码，很消耗性能。

11.b 继承 a 的方法

12.写一个获取非行间样式的函数

```
function getStyle(obj,attr,value)
```

```
{
```

```
    if(!value)
```

```
{  
    if(obj.currentStyle)  
    {  
        return obj.currentStyle(attr)  
    }  
    else  
    {  
        obj.getComputedStyle(attr,false)  
    }  
}  
else  
{  
    obj.style[attr]=value  
}  
}
```

13.事件委托是什么

让利用事件冒泡的原理，让自己的所触发的事件，让他的父元素代替执行！

<http://www.webasily.com/?p=78> 例子可见此链接

14.闭包是什么，有什么特性，对页面有什么影响

闭包就是能够读取其他函数内部变量的函数。

15.如何阻止事件冒泡和默认事件

cancelBubble return false

16.添加 删除 替换 插入到某个接点的方法

- obj.appendChild()
- obj.insertBefore
- obj.replaceChild
- obj.removeChild

17.解释 jsonp 的原理，以及为什么不是真正的 ajax

- 动态创建 script 标签，回调函数
- Ajax 是页面无刷新请求数据操作

18.javascript 的本地对象，内置对象和宿主对象

- 本地对象为 array obj regexp 等可以 new 实例化
- 内置对象为 global Math 等不可以实例化的
- 宿主为浏览器自带的 document,window 等

19.document load 和 document ready 的区别

Document.onload 是在结构和样式加载完才执行 js

Document.ready 原生种没有这个方法，jquery 中有 `$.ready(function)`

20."==" 和 "===" 的不同

前者会自动转换类型

后者不会

21.javascript 的同源策略

一段脚本只能读取来自于同一起来源的窗口和文档的属性，这里的同一起来源指的是主机名、协议和端口号的组合

22.编写一个数组去重的方法

```
function oSort(arr)
{
    var result = {};
    var newArr=[];
    for(var i=0;i<arr.length;i++)
    {
        if(!result[arr])
```

```
{  
  newArr.push(arr)  
  result[arr]=1  
}  
}  
return newArr  
}
```