1. **js:**

```
function reverse(arr, len) { if (len > 1) {  for (let i = 0; i < len - 1; i++) {  let temp = arr[i];  arr[i] = arr[i + 1];  arr[i + 1] = temp;  }  len--;  reverse(arr, len); } return arr; }
```

```
function handleRes(str) { var arr = str.split(''); var len = arr.length; var answer = reverse(arr, len).join(''); return answer; }
```

2.
3. **js:**

   ```
   function merge(left, right) { var result = []; while (left.length > 0 && right.length > 0) {  if (left[0] < right[0]) {  result.push(left.shift());  } else {  result.push(right.shift());  } } return result.concat(left).concat(right); }
   ```

```
function mergeSort(arr) { if (arr.length == 1) {  return arr } var mid = Math.floor(arr.length / 2); var left_arr = arr.slice(0, mid),  right_arr = arr.slice(mid); return merge(mergeSort(left_arr), mergeSort(right_arr)); }
```

```
console.log(mergeSort([3, 1, 3, 4, 1, 5, 9, 2, 6]));
```

4. **js:**

   ```
   var arr = [{ "name": "张三", "serial": "0001" }, { "name": "李四", "serial": "0002" }, { "name": "王五", "serial": "0003" }, { "name": "王五2", "serial": "0003" }, { "name": "赵四", "serial": "0004" }, { "name": "小明", "serial": "005" }, { "name": "小张", "serial": "006" }, { "name": "小李", "serial": "006" }, { "name": "小李2", "serial": "006" }, { "name": "赵四2", "serial": "0004" }]; var result = [arr[0]]; for (var i = 1;i < arr.length;i++) { var flag = false; for(j = 0;j < result.length;j++) {  if (arr[i].serial === arr[j].serial) {  flag = true;  break;  } } if (!flag) {  result.push(arr[i]); } } console.log(result);
   ```

5. **js:**

   ```
   var s = [{  "id": "1",  "name": "中国",  "code": "110",  "parent": "" }, {  "id": "2",  "name": "北京市",  "code": "110000",  "parent": "110" }, {  "id": "3",  "name": "河北省",  "code": "130000",  "parent": "110" }, {  "id": "4",  "name": "四川省",  "code": "510000",  "parent": "110" }, {  "id": "5",  "name": "石家庄市",  "code": "130001",  "parent": "130000" }, {  "id": "6",  "name": "唐山市",  "code": "130002",  "parent": "130000" }, {  "id": "7",  "name": "邢台市",  "code": "130003",  "parent": "130000" }, {  "id": "8",  "name": "成都市",  "code": "510001",  "parent": "510000" }, {  "id": "9",  "name": "简阳市",  "code": "510002",  "parent": "510000" }, {  "id": "10",  "name": "武侯区",  "code": "51000101",  "parent": "510001" }, {  "id": "11",  "name": "金牛区",  "code": "51000102",  "parent": "510001" } ];
   ```

```
function tree(s) { var ind = 0; //判断第一层是不是还有子树 if (s.length > 1) {  for (var i = 0; i < s.length; i++) {  var a = 0; //计数信号量  for (var j = i + 1; j < s.length; j++) {  if (s[j].parent === s[i].code) { //判断是否有子树  a++; //子树计数  ind++;  }  }  if (a === 0 && s[i].parent !== '') { //没有子树，即树的最底层  for (var n in s) {  //定义children，避免undefined  s[n].children = s[n].children ? s[n].children : [];  if (s[n].code === s[i].parent) {  s[n].children.push(s[i]);  }  }  s.splice(i, 1); //删除，该子树已经加入了某项底层  i--; //删掉子树后后面的数据会填补空缺，退一步才能遍历完全  }  }  if (ind !== 0) { //如果还有子树继续遍历第一层  tree(s);  } } return s; }
```

```
console.log(tree(s));
```