

简则面试题

1. 总共有多少总排版，有穷无穷
2. 每一种怎么表示（数据结构如何）
3. 前端怎么基于第2步的表示，用html css画出如图中的排版样式呢

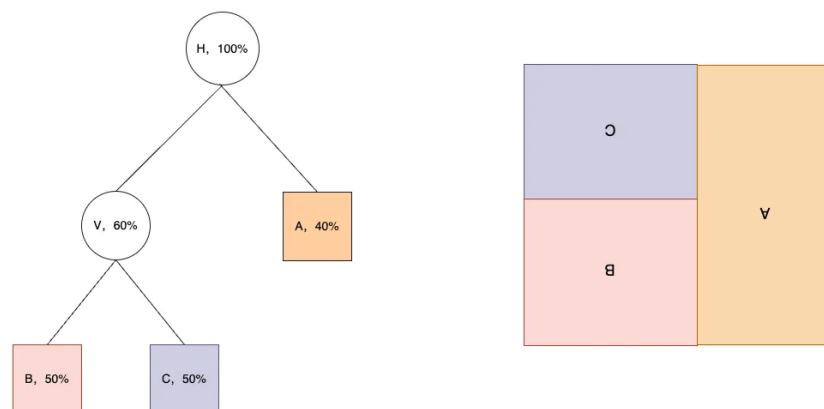
1. 总共有多少总排版，有穷无穷

有穷，总共有36种

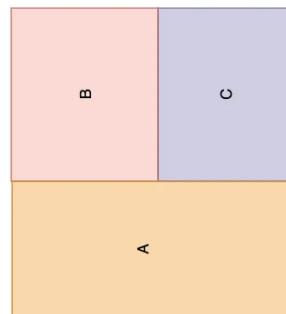
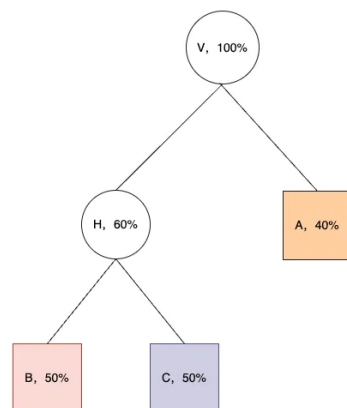
- 每种排版下，图片，文本，列表都有6种排列组合 $C_3^1 * C_2^1 = 6$
- 共有6种排版（下面列举）

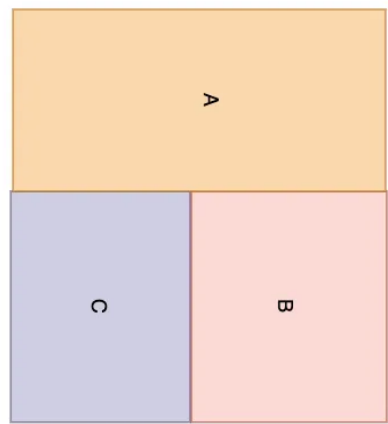
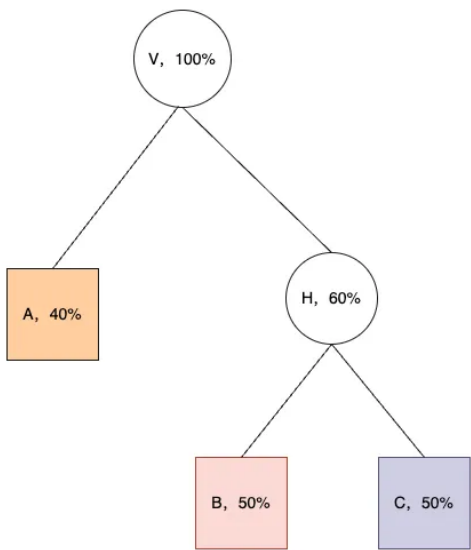
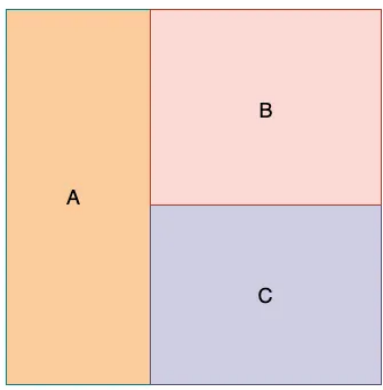
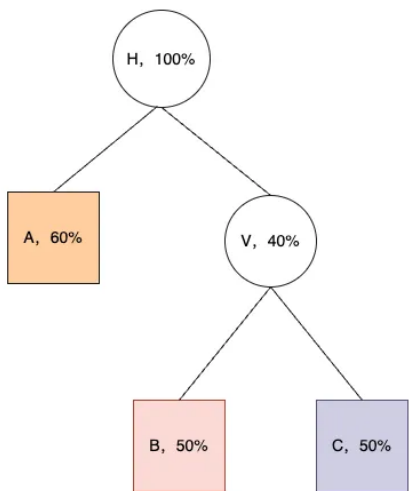
2. 每一种怎么表示（数据结构如何）

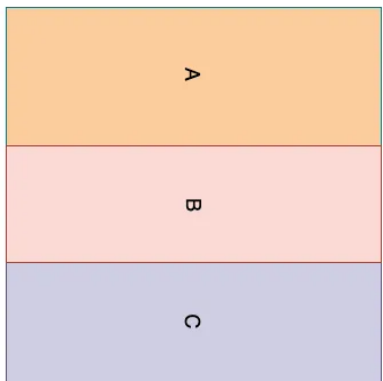
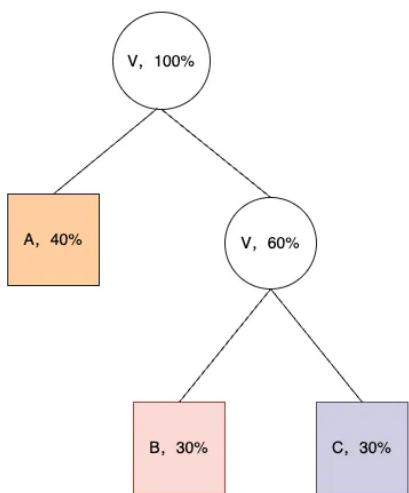
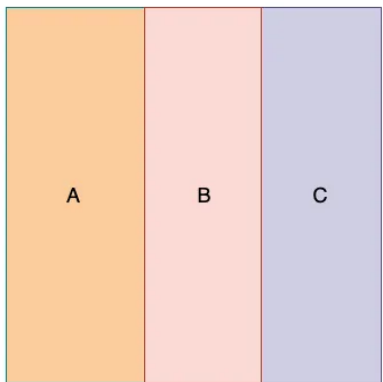
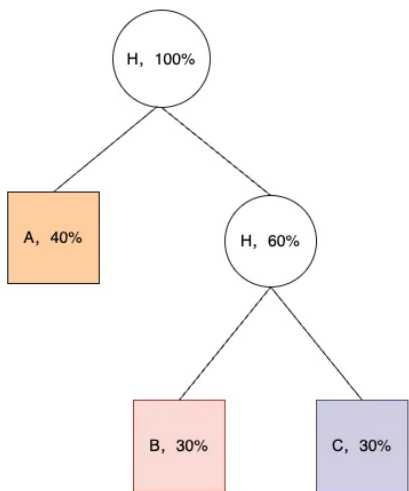
- 可以用一棵二叉树来表示
- 圆圈节点表示容器节点(container)，方块节点表示元素节点(文本，列表，图片)
- 容器节点确定布局方向



```
1  const tree = {  
2    type: 'workspace',  
3    orientation: 'horizontal',  
4    data: '100%',  
5    left: {  
6      type: 'container',  
7      data: '40%',  
8      orientation: 'vertical',  
9      left: {  
10       type: 'terminal',  
11       data: '40%',  
12     },  
13     right: {  
14       type: 'terminal',  
15       data: '60%',  
16     }  
17   },  
18   right: {  
19     type: 'terminal',  
20     data: '60%',  
21   }  
22 }  
23
```







3. 前端怎么基于第2步的表示，用html css画出如图中的排版样式呢

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8     <link rel="stylesheet" href="./index.css">
9 </head>
10
11 <body>
12     <div class="workspace horizontal">
13         <div class="container vertical" style="width: 40%">
14             <div class="terminal bg-red" style="height: 40%">40%</div>
15             <div class="terminal bg-yellow" style="height: 60%">60%</div>
16         </div>
17         <div class="terminal bg-blue" style="width: 60%">60%</div>
18     </div>
19     <script>
20         const tree = {
21             type: 'workspace',
22             orientation: 'horizontal',
23             data: '100%',
24             left: {
25                 type: 'container',
26                 data: '40%',
27                 orientation: 'vertical',
28                 left: {
29                     type: 'terminal',
30                     data: '40%',
31                 },
32                 right: {
33                     type: 'terminal',
34                     data: '60%',
35                 }
36             },
37             right: {
38                 type: 'terminal',
39                 data: '60%',
40             }
41         }
42
43         // 转化为html结构
44         // 递归函数：将树形结构转换为HTML节点
45         function createHTMLNode(node) {
```

```

46     const element = document.createElement('div');
47
48     if (node.type === 'workspace') {
49         element.classList.add('workspace');
50         element.classList.add(node.orientation);
51
52         const leftNode = createHTMLNode(node.left);
53         const rightNode = createHTMLNode(node.right);
54
55         element.appendChild(leftNode);
56         element.appendChild(rightNode);
57     } else if (node.type === 'container') {
58         element.classList.add('container');
59         element.classList.add(node.orientation);
60
61         const leftNode = createHTMLNode(node.left);
62         const rightNode = createHTMLNode(node.right);
63
64         element.appendChild(leftNode);
65         element.appendChild(rightNode);
66     } else if (node.type === 'terminal') {
67         element.classList.add('terminal');
68         element.textContent = node.data;
69     }
70
71     return element;
72 }
73
74 // // 获取根节点并将其添加到页面中
75 // const rootNode = createHTMLNode(tree);
76 // document.body.appendChild(rootNode);
77 </script>
78 </body>
79
80 </html>

```