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

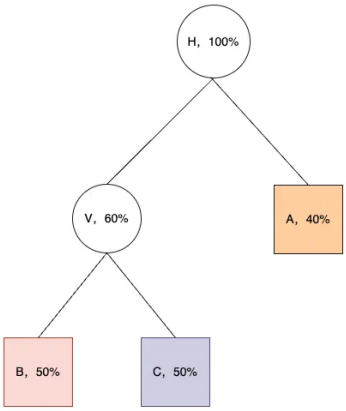
1. 总共有多少总排版，有穷无穷

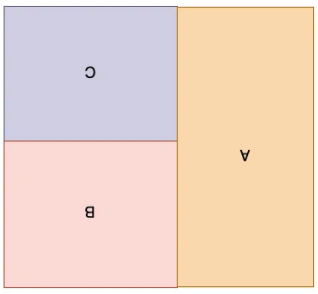
有穷，总共有36种

- 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）

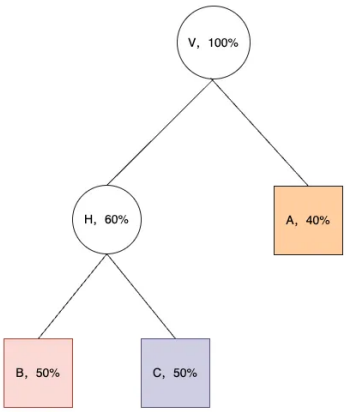
2. 每一种怎么表示（数据结构如何）

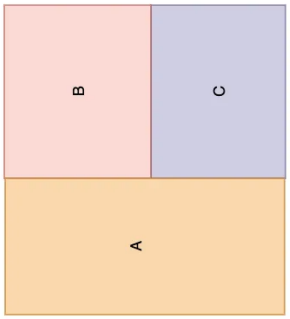
- 可以用一棵二叉树来表示
- 圆圈节点表示容器节点(container)，方块节点表示元素节点(文本，列表，图片)
- 容器节点确定布局方向

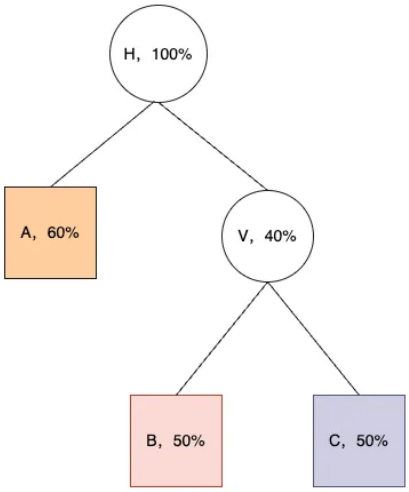


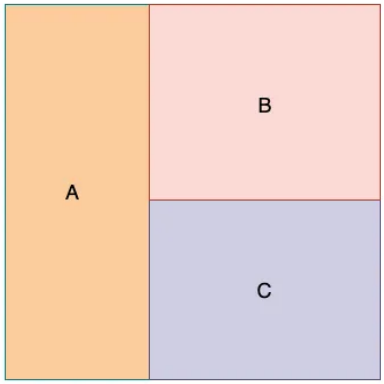


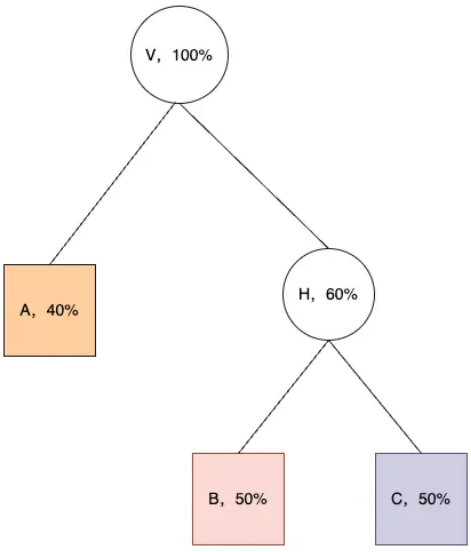
```javascript
const tree = {
    type: 'workspace',
    orientation: 'horizontal',
    data:'100%',
    left: {
        type: 'container',
        data: '40%',
        orientation: 'vertical',
        left: {
            type: 'terminal',
            data: '40%',
        },
        right: {
            type: 'terminal',
            data: '60%',
        }
    },
    right: {
        type: 'terminal',
        data: '60%',
    }
}

```

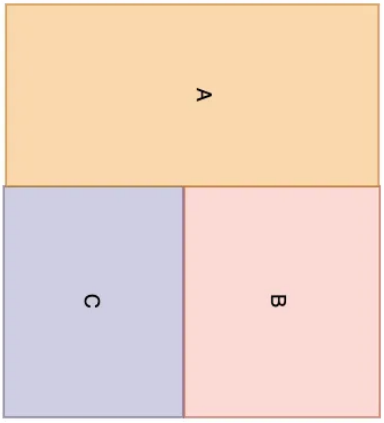


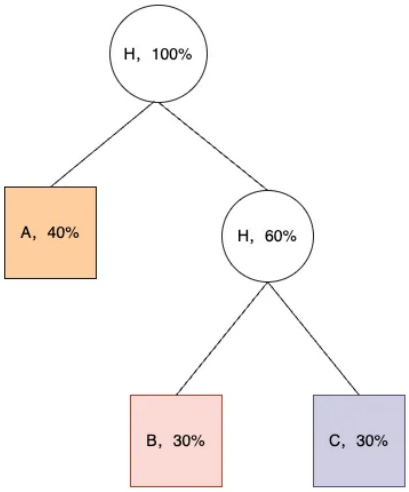


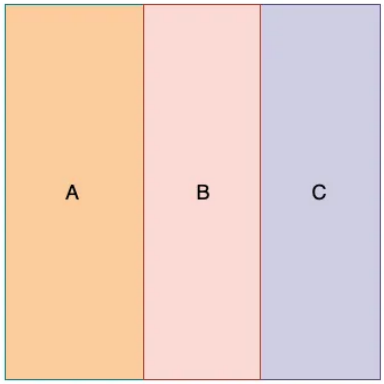

H, 100%
A, 60%
V, 40%
B, 50%
C, 50%

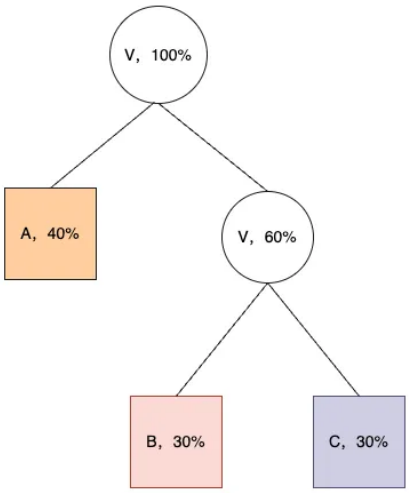

A
B
C

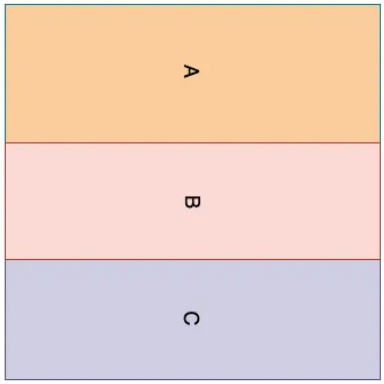

V, 100%
A, 40%
H, 60%
B, 50%
C, 50%


A
C
B


H, 100%
A, 40%
H, 60%
B, 30%
C, 30%


A
B
C


V, 100%
A, 40%
V, 60%
B, 30%
C, 30%


A
B
C

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

HTML

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="./index.css">
</head>

<body>
    <div class="workspace horizontal">
        <div class="container vertical" style="width: 40%">
            <div class="terminal bg-red" style="height: 40%">40%</div>
            <div class="terminal bg-yellow" style="height: 60%">60%</div>
        </div>
        <div class="terminal bg-blue" style="width: 60%">60%</div>
    </div>
    <script>
        const tree = {
            type: 'workspace',
            orientation: 'horizontal',
            data:'100%',
            left: {
                type: 'container',
                data: '40%',
                orientation: 'vertical',
                left: {
                    type: 'terminal',
                    data: '40%',
                },
                right: {
                    type: 'terminal',
                    data: '60%',
                }
            },
            right: {
                type: 'terminal',
                data: '60%',
            }
        }

        // 转化为html结构
        // 递归函数：将树形结构转换为HTML节点
        function createHTMLNode(node) {
```

```
            const element = document.createElement('div');

            if (node.type === 'workspace') {
                element.classList.add('workspace');
                element.classList.add(node.orientation);

                const leftNode = createHTMLNode(node.left);
                const rightNode = createHTMLNode(node.right);

                element.appendChild(leftNode);
                element.appendChild(rightNode);
            } else if (node.type === 'container') {
                element.classList.add('container');
                element.classList.add(node.orientation);

                const leftNode = createHTMLNode(node.left);
                const rightNode = createHTMLNode(node.right);

                element.appendChild(leftNode);
                element.appendChild(rightNode);
            } else if (node.type === 'terminal') {
                element.classList.add('terminal');
                element.textContent = node.data;
            }

            return element;
        }

        // // 获取根节点并将其添加到页面中
        // const rootNode = createHTMLNode(tree);
        // document.body.appendChild(rootNode);
    </script>
</body>

</html>
```