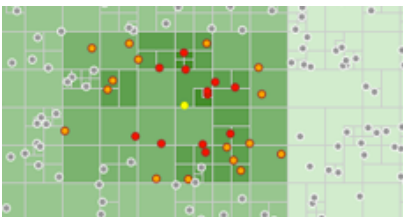
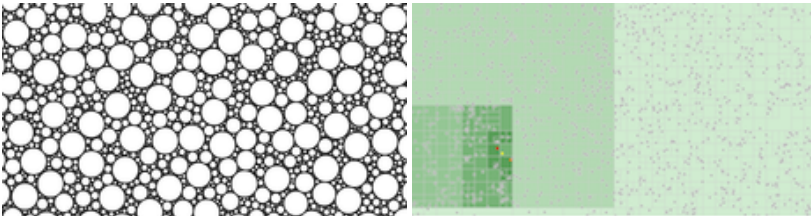
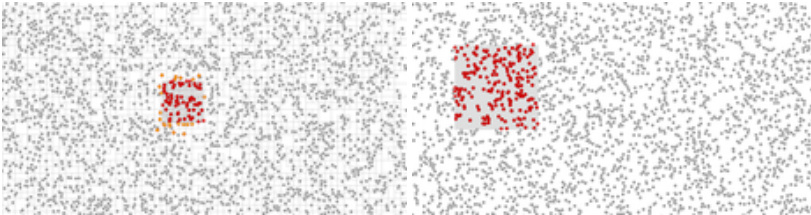


[Wiki](#) ▸ [\[\[API--中文手册\]\]](#) ▸ [\[\[几何\]\]](#) ▸ **四叉树**

- 如发现翻译不当或有其他问题可以通过以下方式联系译者:
- 邮箱: zhang_tianxu@sina.com
- QQ群: [D3数据可视化](#)205076374, [大数据可视化](#)436442115

四叉树是一个二维空间递归细分。它使用了平分划分实现，将每一个块平分成了四个同等大小的方块。每个点存在于一个唯一的节点中；如果同一位置包含多个点，那么这多个点中的其中一些点将存储于内部节点，而非叶子结点。四叉树可用于加速各种空间操作，例如计算正多边形的体积的Barnes-Hut近似算法或者冲突检验。



`d3.geom.quadtree()`

创建一个新的四叉树工厂使用默认的x访问器([x-accessor](#))，y访问器([y-accessor](#))及范围([extent](#))。[返回的函数](#)可以用来从带有工厂的配置的数据创建任意个四叉树。

`quadtree(points)`,

`quadtree(points, x2, y2)`,

`quadtree(points, x1, y1, x2, y2)`

为指定的点数据数组`points`构造一个新的四叉树，返回新四叉树的根结点。每一个点的`_x`和`_y`坐标使用当前[x](#)-和[y](#)-访问器函数确定。通过增量地添加点来建立四叉树，指定的点数组可以为空，之后点可以随后添加([added](#))到返回的根节点；在这种情况下，你必须指定四叉树的范围([extent](#))。

四叉树的每一个结点都有多个属性。

- `nodes` - 按顺序排列四个子结点的稀疏数组：top-left, top-right, bottom-left, bottom-right。
- `leaf` - 内部结点与叶子结点的布尔表示。
- `point` - 这个点关联的节点，如果有的话（可能适用于内部或叶节点）。
- `x` - 关联点的`_x`坐标，如果有的话。
- `y` - 关联点的`_y`坐标，如果有的话。

返回的根结点也可定义了增加方法([add](#))或者访问方法([visit](#))。

`# root.add(point)`

为四叉树增加指定的新点`point`。

`# root.visit(callback)`

访问四叉树的每个结点，并为每个利用参数结点调用指定的回调函数`callback`，入参是 `{node, x1, y1, x2, y2}`。其中`node`是被访问过的节点，剩下的参数分别是节点左上角和右下角的坐标（注意：四叉树中使用的坐标系定义是任意的，所以更精确的规则是 $x1 \leq x2$ 以及 $y1 \leq y2$ 。SVG和Canvas中使用的坐标系的原点(0,0)是左上角，同理 $(x1, y1)$ 也是当前节点的左上角。）。结点按先序遍历。若调用给定结点的回调函数返回值为`true`，则表示此结点的子结点未被访问，否则表示所有子结点均被访问过。

`# root.find(point)`

给定任意点`*[x,y]*`，返回在四叉树中最近的点。

quadtree.x([x])

若设置了`x`参数，则为四叉树设置横坐标访问器，并返回四叉树。反之，则返回当前默认的横坐标访问器。

```
function(d) { return d[0]; }
```

对每一个加入到四叉树中的点，不管是初始化构造的([initial construction](#))还是后面新增的([added](#))，都可以通过参数`d, i`调用`x`访问器，`d`表示当前点，`i`表示所有点数组的索引。`x`访问器必须返回一个数值，表明给定点的`x`坐标。如果需要的话，`x`访问器也可以被定义为一个常数，而非一个函数。

`# quadtree.y([y])`

若设置了`y`参数，则为四叉树设置纵坐标访问器，并返回四叉树。反之，则返回当前默认的纵坐标访问器

```
function(d) { return d[1]; }
```

对每一个加入到四叉树中的点，不管是初始化构造的([initial construction](#))还是后面新增的([added](#))，都可以通过参数 `{d, i}`调用`y`访问器，`d`表示当前点，`i`表示所有点数组的索引。`y`访问器必须返回一个数值，表明给定点的`y`坐标。如果需要的话，`y`访问器也可以被定义为一个常数，而非一个函数。

`# quadtree.extent([extent])`

若设置了`extent`参数，则为四叉树设置范围，然后返回四叉树。反之，则返回当前范围，其默认为空。当范围为空时，范围将会自动扫描输入点数组自动计算并传到四叉树构造器([quadtree constructor](#))。否则，范围必须用二维数组`[[x0, y0], [x1, y1]]`明确定义，`x0`和`y0`为范围的下限，`x1`和`y1`为范围的上限。当从最初为空的节点慢慢构建一个四叉树，设置范围是非常必要的。

本文参与	人员	组织	时间
翻译	曼妙征程	VisualCrew小组	20141127
校对	大傻	VisualCrew小组	2014-12-8 20:24:31
排版	liang42hao	VisualCrew小组	2016-3-16 22:28