

## 3.2.1 Adjacency Matrix

Main Principle:

- Node list as rows and columns of table
- item Cell shows presence/absence of edge
- item Edge between nodes can be weighted or have attributes

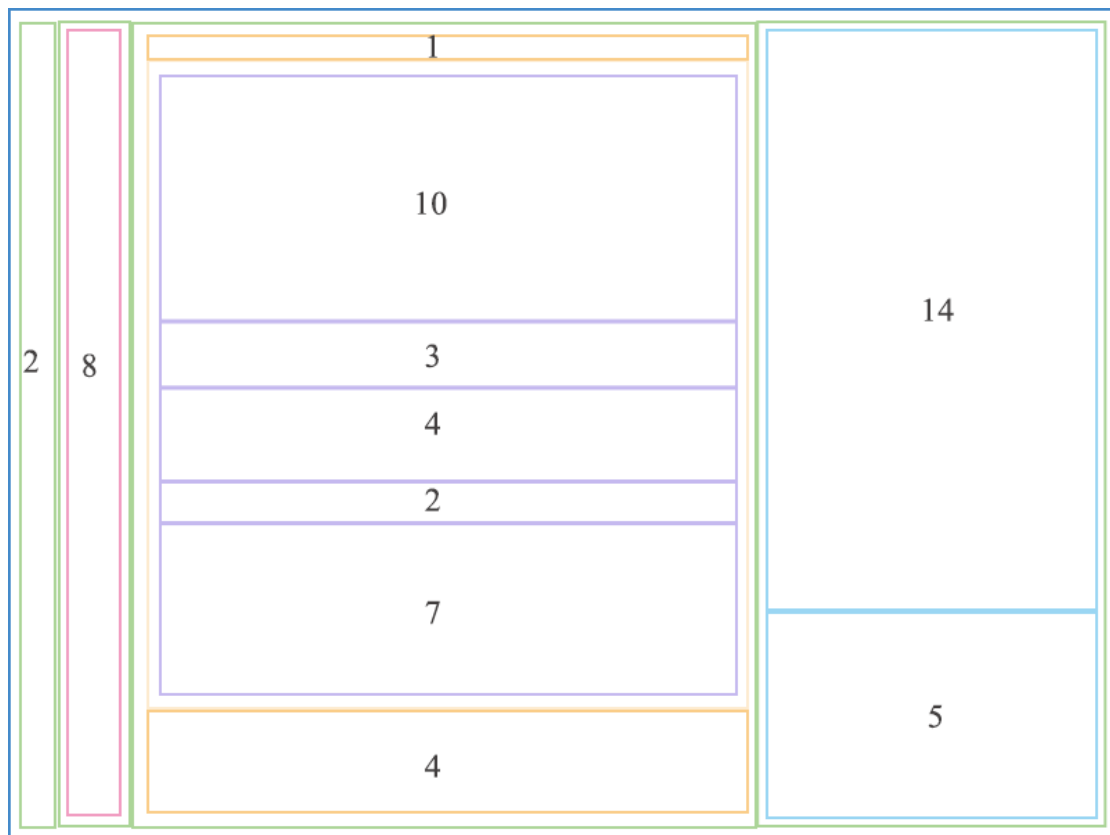
Matrix

$$\begin{bmatrix} 0 & 0 & 5 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \end{bmatrix}$$

Visible characterist: }

- adjacency matrix: predictability, scalability, supports reordering, topology tasks trainalbe
- item node-link graphic: topology understanding, path tracing, intuitive

### 3.2.2 TreeMap



Start with root, the rectangle will be separated into four partitions. The first node is leaf, so that the value of the first partition is 2. The second node has a child as leaf with value 8. The third node has three children, therefore the third partition will be separated again into three (yellow) parts. On the same way, the middle yellow part has five sections because of its subtrees. The fourth node has two children with value 14 and 5.