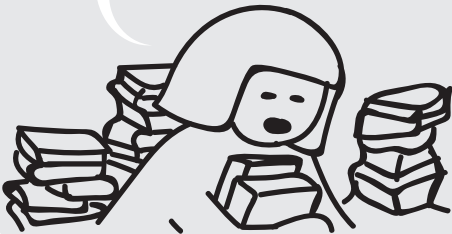


How is organising the library book collection going?



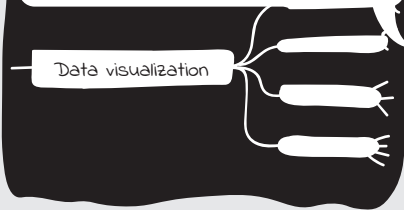
I'm running out of space quickly! There are so many genres, subgenres, and topics, the entire blackboard is now all but covered!



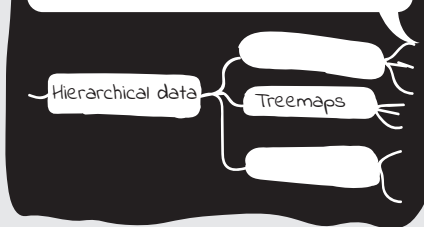
Hmmm...



Maybe we should have a look at the library books on data visualization?



For example, treemaps sound pretty interesting.



Not quite sure how knowing the way to the campus oaks is gonna help us.

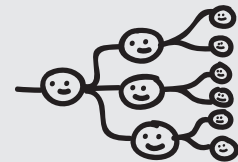
No, no, that doesn't seem what they are about.



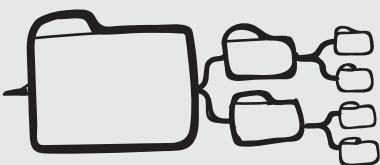
Treemaps is a way of visualizing hierarchical data with **part-whole** relationships...



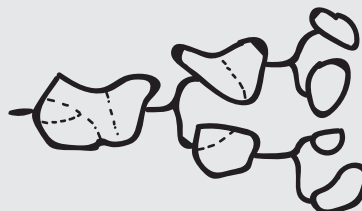
Hierarchical data is, as you know, organized into tree-like levels, but not all trees are part-whole. For example, children in a family tree aren't part of their parents.



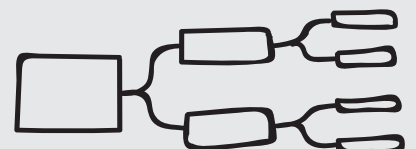
But sub-directories in a file system are part of the directory they are in...



... geographic areas are part of larger regions and provinces...

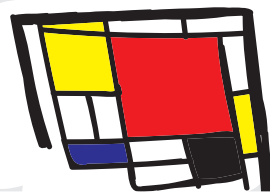
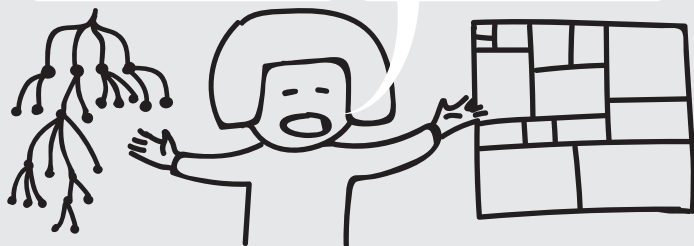


... and as are categorical systems like that of library book organization.



However, instead of representing them using nodes and links...

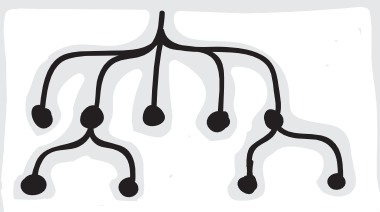
... they represent it through rectangles nested inside bigger ones.



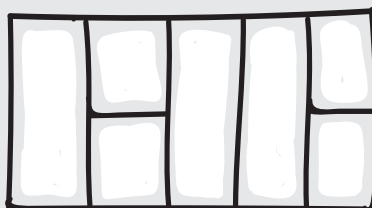
Again, not sure what modern artworks have to do with this...



Think about it! Node-and-link diagrams have tons of unused whitespace.



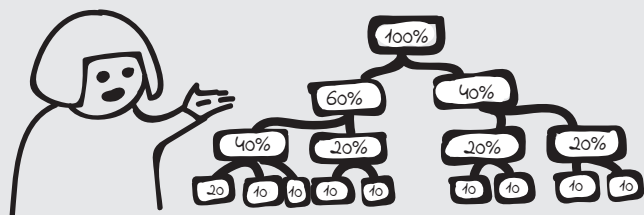
With nested hierarchies you can take the same amount of space...



... but if you need to add more elements you can just subdivide the same old space as before.



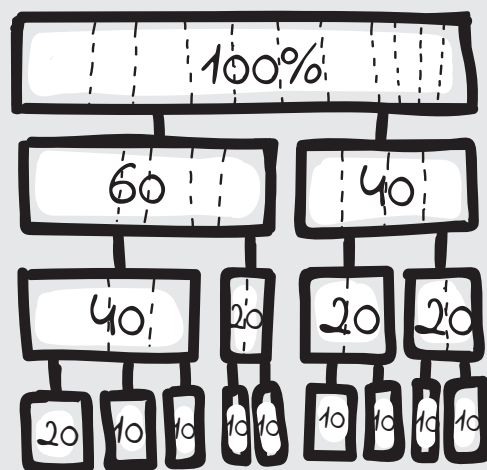
What's more, node-and-link diagrams have no convenient way of conveying the size or proportion that a particular node constitutes.



With nested hierarchies, you can simply let the size of the area symbolise the size of the node it represents.



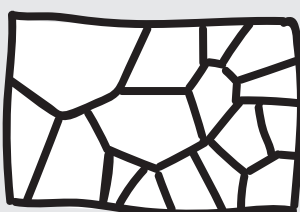
This means that all nodes at each level will add up to the same grand total. Values of sub-branches add up to the value of the branch as a whole.



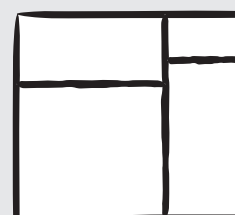
In principle you could choose to nest circles instead.



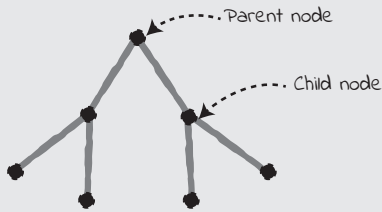
Or even irregular polygons.



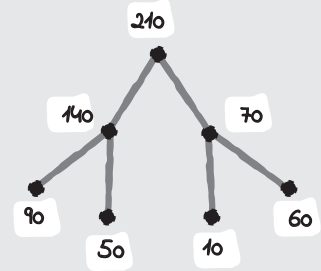
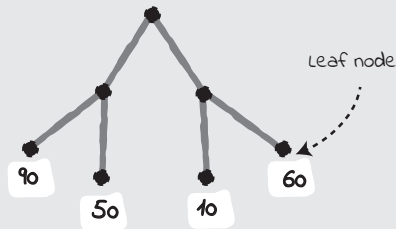
Rectangles is the most efficient and simple to read, however.



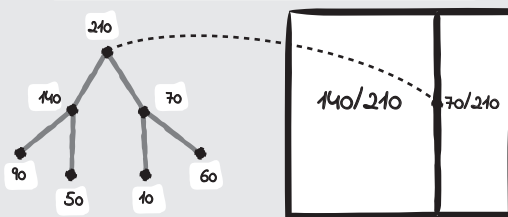
The algorithm accepts a tree data structure:



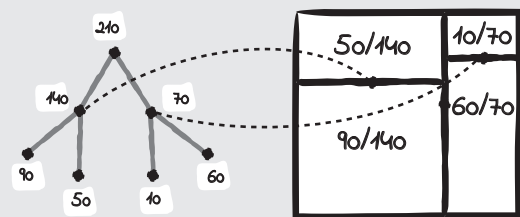
Each leaf node has a value attached to it so that the value can be computed for each node in a bottom-up manner.



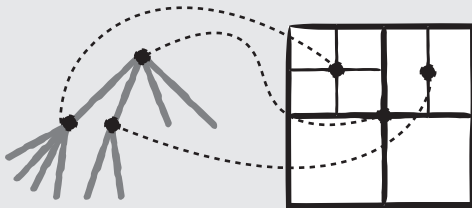
The algorithm then proceeds top-down to split the area according to the size proportions of the first level of nodes.



It then does the same for the next level, but switches to horizontal subdivisions.



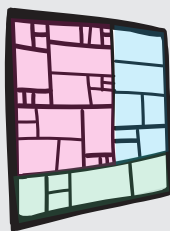
If the nodes do not have any particular size attached, then the rectangles of each level will be made to be of equal size.



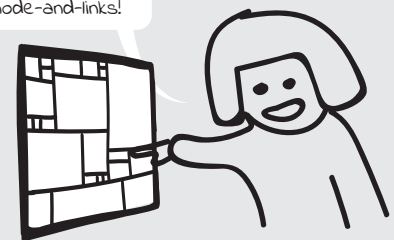
At some point the subdivisions become impossible to see, but if you're able to zoom in on areas that's not a problem!



Besides, you can use hues to distinguish main categories and intensity to highlight depth.



which again is what makes your new library visualization so compact compared with the node-and-links!



Are you...?



Are you tiling the room with the treemap?



well, I had to do something with all that leftover space!

