**QUADTREE**

A quadtree is a data structure that is commonly used for image processing and computer graphics. It can be used to represent an image by recursively subdividing it into smaller regions, called quadrants.

The process of representing an image using a quadtree typically involves the following steps:

* Divide the image into four quadrants, each representing a smaller region of the image.
* Check if each quadrant is homogeneous, meaning that it contains pixels of the same color or intensity. If a quadrant is homogeneous, it is represented by a single pixel of that color or intensity.
* If a quadrant is not homogeneous, it is further subdivided into four smaller quadrants, and the process is repeated recursively until all the quadrants are homogeneous.
* The resulting quadtree is a tree data structure, where each node represents a quadrant of the image. The root node represents the entire image, and its children represent the four quadrants into which it was initially divided.
* The leaves of the quadtree represent the smallest homogeneous regions of the image, which can be represented by a single pixel.

To reconstruct the original image from the quadtree representation, one can simply traverse the tree in a depth-first manner, concatenating the pixels represented by each leaf node to form the complete image.

Application: Image representation and compression. Geolocation Service like nearest locations(s).

**ORDERBOOK**

Stock exchanges use **order books**. An order book is an electronic list of buy and sell orders, organized by price levels. It has a buy book and a sell book, where each side of the book contains a bunch of price levels, and each price level contains a list of orders (first in first out).  
The diagram below is an example of price levels and the queued quantity at each price level.



So what happens when you place a market order to buy 2700 shares in the diagram?

1. The buy order is matched with all the sell orders at price 100.10, and the first order at price 100.11 (illustrated in light red).
2. Now because of the big buy order which “eats up” the first price level on the sell book, the best ask price goes up from 100.10 to 100.11.
3. So when the market is bullish, people tend to buy stocks aggressively, and the price goes up and up.

An efficient data structure for an order book must satisfy:

* Constant lookup time. Operations include: get volume at a price level or between price levels, query best bid/ask.
* Fast add/cancel/execute/update operations, preferably O(1) time complexity. Operations include: place a new order, cancel an order, and match an order.