A Day of Puzzles

Practice Solving Programming Puzzles

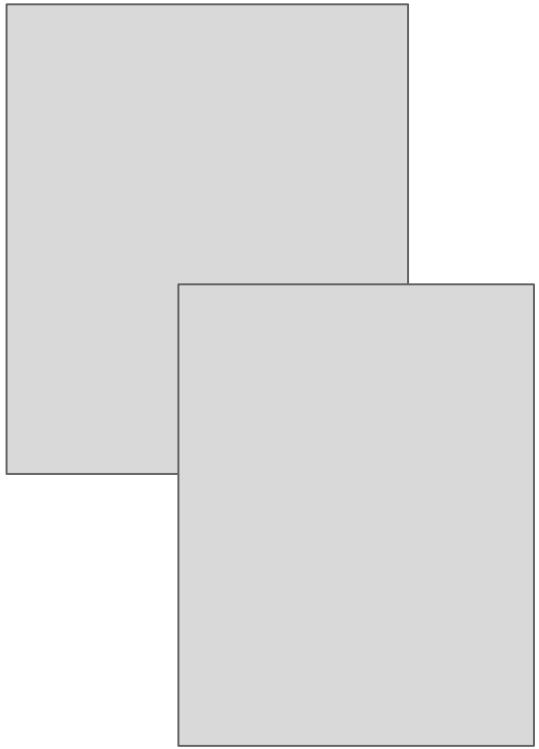
Intersecting Boxes

Consider two rectangular boxes, R_1 and R_2 in two dimensions. The boxes are aligned along the horizontal axis.

- Determine of the boxes intersect
- Find the rectangular intersection

A rectangle is defined by:

```
Rectange (double lowerLeftX,  double lowerLeftY,
width, height)
```



Spiral Ordering of a 2D-Matrix

Traverse a matrix of numbers in a spiral order, starting with the top-left element, move right all the way, then move down all the way, then move left all the way, then move up all the way but stop short of visiting the visited elements. Turn right again, and so forth, till all the elements have been visited.

- Do it clockwise
- Do it anticlockwise
- Do it from center outwards in increasing spiral radius

Laterally Symmetric Binary Tree

Test if a binary tree is symmetric: i.e., its left subtree is a mirror i laterally, of the right subtree.

Hints:

- It is easier than you think
- Think recursion

Specific cost path in a binary tree

Let each node of a binary tree be an integer.

Find a path to a node in the tree, if it exists, such that the sum of values of the visited nodes adds up to a specific cost (value).

Write a function that looks like this:

```
List<Integer> findPath(Node treeRoot, int cost)
```

Now, write a variant that finds all such paths.

k-nearest stars in the galaxy

This is the classic problem in interviews:

Given a list of n stars whose coordinates you know, say of a galaxy. How would you find the k-nearest stars (say the 10-closest stars) to a given star.

Exact median of online data

Suppose you are being given data one by one (i.e. in streaming fashion).

How would you find the median of the data at any given point of time?

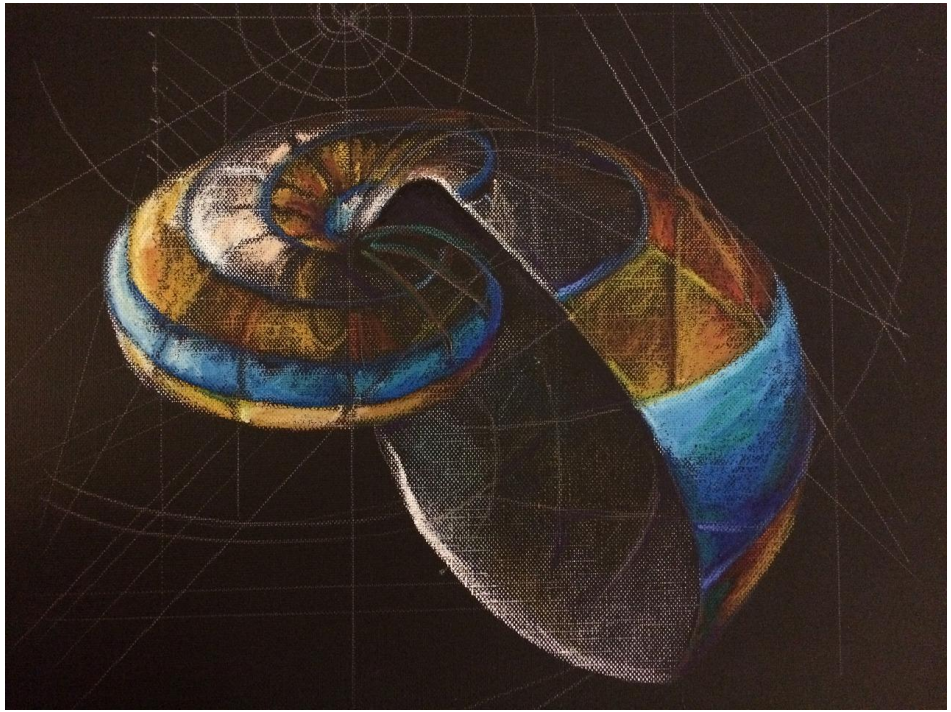
Fibonacci without recursion

Write a function that computes the fibonacci number without using recursion.

Compare its performance to recursive solution for values of 1, 10, 100, 1000

Now, write a recursive fibonacci function, but use a Least-Recently Used Cache (LRUCache). How does it help?

(For python users, look also at functools module)



Find the existing majority

Assume that a large list of different numbers exist, but that more than half of the numbers is one specific number. (If it helps, think different kinds of fruits instead).

Write a function that finds the majority element in the list in O(n) time.

Find the k-smallest and k-largest

How would you find the k-smallest and k-largest elements in a large list of numbers that you are allowed to traverse only once.

Very heavy statues

Imagine that in a part, N very heavy statues are in an unsorted order, along a line.

How would you sort the statues in the line, with the minimum amount of back-breaking work (heavy lifting movements)?