Lesson 7 Closest-pair algorithm

# Learning objectives

1. Gain an intuition for divide-and-conquer algorithms
2. Notice the similarities between the closest-pair algorithm and merge sort
3. Explain at a basic level how the closest-pair algorithm works
4. Develop skill at reading and following mathematical arguments and explanations that are presented in print-form (as opposed to interactive form)
5. Prove the Fibonacci Tree theorem.

# Agenda

1. Proof of the Fibonacci Tree Theorem
2. Intro to the Closest-Pair problem
3. Brute force algorithm runs in quadratic time
4. A divide-and-conquer approach
   1. Basic idea
   2. The key observation about the bounding box
   3. Main result
   4. Read 3 different presentations of the algorithm and pick the clearest one
5. Independent work
   1. Guessing game
   2. Assignment 5
   3. Koch snow flake or Sierpinski carpet
   4. Thinking about final project

## Proof of Fibonacci Tree theorem

**Theorem**

Let n = the generation of growth in a Fibonacci tree described by the rules above.

Let t(n) = number of branches in the nth generation

Then the number of branches in a Fibonacci tree is given by the Fibonacci sequence. That is,

t(1) = 1

t(2) = 1

t(n) = t(n-1) + t(n-2), for n > 2, n in **Z**+

**Proof**

Let b(n) = # of baby branches in generation n

k(n) = # of kid branches in generation n

a(n) = # of adult branches in generation n

By definition,

t(n) = a(n) + k(n) + b(n)

Observe that

b(n) = a(n-1) + k(n-1)

k(n) = b(n-1)

a(n) = a(n-1) + k(n-1)

Substituting

So t(n) = a(n-1) + k(n-1) + b(n-1) + a(n-1) + k(n-1)

= t(n-1) + a(n-1) + k(n-1)

= t(n-1) + a(n-2) + k(n-2) + b(n-2)

= t(n-1) + t(n-2)

Further, t(1) = 1 and t(2) = 1, so we have proved the theorem.