Recursion

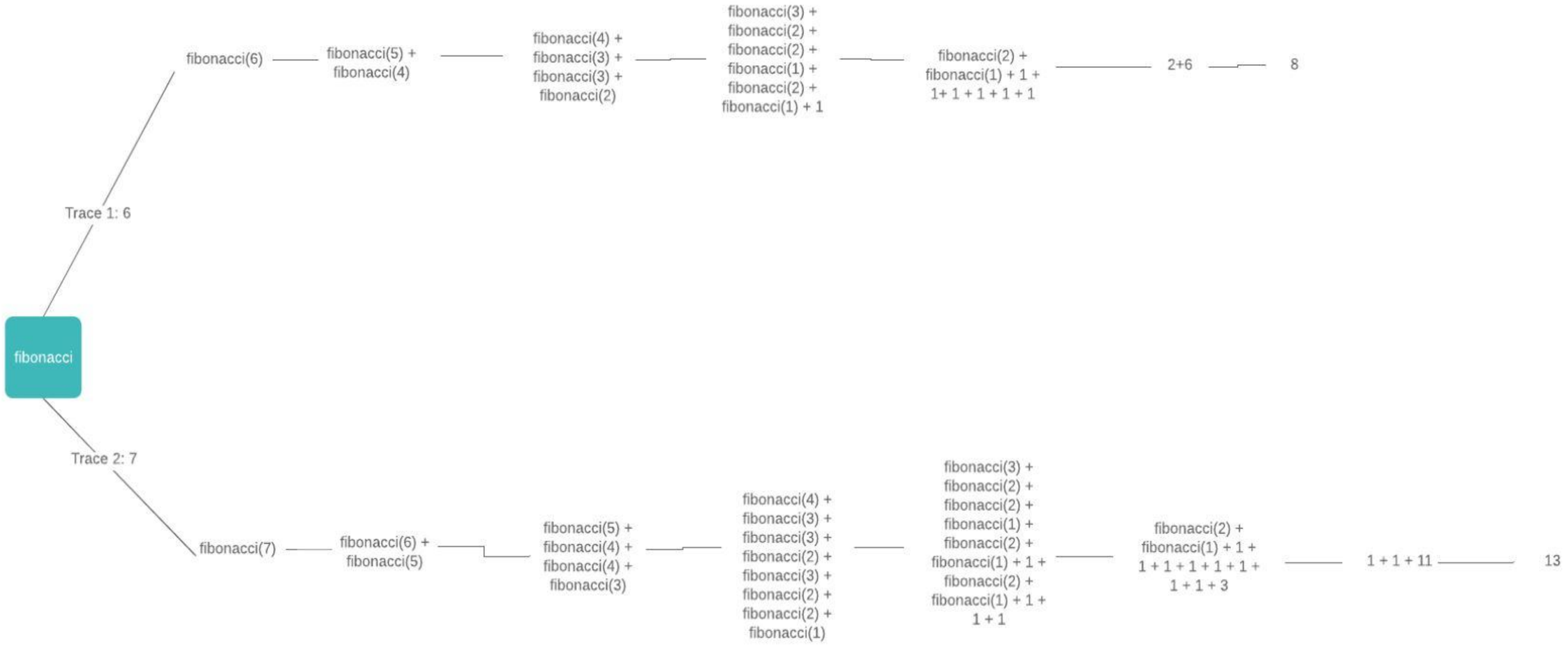
Jerry Zhu

What is Recursion and When to Apply it?

- Recursion is a problem-solving method whereby the method calls itself or other methods.
- Typically, a certain condition must be met for a recursive call to be made.
- A recursive method runs until it reaches the base case, at which point a specific value is returned, and the recursive calls are traced backwards with the determined value.
- Recursion is the most effective at solving tree-structures, namely repetitive actions. It keeps simplying a complex problem until it gets to the base case where it solves the problem.

How does it Work?

- The trace for a recursive method calculating the Fibonacci Sequence:



What are Some Types of Recursion?

- **Linear recursion** - when a method only makes one call to itself each time it runs.
  - For example, the Factorial method uses this.
    - **Tail recursion** - a type of linear recursion, where the recursive call is the last thing the method does - can often be implemented iteratively
      - For example, finding the GCD given two numbers.
- **Mutual recursion** - a form of recursion where multiple methods recursively call each other.
  - For example, the Fibonnacci method uses this.

Pros VS Cons

- If applied properly, recursion is a problem-solving method that simplifies many complications that would have otherwise been forced to implement.
  - For example, a recursive call may subsitute several loops and conditionals.

- A limitation to recursion is the possibility of that it can take up much space as each call adds to the stack on which the computer must constantly create more storage space to process the data.
- A recursive method can be very inefficient if the magnitude to which the method must recurse is big.

Tips for Success

1. Never forget your base call. Without it, infinite recursive calls happen and a StackOverFlow Error is thrown.
2. Test your recursive method through printing the values used out. This could be your variables, your return methods, your loop indexes, etc.
3. Trace your recursion by hand to understand, on a profound level, how it works so that you can write the method more effectively.

Personal Reflection

- At the beginning, recursion stumped me, because this unit was about applying recursion, not just tracing basic methods. With the latter, I improved with more practice. But with the former, I practiced, but it still remained to be one of the most unintuitive concepts I have ever learned to apply.
- Consequently, I had a hard time with the assignments, especially given that you had to complete them in the minimal number of lines possible.
- How I overcame this challenge was through taking the time to sit down and think about the problem. I took the time to trace through my methods and it helped me visualize the problems and eventually solve them.