```java
class Fibber {
    public static int FibFun(int x) {
        if (x == 0) {
            return 0;
        } else if (x == 1) {
            return 1;
        }
        return FibFun(x - 1) + FibFun(x - 2);
    }

    public static void main(String a[]) {
        System.out.println(FibFun(4));
    }
}
```

Suppose `FibFun(3)` is called. The blue circled numbers reflect what happens in order of time.

**LineNumberTable:**
- Line 4: 0
- Line 6: 4
- Line 8: 6
- Line 10: 11
- Line 12: 13

**LocalVariableTable:**
- Start Length Slot Name Signature
  - 0 27 0 x I

**Code:**
- `aload_0` 0
- `ifne_6` 1
- `ireturn` 5
- `iload_0` 6
- `iconst_0` 7
- `if_icmpne_2` 8
- `iconst_1` 11
- `ireturn` 12

Some JVM memories are in a stack:
- `if` (x == 0) if not go to 6
- `if` true when x = 3: [stack]
  - 1. Tests if 3 == 0.
  - 2. Pops the stack
  - 3. Make the JVM
    - go to the instruction
    - starting at 6:

First Recursive Call:
- FibFun(3)
- Returns 2 (the return value is left at the top of the stack)
- return FibFun(x-1) + FibFun(x-2)

Second Recursive Call:
- FibFun(2)
- return 1

- FibFun(1)
- return 1

- FibFun(0)
- return 0