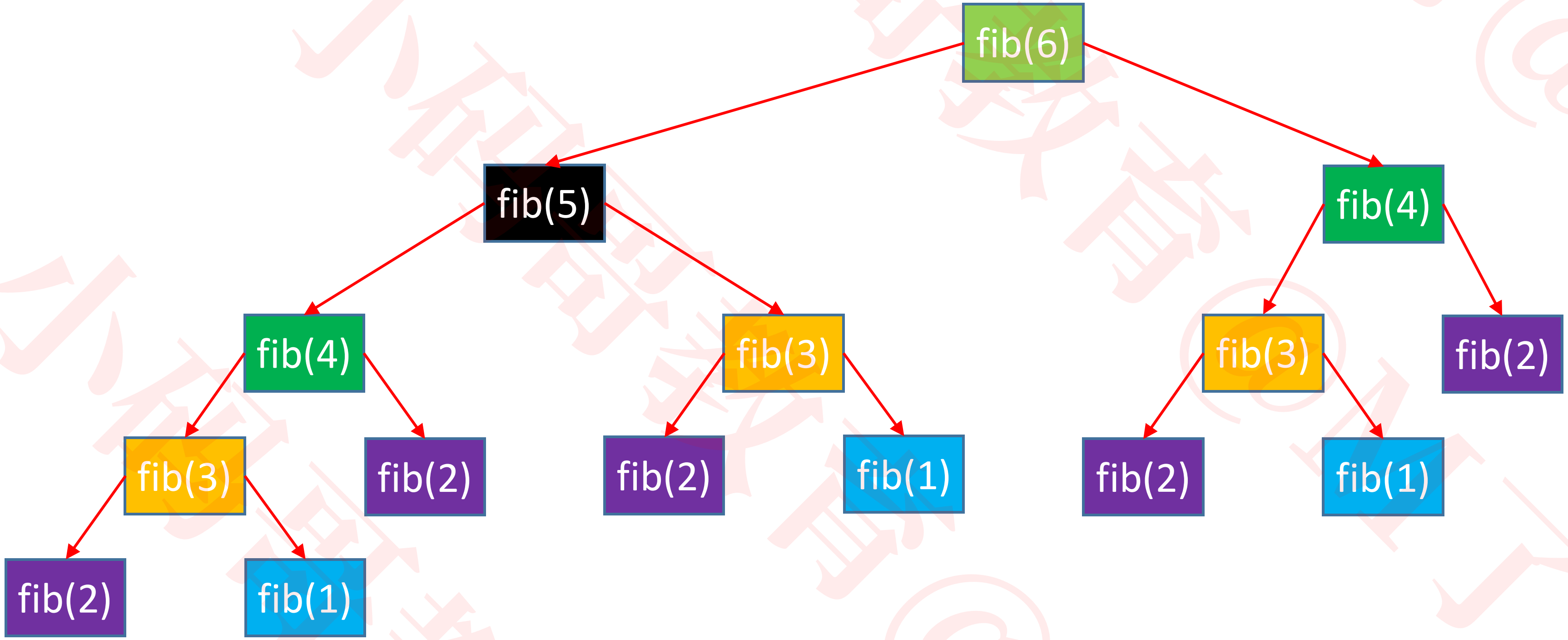
练习1 – 斐波那契数列

- 斐波那契数列：1、1、2、3、5、8、13、21、34、……
  - F(1)=1，F(2)=1, F(n)=F(n-1)+F(n-2)（n≥3）
- 编写一个函数求第 n 项斐波那契数

```
int fib(int n) {
    if (n <= 2) return 1;
    return fib(n - 1) + fib(n - 2);
}
```

- 根据递推式 $T(n) = T(n-1) + T(n-2) + O(1)$，可得知时间复杂度：$O(2^n)$
- 空间复杂度：$O(n)$
  - 递归调用的空间复杂度 = 递归深度 * 每次调用所需的辅助空间

fib函数的调用过程



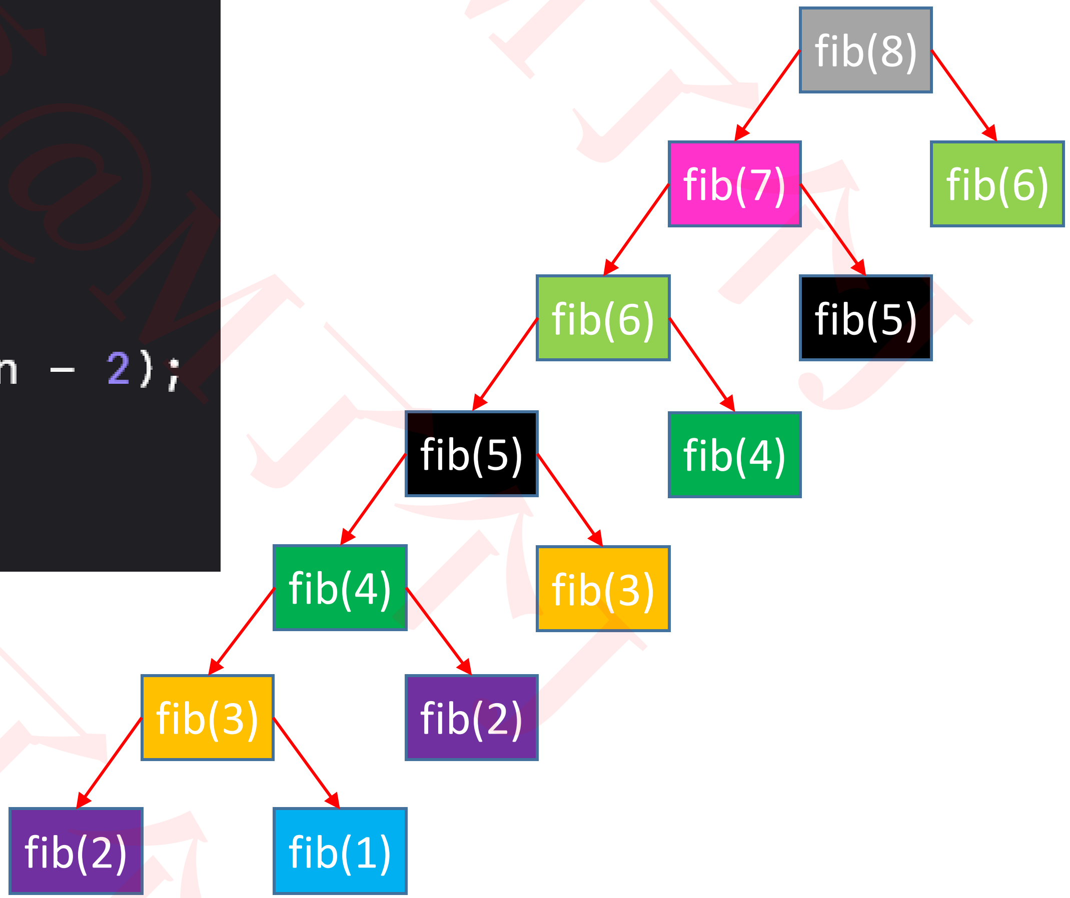
- 出现了特别多的重复计算
- 这是一种“自顶向下”的调用过程

fib优化1 – 记忆化

- 用数组存放计算过的结果，避免重复计算

```
int fib(int n) {
    if(n <= 2) return 1;
    int[] array = new int[n + 1];
    array[2] = array[1] = 1;
    return fib(array, n);
}

int fib(int[] array, int n) {
    if (array[n] == 0) {
        array[n] = fib(array, n - 1) + fib(array, n - 2);
    }
    return array[n];
}
```



- 时间复杂度：O(n)，空间复杂度：O(n)

fib优化2

- 去除递归调用

```
int fib(int n) {
    if (n <= 2) return 1;
    int[] array = new int[n + 1];
    array[2] = array[1] = 1;
    for (int i = 3; i <= n; i++) {
        array[i] = array[i - 1] + array[i - 2];
    }
    return array[n];
}
```

- 时间复杂度：O(n)，空间复杂度：O(n)
- 这是一种“自底向上”的计算过程

fib优化3

- 由于每次运算只需要用到数组中的 2 个元素，所以可以使用**滚动数组**来优化

```
int fib(int n) {
    if (n <= 2) return 1;
    int[] array = new int[2];
    array[0] = array[1] = 1;
    for (int i = 3; i <= n; i++) {
        array[i % 2] = array[(i - 1) % 2] + array[(i - 2) % 2];
    }
    return array[n % 2];
}
```

- 时间复杂度：O(n)，空间复杂度：O(1)

fib优化4 – 位运算取代模运算

- 乘、除、模运算效率较低，建议用其他方式取代

```
int fib(int n) {
    if (n <= 2) return 1;
    int[] array = new int[2];
    array[0] = array[1] = 1;
    for (int i = 3; i <= n; i++) {
        array[i & 1] = array[(i - 1) & 1] + array[(i - 2) & 1];
    }
    return array[n & 1];
}
```

fib优化5

```
int fib(int n) {
    if (n <= 2) return 1;
    int first = 1;
    int second = 1;
    for (int i = 3; i <= n; i++) {
        second = first + second;
        first = second - first;
    }
    return second;
}
```

- 时间复杂度：O(n)，空间复杂度：O(1)

fib优化6

- 斐波那契数列有个线性代数解法：特征方程

$$F(n) = c_1 x_1^n + c_2 x_2^n. \qquad x_1 = \frac{1+\sqrt{5}}{2}, x_2 = \frac{1-\sqrt{5}}{2}. \qquad c_1 = \frac{1}{\sqrt{5}}, c_2 = -\frac{1}{\sqrt{5}}.$$

$$F(n) = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right].$$

```
int fib(int n) {
    double c = Math.sqrt(5);
    return (int)((Math.pow((1 + c) / 2, n) - Math.pow((1 - c) / 2, n)) / c);
}
```

- 时间复杂度、空间复杂度取决于 pow 函数（至少可以低至O(logn)）