Recursion Tree and code relation

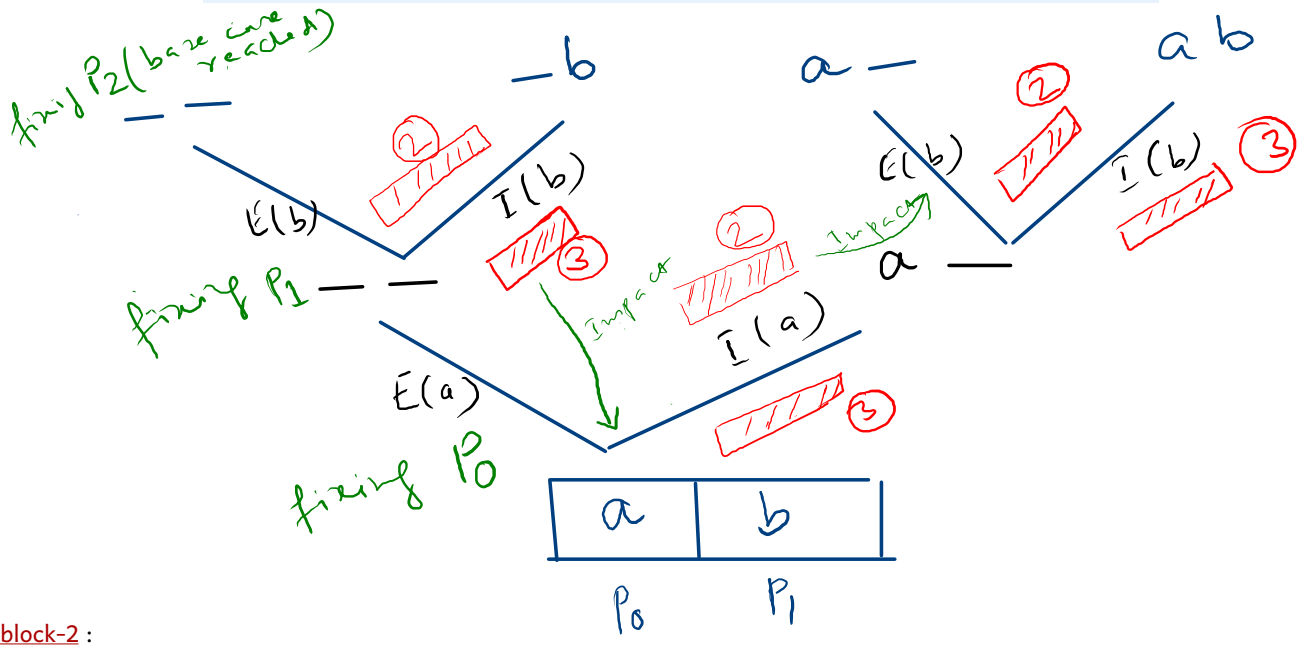
```java
public static void printPowerSetByFixingPosition(char[] input, char[] output, int pos) {
    if (pos == output.length) {
        System.out.println(Arrays.toString(output));
        return;
    }
    // excluding p(i)
    printPowerSetByFixingPosition(input, output, pos + 1);

    output[pos] = input[pos];

    printPowerSetByFixingPosition(input, output, pos + 1);

    output[pos] = Character.MIN_VALUE;

}
```



Note:

1. code-block-2 :

-- Will be executed as **post** of **exclude-recursion** and **pre** of **include-recursion** at sibling branch which is obvious as per the code placement.

-- Will be executed **before** **pre** of **exclude-recursion** while transiting to **next-level** from **include-recursion,** which is not obvious naturally, but can be visualized through the Euler Tour.

2. code-block-3 :

-- Will be executed as **post** of **include-recursion** which is obvious as per the code placement .

-- Will be exeucuted **before** (**post** of **exclude-recursion** and **pre** of **include-recursion)** while recursion falls back to previous level from include-recursion to exclude-recursion, which is not obvious naturally, but can be visualized through Euler Tour.