

```
public static int compute(String s) {  
    if (s.equals("0")){  
        return 0;  
    }  
    if (s.equals(s("1"))){  
        return 1;  
    }  
    if (s.charAt(s.length()-1) == '0'){  
        return 2 * compute(s.substring(0,s.length()-1));  
    }  
    if (s.charAt(s.length()-1) == '1'){  
        return 1 + 2 * compute(s.substring(0,s.length()-1));  
    }  
    return -1;  
}
```

The image displays a Java method named `compute` that takes a `String s` as input and returns an `int`. The code is annotated with a dense network graph. The graph consists of numerous nodes, represented by colored circles (purple, orange, yellow, red, and grey), which are interconnected by lines of the same colors. These nodes and edges are positioned over the code, often highlighting specific tokens or expressions. For example, a large cluster of purple nodes is centered around the `compute` method signature and its recursive calls. A horizontal line of orange and red nodes spans across the `return 2 * compute(s.substring(0,s.length()-1));` line. The graph appears to represent a control flow graph or a data flow graph, illustrating the relationships and dependencies between different parts of the program's execution.