

```
public static int compute(int a, int b) {  
    int result = a * b;  
    for(int i = 1; i < a * b; i++){  
        if(i % a == 0 || i % b == 0){  
            result += 1;  
            break;  
        }  
    }  
    return result;  
}
```

The image displays a network graph overlaid on a Java code snippet. The graph consists of numerous nodes, represented by colored circles (purple, orange, yellow, and grey), and edges, represented by thin lines connecting these nodes. The nodes are distributed across the code, with a high concentration in the loop body. The edges represent relationships between different parts of the code, such as variable references, control flow, and data dependencies. The code itself is a function named 'compute' that takes two integers 'a' and 'b' as input. It calculates the product 'a * b' and stores it in 'result'. It then enters a 'for' loop that iterates from 1 to 'a * b - 1'. Inside the loop, it checks if the current value 'i' is divisible by either 'a' or 'b'. If so, it increments 'result' by 1 and breaks out of the loop. Finally, it returns the value of 'result'.