Internet Fundamentals

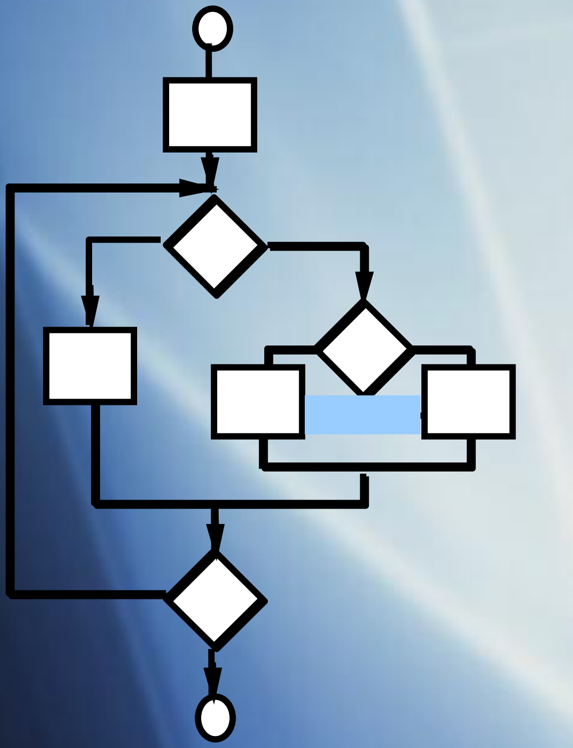
Lecture-31

Cyclomatic Complexity

What is it?

- A software metric used to measure the complexity of software
- Developed by Thomas McCabe
- Described (informally) as the number of decision points + 1

Cyclomatic Complexity V(G)



Computing the cyclomatic complexity:

number of simple decisions + 1

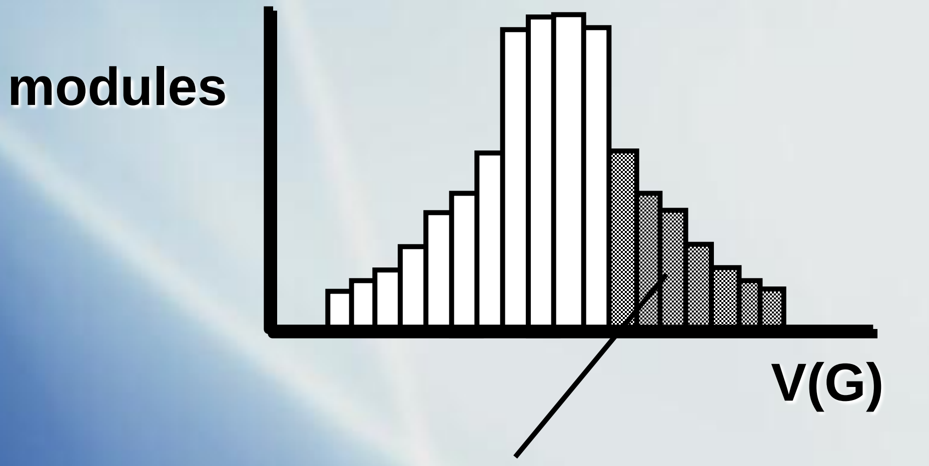
or

number of enclosed areas + 1

In this case, V(G) = 4

Graph Complexity (Cyclomatic Complexity)

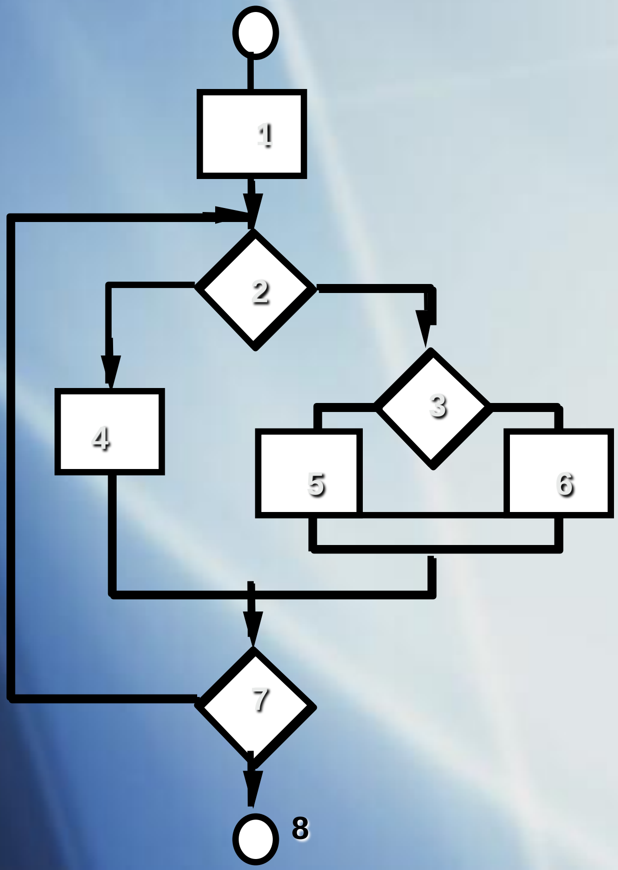
A number of industry studies have indicated that the higher V(G), the higher the probability or errors.



modules in this range are more error prone

From Pressman Slides - Software Engineering a Practical Approach 6,e

Basis Path Testing



Next, we derive the independent paths:

Since V(G) = 4, there are four paths

Path 1: 1,2,3,6,7,8
Path 2: 1,2,3,5,7,8
Path 3: 1,2,4,7,8
Path 4: 1,2,4,7,2,4,...7,8

Finally, we derive test cases to exercise these paths.

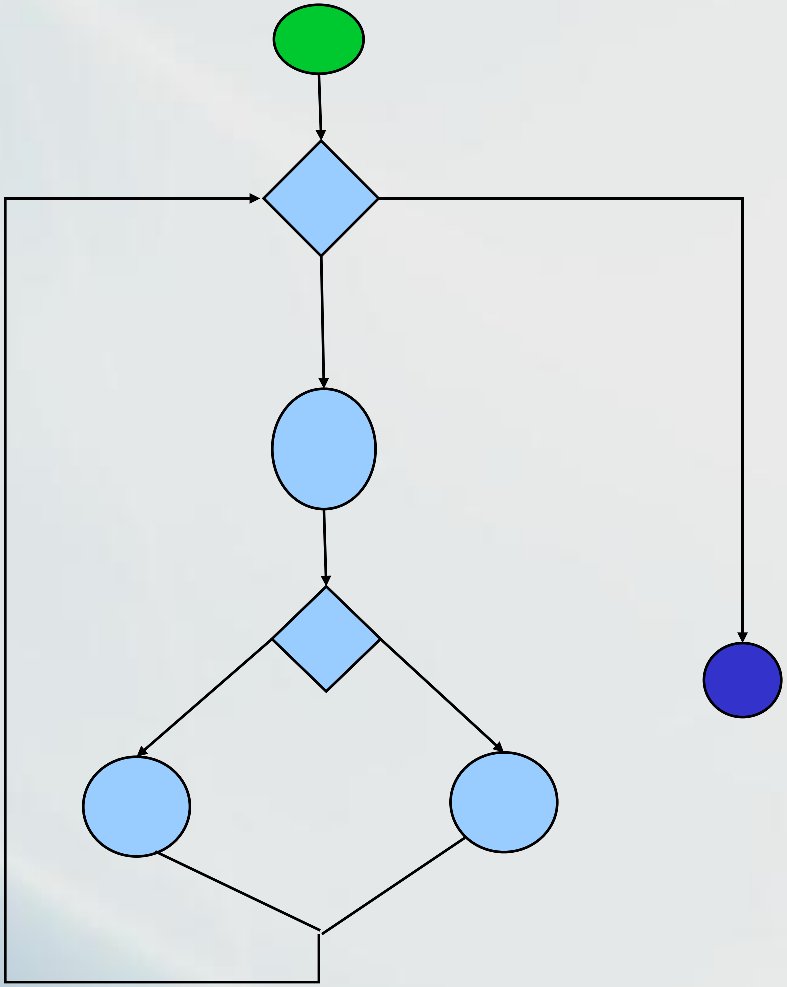
What is the complexity?

```
public void howComplex() {
    int i=20;

    while (i<10) {
      System.out.printf("i is %d", i);
      if (i%2 == 0) {
        System.out.println("even");
      } else {
        System.out.println("odd");
      }
    }
  }
```

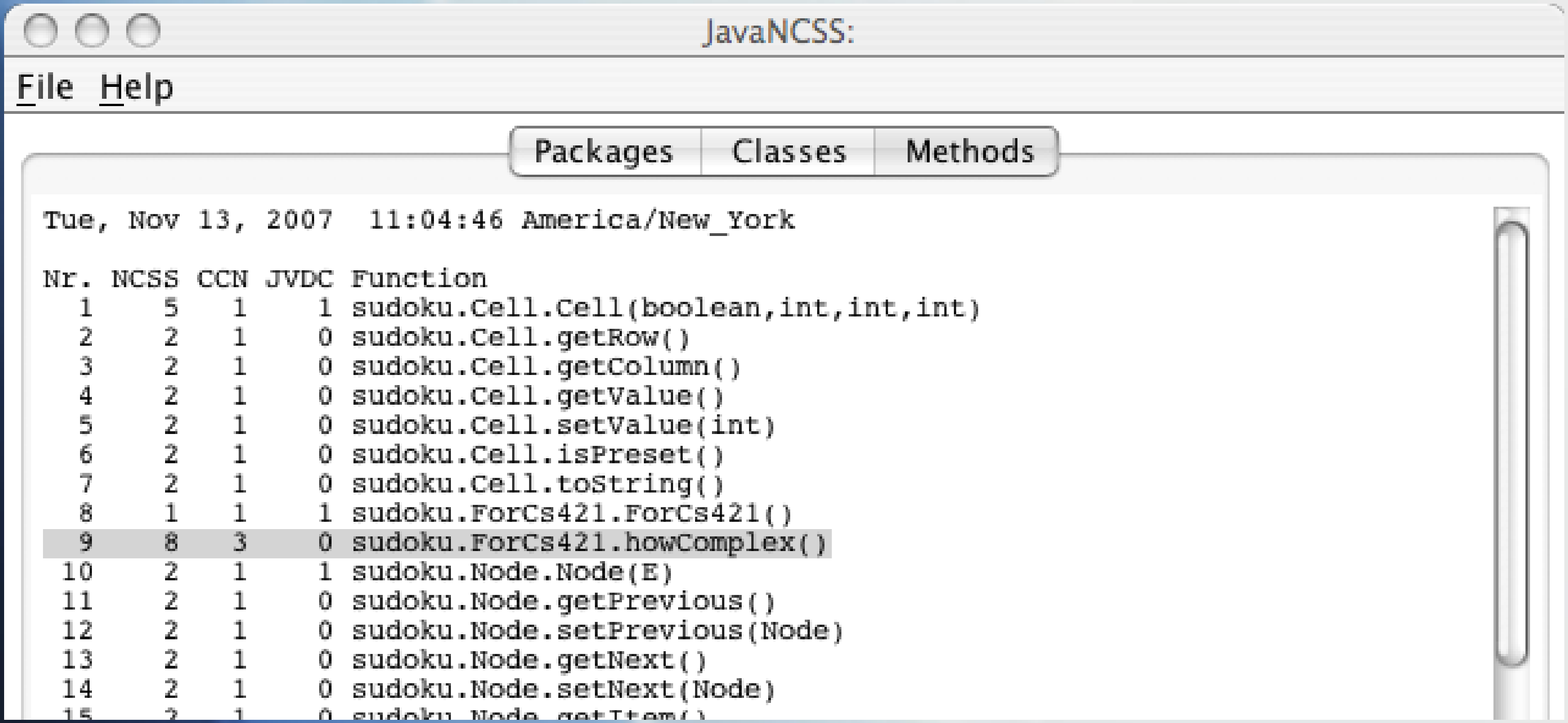
What is the complexity V(G)?

```
public void howComplex() {
      int i=20;

      while (i<10) {
         System.out.printf("i is %d", i);
         if (i%2 == 0) {
            System.out.println("even");
         } else {
            System.out.println("odd");
         }
      }
   }
```

V(G) = 2 enclosed area + 1 = 3



Output from JavaNCSS



```
Tue, Nov 13, 2007  11:04:46 America/New_York

Nr. NCSS CCN JVDC Function
  1    5   1    1 sudoku.Cell.Cell(boolean,int,int,int)
  2    2   1    0 sudoku.Cell.getRow()
  3    2   1    0 sudoku.Cell.getColumn()
  4    2   1    0 sudoku.Cell.getValue()
  5    2   1    0 sudoku.Cell.setValue(int)
  6    2   1    0 sudoku.Cell.isPreset()
  7    2   1    0 sudoku.Cell.toString()
  8    1   1    1 sudoku.ForCs421.ForCs421()
  9    8   3    0 sudoku.ForCs421.howComplex()
 10    2   1    1 sudoku.Node.Node(E)
 11    2   1    0 sudoku.Node.getPrevious()
 12    2   1    0 sudoku.Node.setPrevious(Node)
 13    2   1    0 sudoku.Node.getNext()
 14    2   1    0 sudoku.Node.setNext(Node)
 15    2   1    0 sudoku.Node.getItem()
```