Introduction to Cyclomatic Complexity

Cyclomatic Complexity

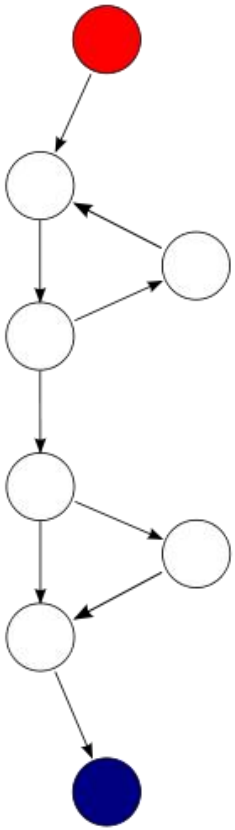
- It was developed by Thomas J. McCabe in 1976 and is used to indicate the complexity of a program.
- CC or Complexity (M)=E-N+2P

  where

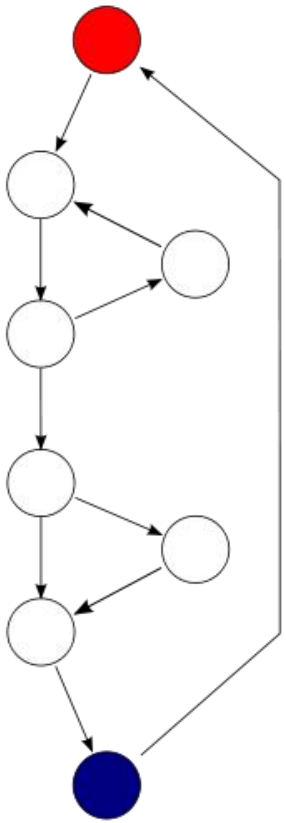
  E = the number of edges of the graph

  N = the number of nodes of the graph

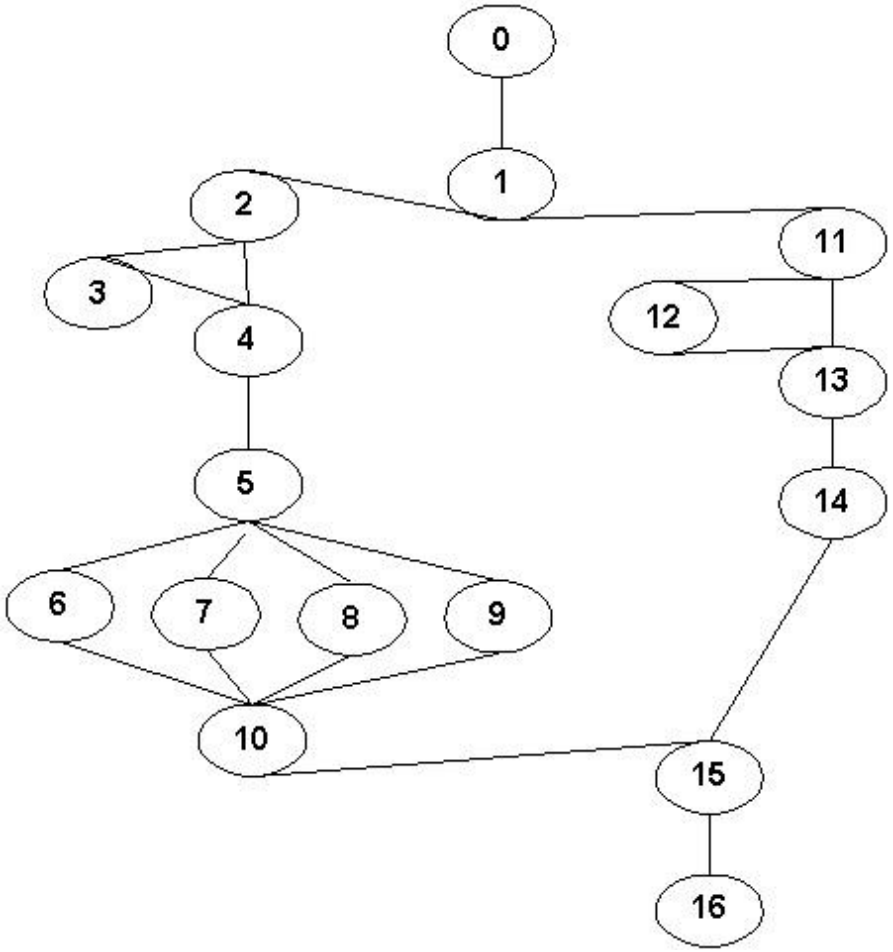
  P = the number of exit



For this graph, E = 9, N = 8 and P = 1, so the cyclomatic complexity of the program is 9 - 8 +(2*1) = 3.



The same function as above, shown as a *strongly connected* control flow graph. For this graph, E = 10, N = 8 and P = 1, so the cyclomatic complexity of the program is 10 - 8 + 1 = 3.


0
1
2
11
3
12
4
13
5
14
6
7
8
9
10
15
16

The answer is:

- CC=7 (22E-17N+2P)

Cyclomatic Complexity

Other considerations:

- Start with 1. There is at least one path through the method.
- Add 1 for each conditional statement or looping statement. (So, add 1 for “if”, “while”, “foreach”)
- Add 1 for each AND or OR logical operator used in a condition.
- Add 1 for each case or switch statement.
- Add 1 for each catch statement.

Example 1

```
public void
  isValidSearchCriteria(SearchCriteria s){
       if(s!=null) {
         return true;
      }else{
         return false;
      }
}
```

The answer is:

- CC=1+1=2 *(one if +1)*

Example 2

```
public WaitableThread(ThreadStart start) {
   this.start = start;
   this.signal = new ManualResetEvent(false);
   this.SafeWaitHandle = signal.SafeWaitHandle;
   this.thread = new Thread(new
   ThreadStart(ExecuteDelegate));
}
```

The answer is:

- CC=1 (No conditionals, loops, or anything else of interest.)

Example 3

```
 foreach (LockFreeQueue q in queueList) {
    if (q.Dequeue(out item)) {
    return true;
    }
   }
   item = default(T);
   return false; }
}
```

The answer is:

- CC=3 (Count 1 for the main path, 1 for the foreach, and 1 for the if.)

Example 4

```
public void findApplications(String id, String name){
  if(id!=null && name!=null) {
      //do something
   }else{
      //do something
   }
}
```

The answer is:

- CC=3 (Count 1 for the main path, 1 for the normal *if-else* decision points and one for the logical *AND* operation.)

Example 5

```
if A=354
   then if B>C
   then A=B
   else A=C
   endif
   endif
print A
```

The answer is:

- CC=3 (There are two decision points)

Example 6

```
   int i, int j  {
if ( i>0 && j>0 ) {
while ( i>j ) {
if ( i% 2 && j % 2)
print( "%d\ n", i );
else
print ("%d\ n", j );
i--;
}
}
}
```

The answer is:

- CC=6 (Starting with 1, two "if" statements, one "while" statement,and two "&&" operators.)

Example 7

```
int i {
    if (n>o) {
      switch(n) {
        case 0: case1: printf( “zero or one\n” );
        break;
        case2: printf( “two\n” );
        break;
        case3: case4: printf( “three or four\n” );
        break;
      }
    } else {
        printf ( “negative\n” );
    }
```

The answer is:

- CC=5 (Starting with 1, one "if" statement, the switch statement has three "case-labeled statements".)

Cyclomatic Complexity	**Risk Evaluation**
1-10	a simple program, without much risk
11-20	more complex, moderate risk
21-50	complex, high risk program
greater than 50	untestable program (very high risk)