SNS COLLEGE OF TECHNOLOGY

Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SOFTWARE ENGINEERING

(Agile UX/UI)

UNIT 5-Software Testing

Topics

- What is Basis Path Testing
- How to create Control Flow Graph
- What is Cyclomatic Complexity
- How to create Independent Paths

Software Testing

A more appropriate definition is:

"Testing is the process of executing a program with the intent of finding errors."

Test Case Design Techniques

- Functional Testing(Black Box Testing)

- Structural Testing(White Box Testing)

Test Case Design Techniques

- Functional Testing(Black Box Testing)
  - Equivalence Partitioning
  - Boundary Value Analysis
  - Comparison Testing
- Structural Testing(White Box Testing)
  - Basis Path Testing
  - Control Structure Testing

Basis Path Testing

For Deriving Test Cases

Step1: Using the design or code, draw the corresponding flow graph.

Step 2: Draw the DD Path Graph

Step 3 :Determine the cyclomatic complexity of the flow graph.

Step 4: Determine a basis set of independent paths.

Step5: Prepare test cases that will force execution of each path in the basis set.

Control Flow Graph

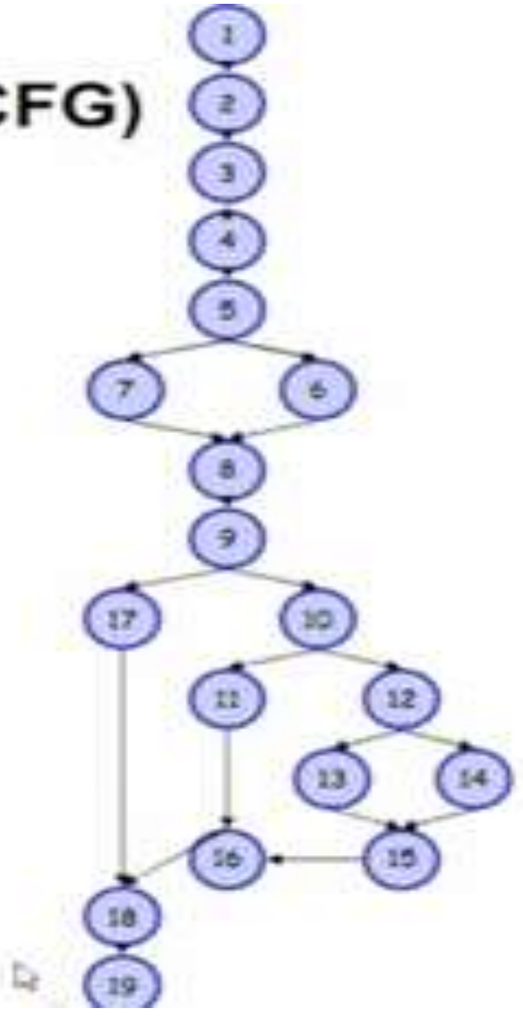
- The control flow of a program can be analyzed using a graphical representation known as flow graph. The flow graph is a directed graph in which node represents statement , and edges represents flow of control.
- In other words , a control flow graph describes how the control flows through the programs.

Control Flow Graph (CFG)

```
1   program TRIANGLE
2   input (a)
3   input (b)
4   input (c)
5   if (a<b+c) AND (b<a+c) AND (c<a+b)
6      then IsATriangle = T
7      else IsATriangle = F
8   endif
9   if IsATriangle
10     then if (a=b) AND (b=c)
11        then Output = "Equilateral"
12        else if (a != b) AND (b != c) AND (a != c)
13              then Output = "Scalene"
14              else Output = "Isosceles"
15        endif
16     endif
17  else Output = "Not a triangle"
18  endif
19  end TRIANGLE
```

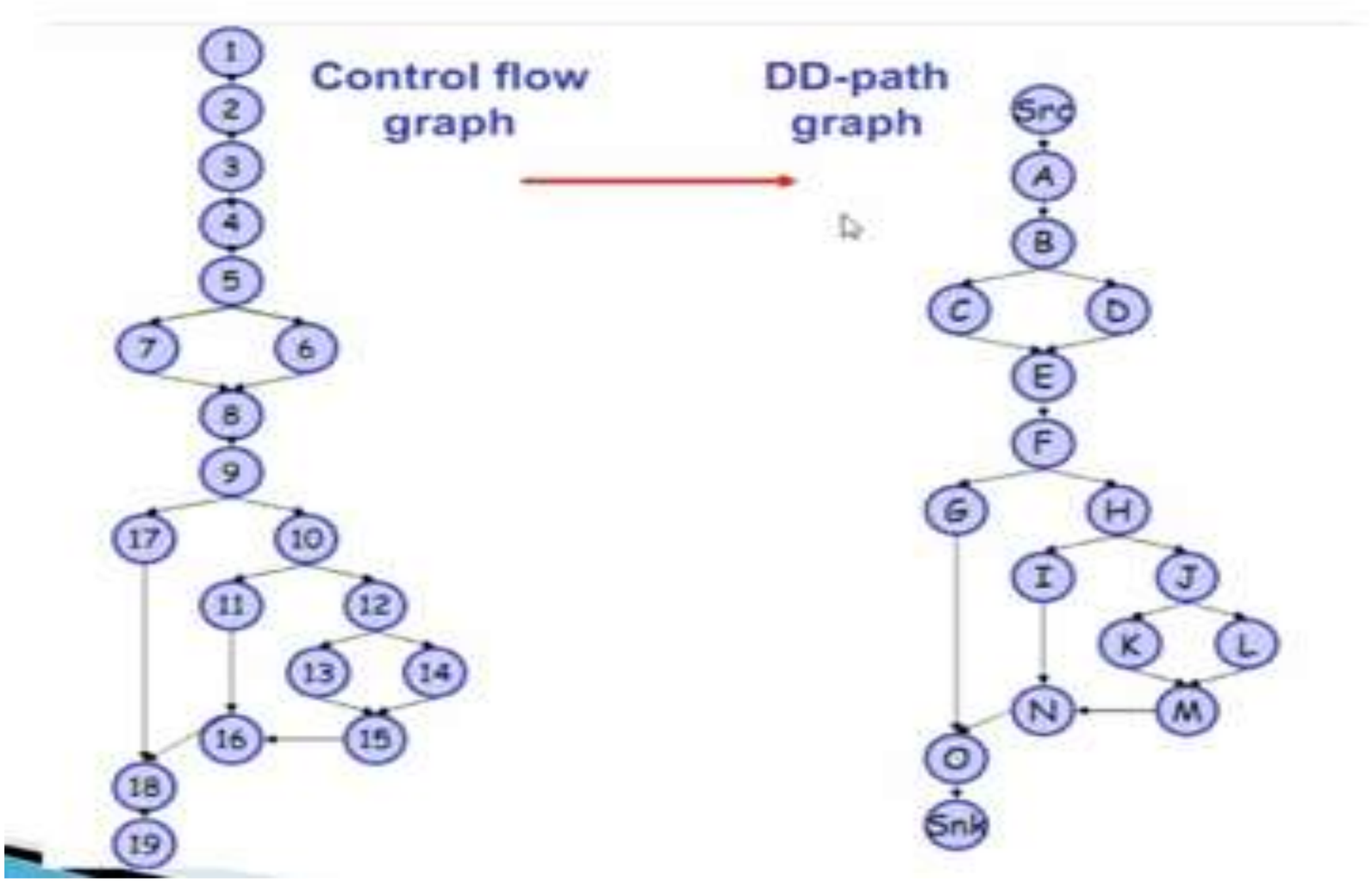
Control Flow Graph (CFG)

```
1   program TRIANGLE
2   input (a)
3   input (b)
4   input (c)
5   if (a<b+c) AND (b<a+c) AND (c<a+b)
6      then IsATriangle = T
7      else IsATriangle = F
8   endif
9   if IsATriangle
10     then if (a=b) AND (b=c)
11        then Output = "Equilateral"
12        else if (a != b) AND (b != c) AND (a != c)
13              then Output = "Scalene"
14              else Output = "Isosceles"
15        endif
16     endif
17  else Output = "Not a triangle"
18  endif
19  end TRIANGLE
```



DD Path Graph

- Flow graph generation is the first step of path testing. The second step is to draw a DD path graph from the flow graph.
- The DD path graph is known as decision to decision path graph. Here, we concentrate only on decision nodes.
- The nodes of flow graph, which are in a sequence are combined in a single node.
- DD path graph is a directed graph in which nodes are sequence of statements and edges represent control flow between nodes.


Control flow graph
DD-path graph
1
2
3
4
5
7
6
8
9
17
10
11
12
13
14
16
15
18
19
Src
A
B
C
D
E
F
G
H
I
J
K
L
N
M
O
Snk

Independent Path

- The DD path graph is used to find independent paths. We are interested to execute all independent paths at least once during path testing.
- It is quite interesting to use independent paths in order to insure that
  - (i) Every statement in the program has been executed at least once.
  - (ii) Every branch has been exercised for true and false condition.

Cyclomatic Complexity V(G)

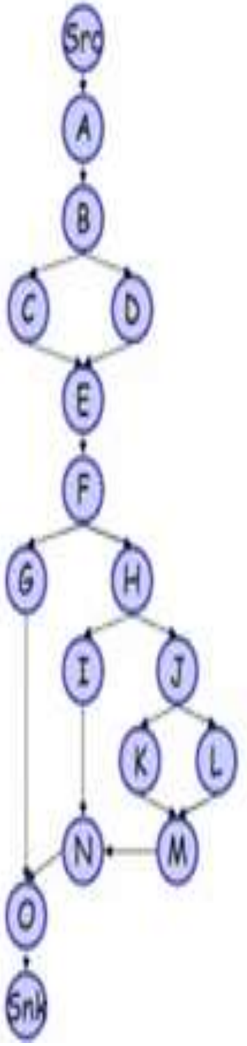
First Method: McCabes cylomatic metric of a graph G,

$$V(G) = e - n + 2P$$

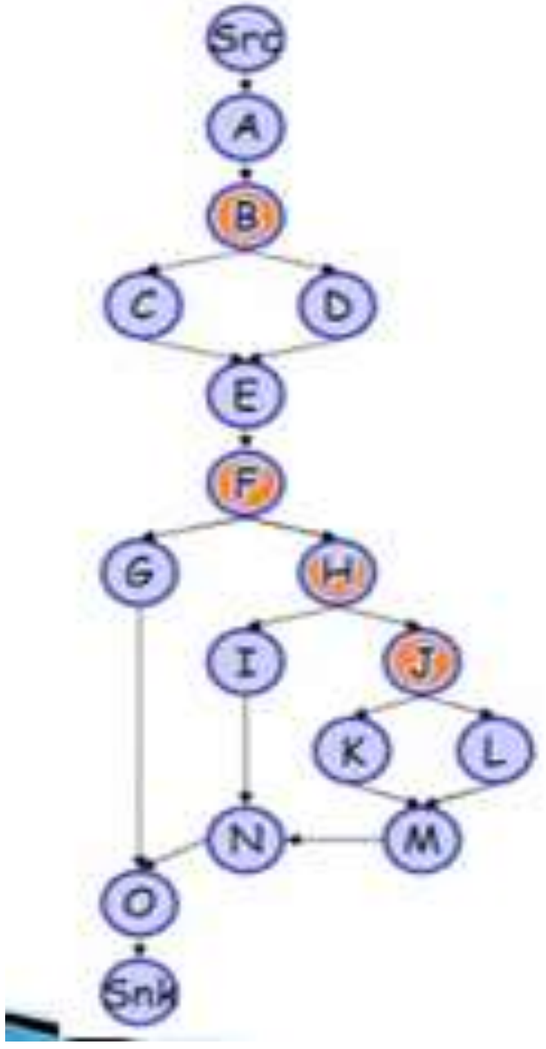
Where e= edges
n= nodes
P= Connected Components

Second Method: predicate node +1

Third Method: number of Regions

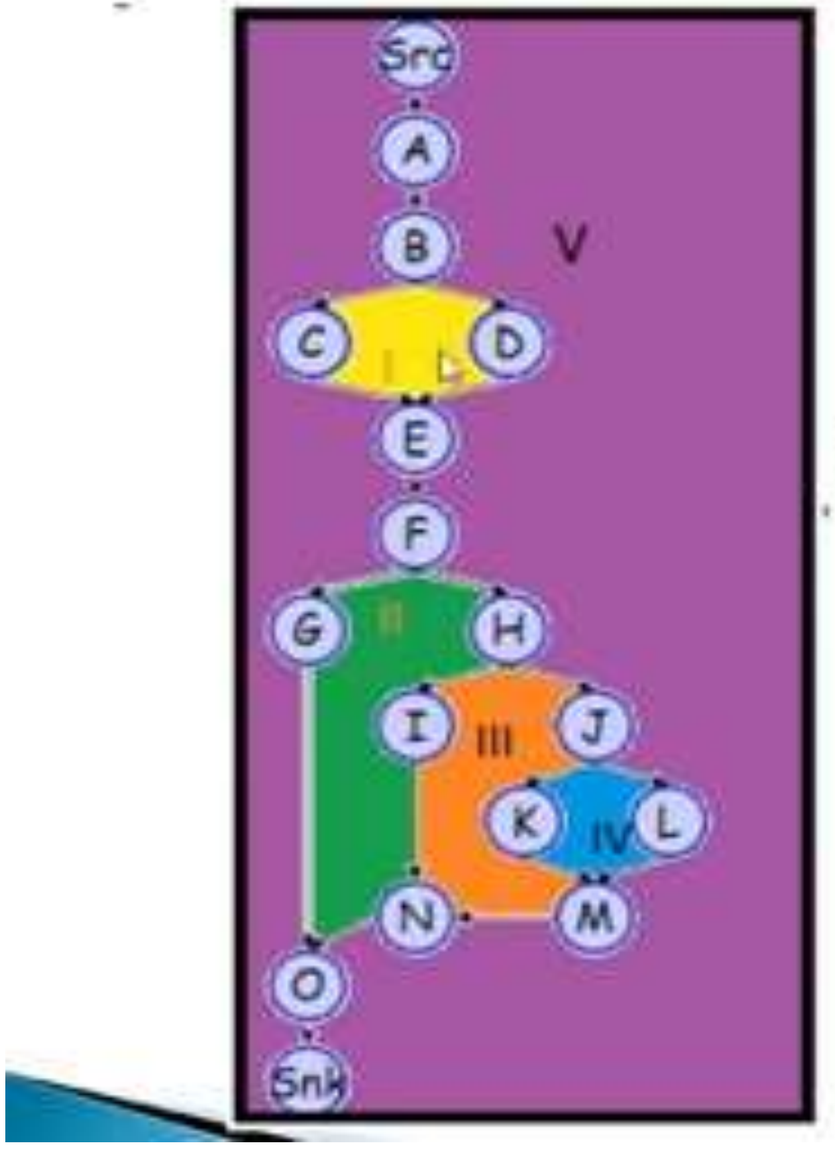


$$V(G) = e-n+2p$$
$$= 20-17+2$$
$$= 5$$



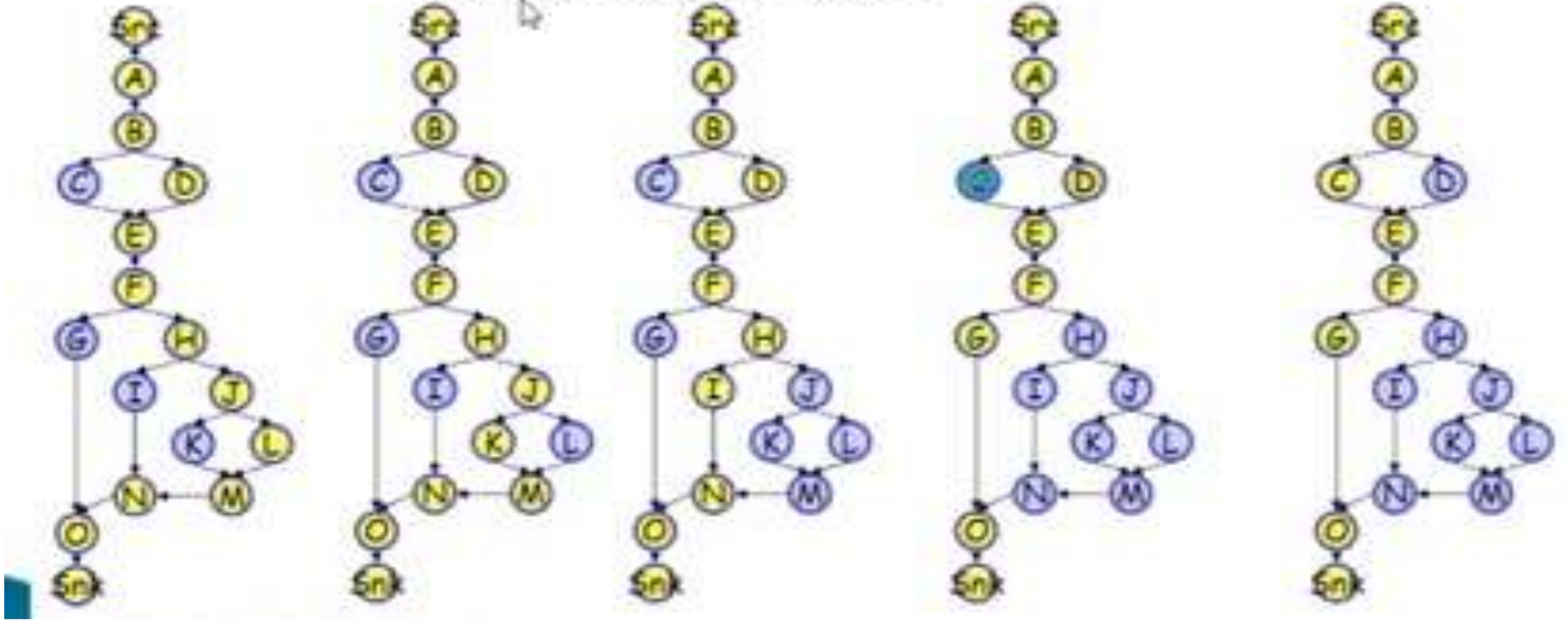
V(G) = e-n+2p
= 20-17+2
= 5

V(G) = Predicate Node + 1
= 4+1
= 5

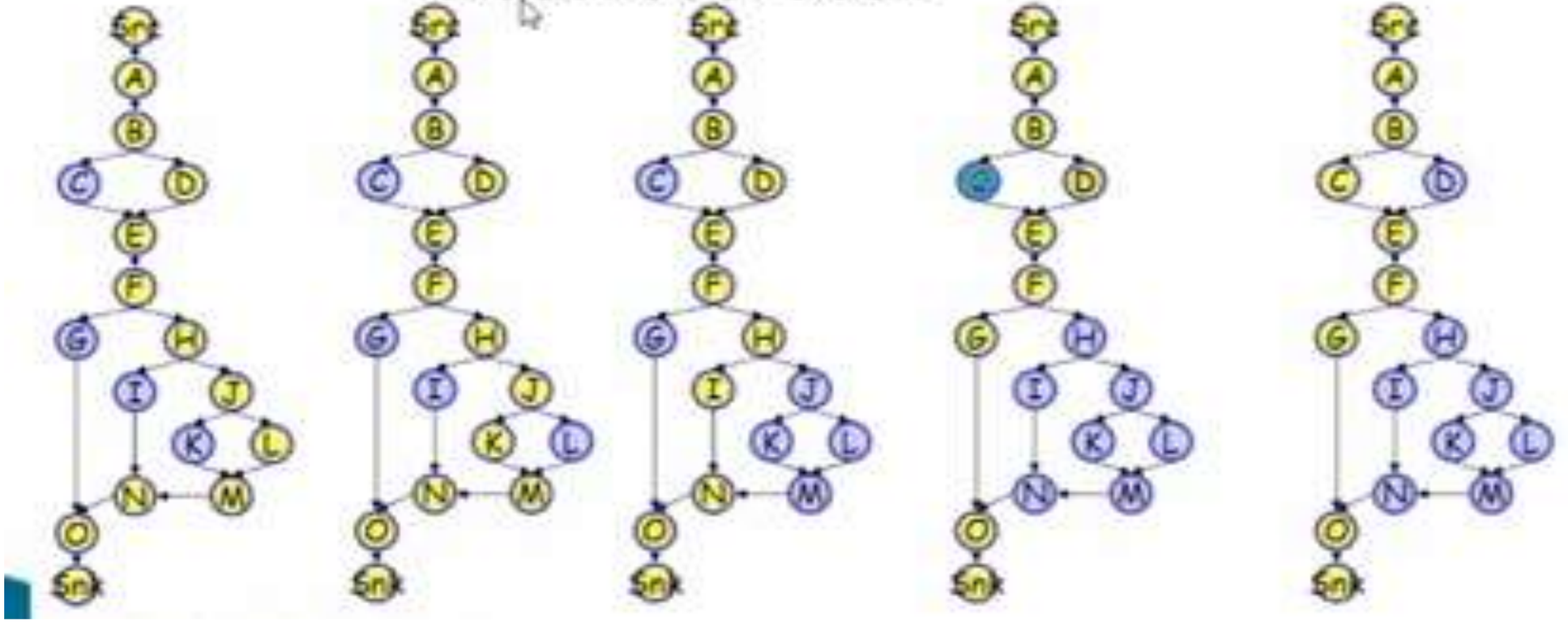


No. of Regions = 5

Path 1: src-A-B-D-E-F-H-J-L-M-N-O-snk
Path 2: src-A-B-D-E-F-H-J-K-M-N-O-snk
Path 3: src-A-B-D-E-F-H-I-N-O-snk
Path 4: src-A-B-D-E-F-G-O-snk
Path 5: src-A-B-C-E-F-G-O-snk



Path 1: src-A-B-D-E-F-H-J-L-M-N-O-snk
Path 2: src-A-B-D-E-F-H-J-K-M-N-O-snk
Path 3: src-A-B-D-E-F-H-I-N-O-snk
Path 4: src-A-B-D-E-F-G-O-snk
Path 5: src-A-B-C-E-F-G-O-snk



Path 1: src-A-B-D-E-F-H-J-L-M-N-O-snk

Path 2: src-A-B-D-E-F-H-J-K-M-N-O-snk

Path 3: src-A-B-D-E-F-H-I-N-O-snk

Path 4: src-A-B-D-E-F-G-O-snk

Path 5: src-A-B-C-E-F-G-O-snk

TEST CASE	Expected outcome
1	Scalene
2	Isosceles
3	Equilateral
4	Not a Triangle
5	Not a Triangle