Software Testing

19BCE10071
ABHISHEK SRIVASTAVA.

CLASS ASSIGNMENT - I

1) White box testing techniques are used for testing the module for initial stage testing. Since white-box testing is complementary to black-box testing, there are categories of bugs that can be revealed by white box testing. Errors which have came from the design phase will also be reflected in the code, therefore, we must execute white box testing. White box testing techniques helps in detecting typographical error that are not observed by black-box testing.

---

2) Different Criteria of logic Coverage.

(a) Statement coverage:
It is the first type of coverage that can be identified in the form of statement. It is assumed that if all statement of the module are executed once, every bug will be notified.

(b) Decision or Branch Coverage.
It states that each decision takes on every possible outcome atleast once.

(c) Condition Coverage.
It states that each condition in a decision takes on all possible outcomes atleast once.

(d) Decision Coverage.
If the decision is being tested, the condition coverage would allow one to write 2 test cases.

Test case 1 : A is True, B is False.
Test case 2 : A is False, B is True.

(e) Multiple Condition Coverage.
Here, even decision/condition outcomes of all conditions. The reason is that we have covered all possible outcomes for each condition in the decision.

3] Basic Path Technique.

It is based on the controlled structure a flow graph is prepared and all the possible paths can be covered and executed during testing.

Basic Path testing is a technique of selecting the paths that provide a basic set of execution paths through the programs.

---

4) Junction Node is a Node with more than one arrow entering it is called a junction.

Decision Node is a Node with more than one arrow leaving it is called a decision node.

5] Independent path is any path through the graph that ~~induces~~ introduces atleast one new set of processing statements on new condⁿ.

An Independent path must move along atleast one edge that has not been travelled before the path is defined.

6] Cyclomatic complexity is a ~~cyct~~ software matric used to determine the complexity of a program. It counts the no. of decisions in the source code. The higher the count, the more complex the code. Cyclometic complexity can be used in two ways. limit code complexity.

Determine the number of test case required.

Formula : $E - N + 2P$.

$E \rightarrow$ No. of Edges.

$N \rightarrow$ No. of Nodes

$P \rightarrow$ No of disconnected paths.

7) When a decision node has exactly two arrows leaving it, then we count it as a single decision node, and this uses the formula to calculate the number of nodes in:

$$d = k - 1,$$ where $k \rightarrow$ no. of arrow leaving

---

8) Let us say that a program P has 3 components: X, Y, Z.

Then we prepare the flow graph for P and for components: X, Y and Z. Then we prepare the flow graph for P and for components X, Y and Z. The complexity no. of the whole program is:

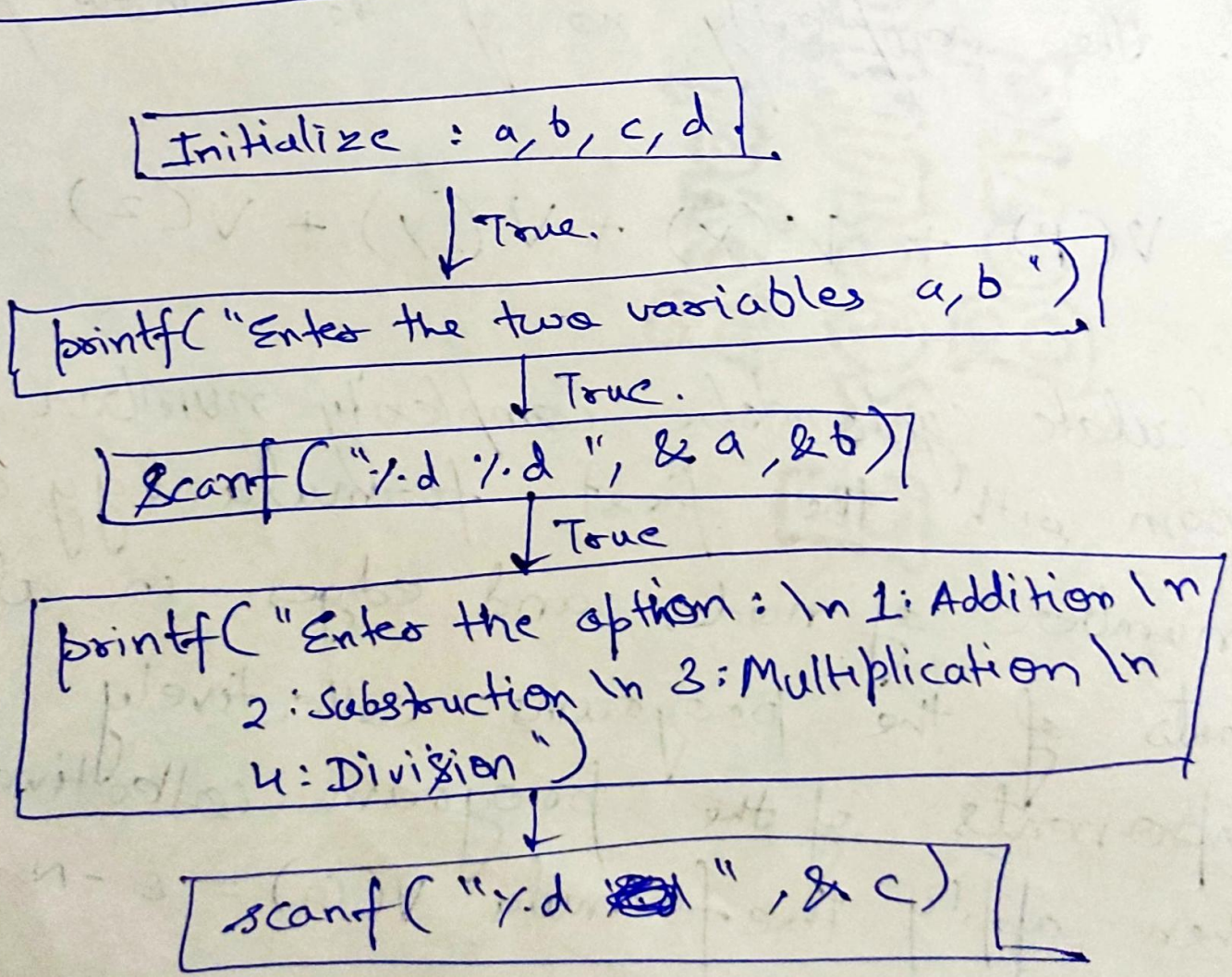
$$V(G) = V(P) + V(X) + V(Y) + V(Z)$$

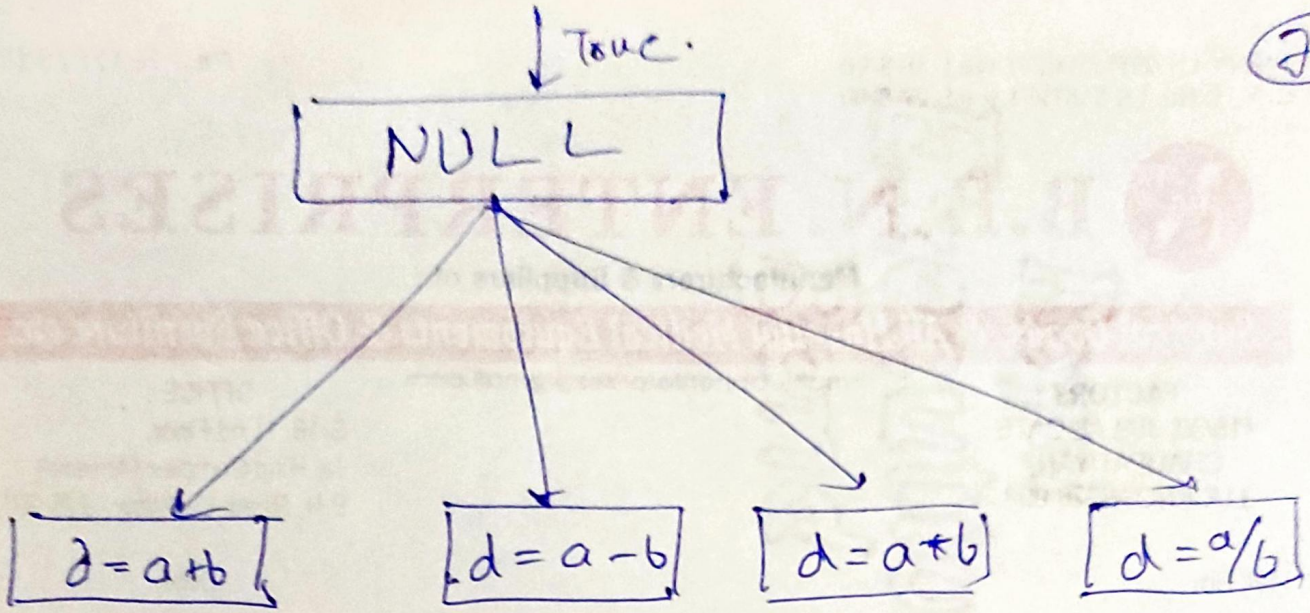
To calculate cyclomatic complexity number of the full program with the first formula by counting the number of nodes and edges in all the components of the programs collectively and the components of the program collectively and then apply the formula $V(G) = E - N + 2$.

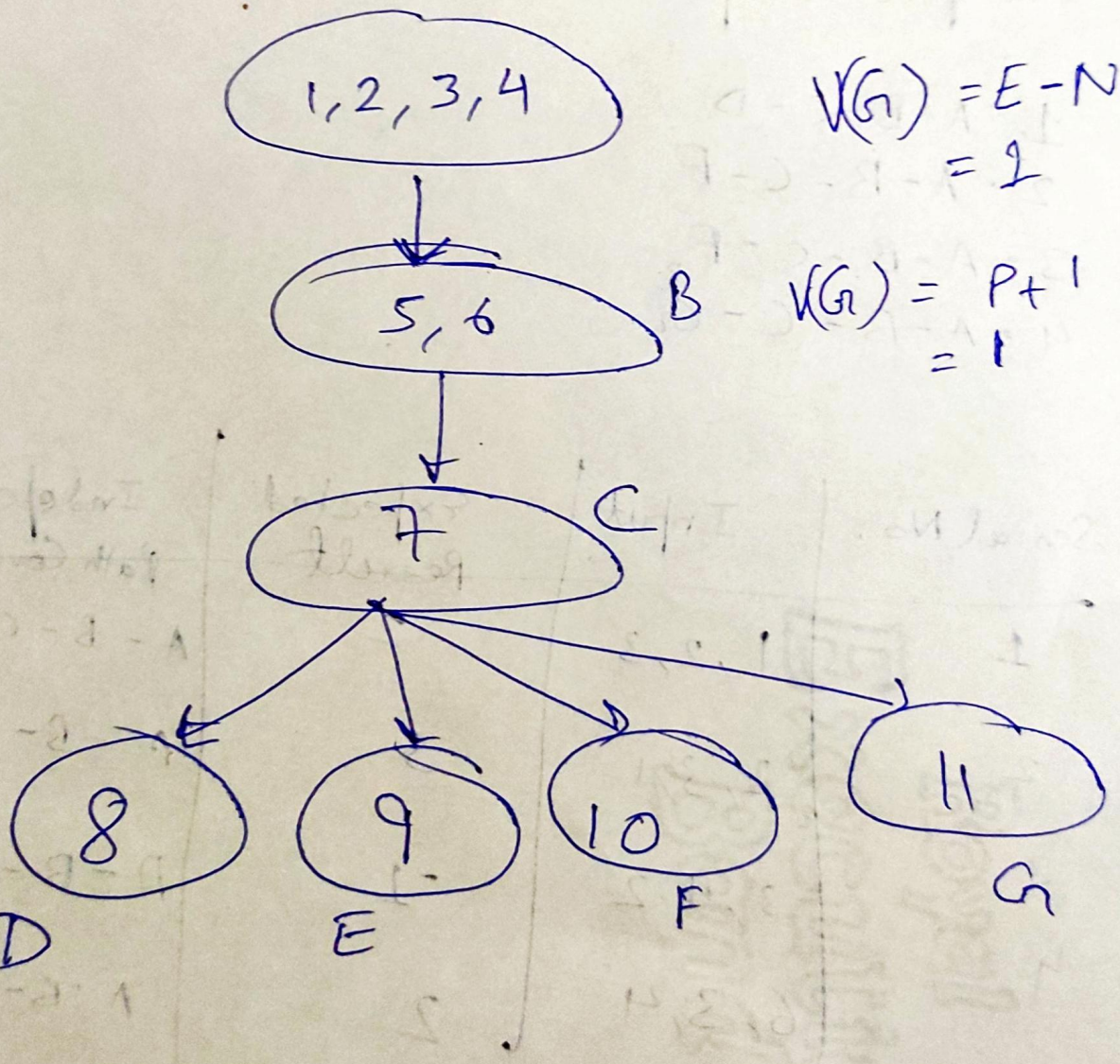
The complexity number denied collectively will be same as calculated above. Then,

$$V(PU \times UY \times UZ) = V(P) + V(X) + V(Y) + V(Z)$$

9] Data Flow Graph







$V(G) = E - N + 2$
$= 2$

$V(G) = P + 1$
$= 1$

(c) Independent path

(8)

1. A-B-C-D
2. A-B-C-F
3. A-B-C-F
4. A-B-C-G

d)

Serial No.	Input	Expected Result	Independent Path Covered.
1	1, 2, 3	2	A-B-C-F
2	2, 3, 1	5	A-B-C-D
3	3, 4, 2	-1	A-B-C-E
4	6, 3, 4	2	A-B-C-G

e) DU paths.

1) A - B - C - D
2) A - B - C - E
3) A - B - C - F
4) A - B - C - G