**PATH TESTING**

* Path testing is an approach to testing where you ensure that every path through a program

has been executed at least once.

* The string point for path testing is a program flow graph. This is selected of all paths

through the program.

* A flow graph consists of nodes representing decisions and edges showing flow of control.
* The flow graph is contracted by replacing program control statements by requirement

diagrams.

**FLOW GRAPH FOR A BINARY SEARN ROUTINE**

* Each node in a flow graph represents line in the program with executable statement.
* By tracking the flow therefore you can see that the independent

Path through the binary search flow graph are:

1,2,3,4,5,6,7,8,9,10,14

1,2,3,4,5,14

1,2,3,4,5,6,11,12,5,......

1,2,3,4,6,7,2,11,13,5,........

* If all of these paths are executed we can be sure that every Statement in the method has

been executed at least once and that every branch has been exercise for true and false

conditions.

* The number of tests that you need to ensure that paths through the program are exercised

is the same as the cyclamate complexity of code fragment that is begin tested.

13

12

11

14

9

10

6

5

8

7

4

3

2

1

While bottom <= up

Bottom > up Elam array

Elem array [Mid]! = key

[Mid]! =key Elam array Elam array

[Mid]>key [Mid] < key

Result:

Thus the program was executed successfully.