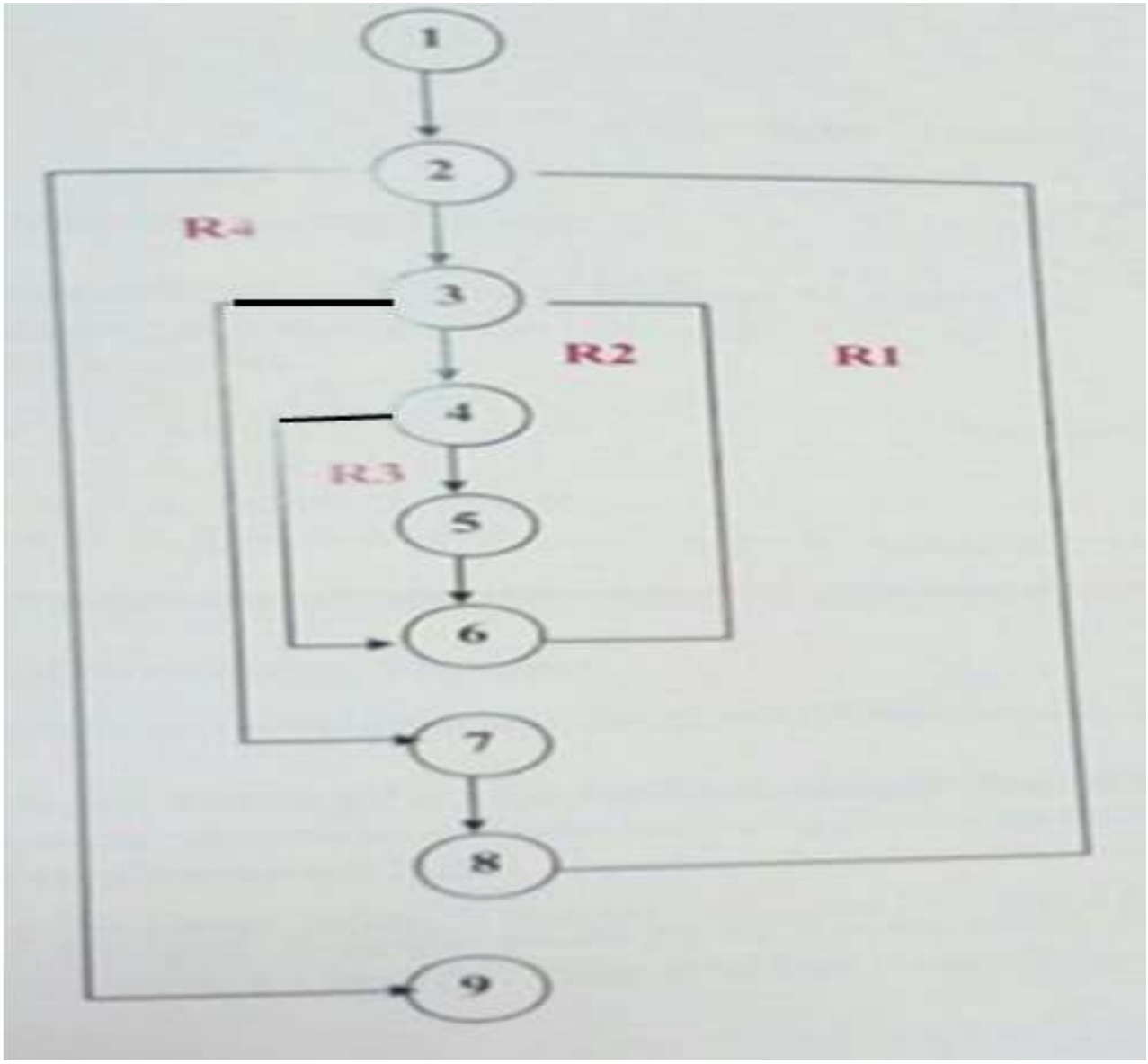
Software Engineering

Lecture nine

- **Draw the control graph and find the Cyclomatic number of the following program:**

```
1)void selectionsort(int[] table) {
        int   min, pos, temp;
 2)   for (int  i=0;i<table.length-1;i++)
          {
            min=table[i];
3)          for (j=i+1 ;j<table.length ;j++)
4)               if (table[j]<min)
5)                 {
                   min=table[j];
                     pos=j;
6)               {
7)               temp=table[i];
                 table[i]=min;
                 table[pos]=temp;
8)                }}//  end for
9)          }//end selectionsort
```

Flow graph:
(there are 4 regions :R1,R2,R3, and R4)



Cyclomatic complexity:

Cyclomatic complexity=number of regions in the flow graph =4
Or: number of condition nodes+1=3+1=4
Or: number of edges-number of nodes+2=11-9+2=4

b) Control structure testing

Control structure testing test the control statements in a program.

-condition testing it is a method of testing design that on the logical conditions of a module. A condition can be: simple, composed, Boolean, or relational. The condition testing strategies include:

a-Branch testing: it is the strategy of the simplest condition tests. For a composed condition, it is necessary to test the True and the False for every branch at least once, for example:

```
If c=1 then
     Statement 1
Else
     Statement 2
End if
```

b-domain testing :this strategy requires three or four tests for very relational expression while testing all possible cases of the control structure(<,> or =) for example:

E1(rational operator)E2

Tests are:E1>E2,E1=E2,E1<E2

c-Path testing:

In this the test case is executed in such a way that every path is executed at least once. All possible control paths taken, including all loop paths taken zero, once, and multiple (ideally, maximum) items in path testing technique, the test cases are prepared based on the logical complexity measure of a procedural design. In this type of testing every statement in the program is guaranteed to be executed at least one time.

• Example:

Read P

Read Q

IF P+Q > 100 THEN

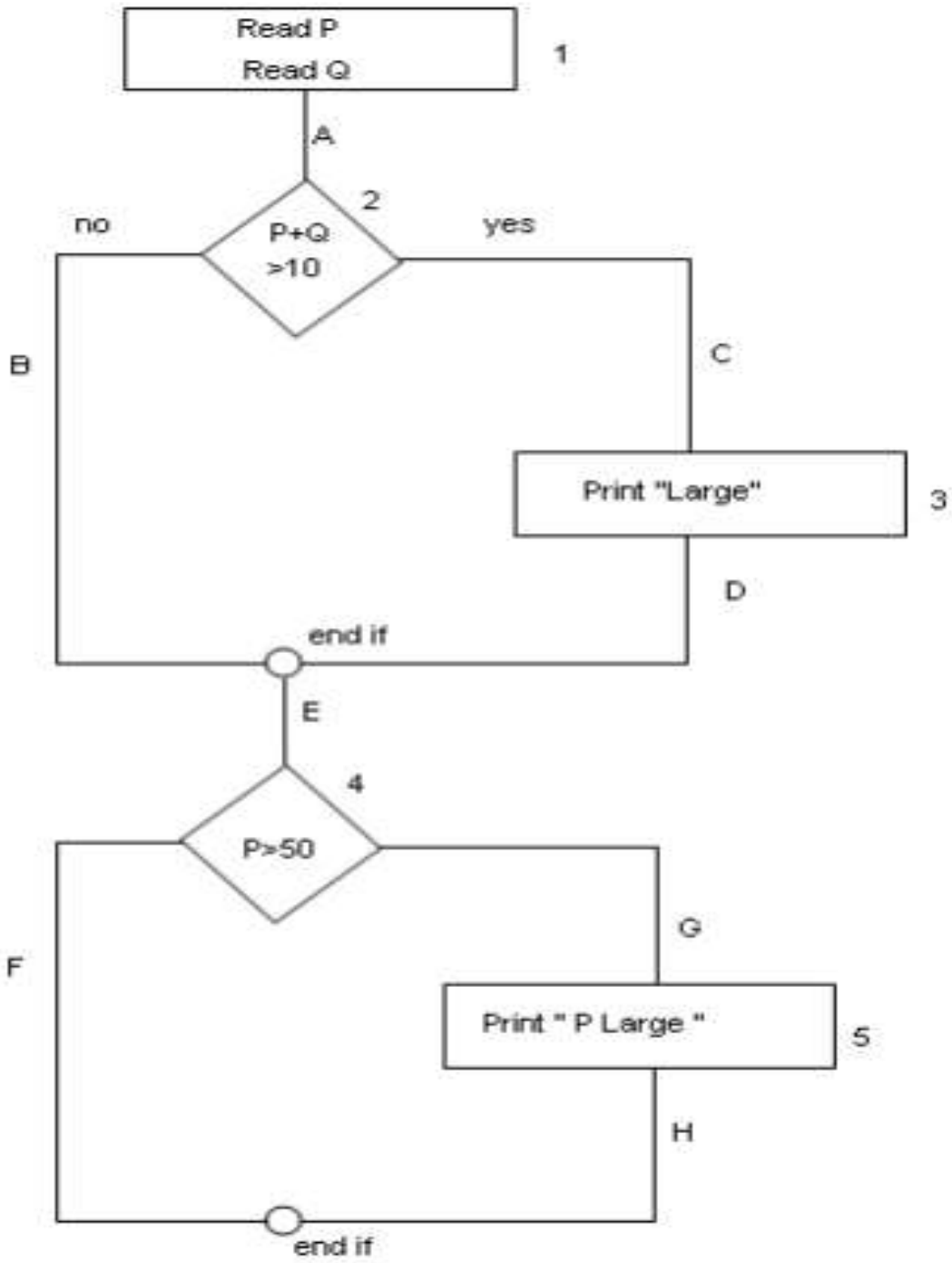
Print “Large”

ENDIF

If P > 50 THEN

Print “P Large”

ENDIF



Statement testing

To calculate Statement testing, find out the shortest number of paths following which all the nodes will be covered. Here by traversing through path 1A-2C-3D-E-4G-5H all the nodes are covered. So by traveling through only one path all the nodes 12345 are covered, so the Statement coverage in this case is 1.

domain testing

To calculate domain testing, find out the minimum number of paths which will ensure covering of all the edges. In this case there is no single path which will ensure coverage of all the edges at one go. By following paths 1A-2C-3D-E-4G-5H, maximum numbers of edges (A, C, D, E, G and H) are covered but edges B and F are left. To covers these edges we 3 can follow 1A-2B-E-4F. By the combining the above two paths we can ensure of traveling through all the paths. Hence Branch Coverage is 2. The aim is to cover all possible true/false decisions.

Path testing

Path testing ensures covering of all the paths from start to end.

All possible paths are-

1A-2B-E-4F

1A-2B-E-4G-5H

1A-2C-3D-E-4G-5H

1A-2C-3D-E-4F

So path coverage is 4.