

Test Case Design

2013

What is it?

- Vary definitions
- Covers all features
- Not too many test cases
- WhiteBox & BlackBox testing

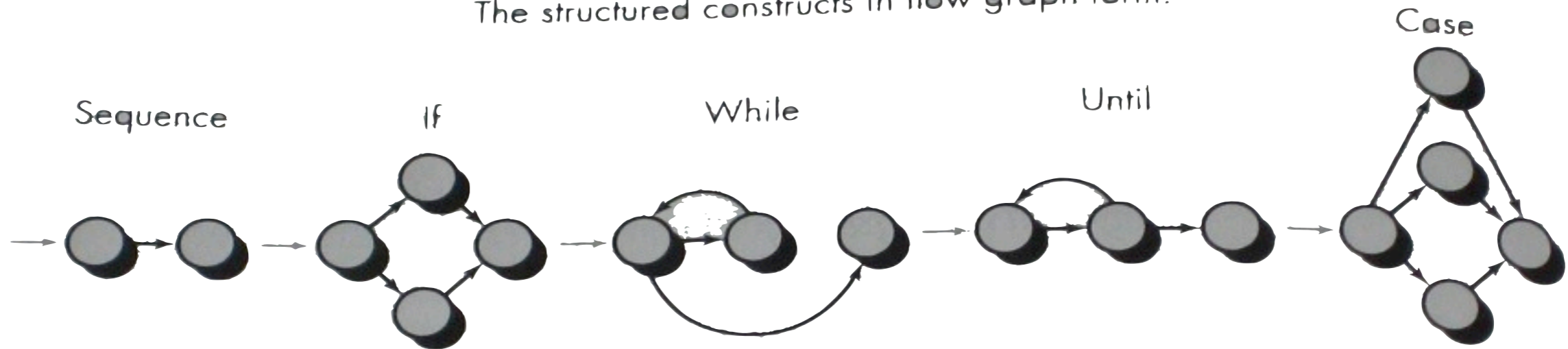
WhiteBox testing

- Glass box testing
- Uses control structure to derive test cases
- Guarantee that all independent paths must be execute at least once
- exercise all logical decision
- execute all loops within boundaries
- exercise internal data (validity)

Basis path testing

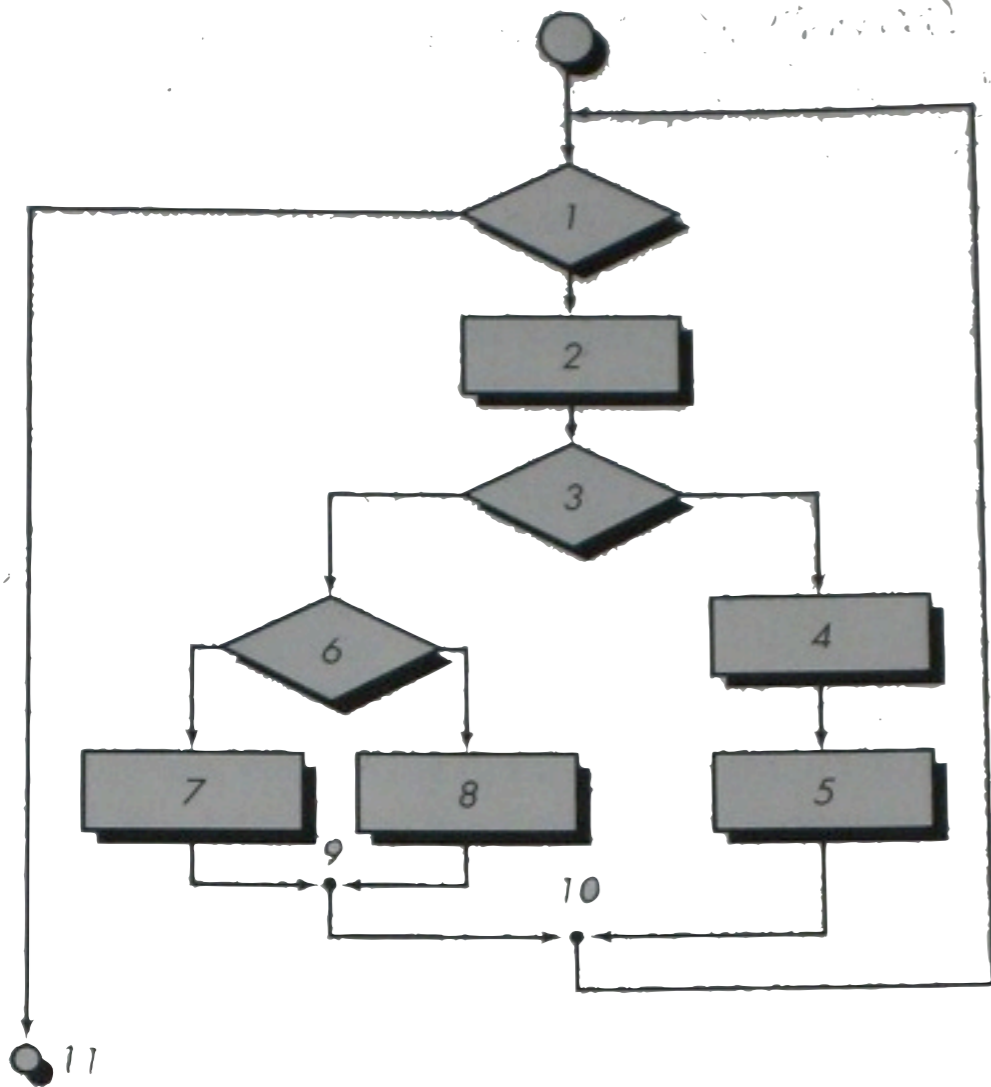
- WhiteBox testing
- Guarantee all statements will be execute once during testing
- Using Flow Graph

The structured constructs in flow graph form:

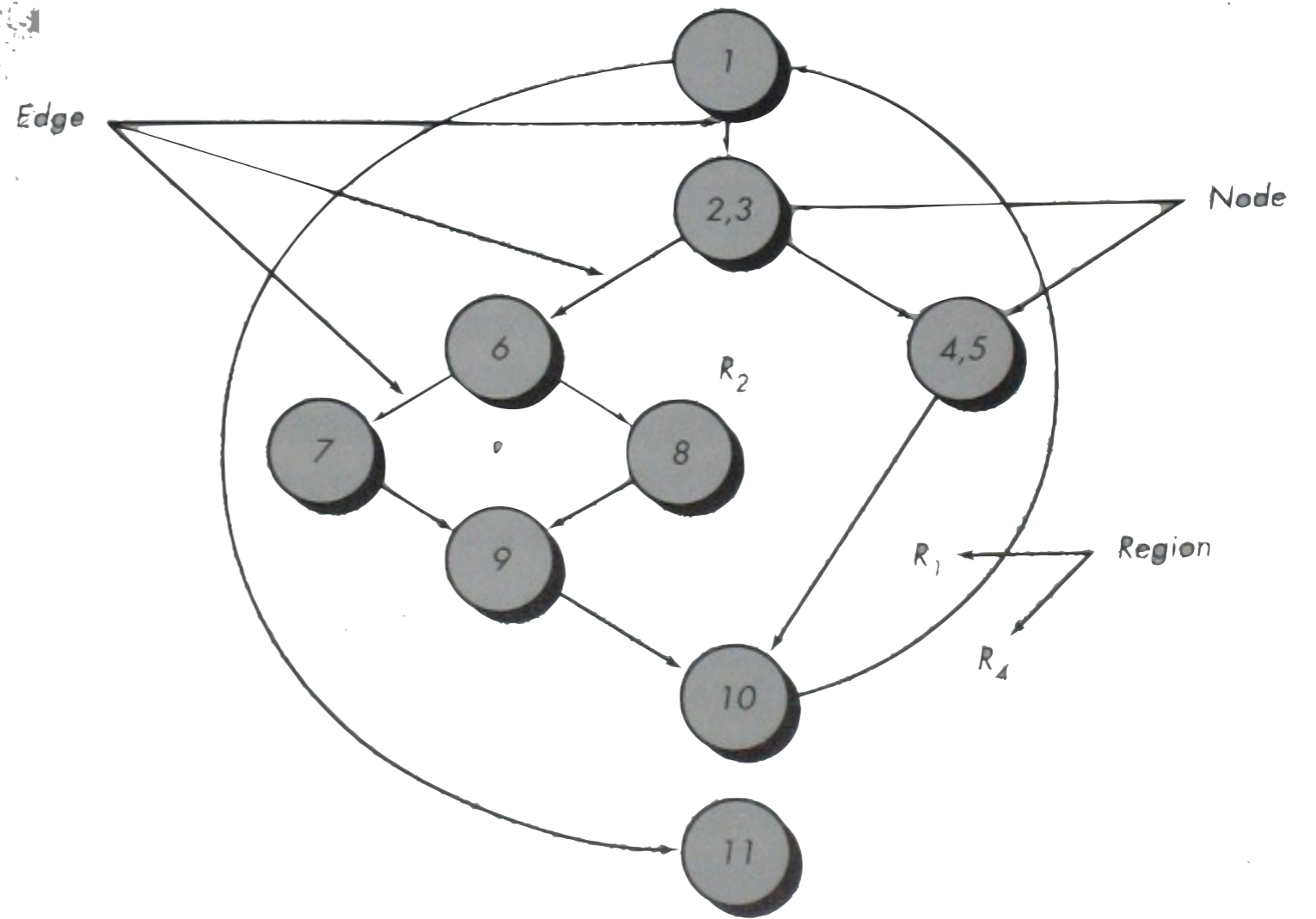


Where each circle represents one or more nonbranching PDL or source code statements

Flow graph notation



(A)



(B)

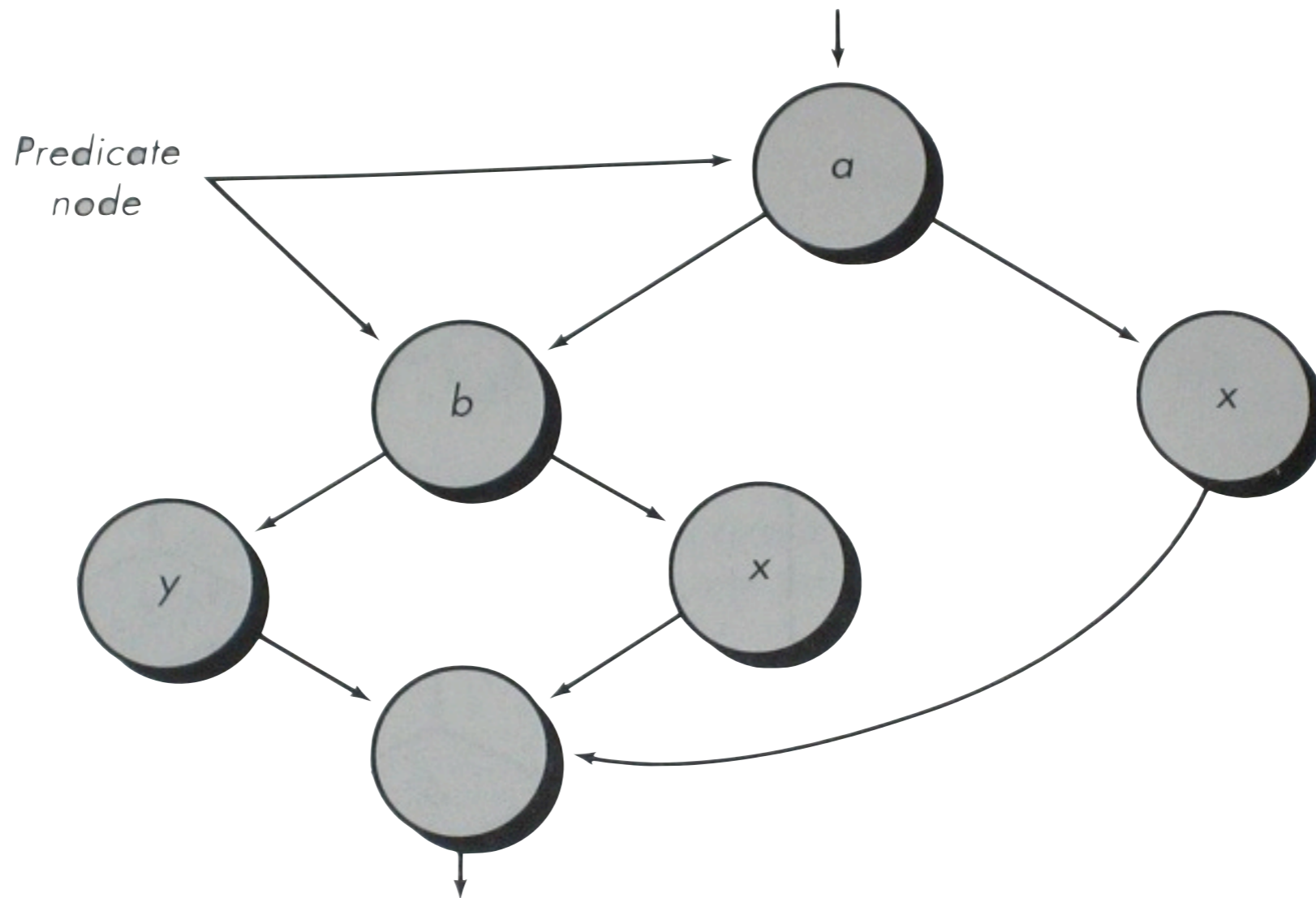
Flow chart to graph

Cyclomatic complexity

- Metric of logical complexity of program
- $V(G)$ = number of Region
- $V(G) = E - N + 2$
- $V(G) = P + 1$
- Upper bound for number of test cases

```
i = 1;
total.input = total.valid = 0;
sum = 0;
Do While value[i] <> -999 && total.input < 100
    total.input++;
    If value[i] >= minimum && value[i] <= maximum
        Then total.valid++;
            sum = sum + value[i];
        Else skip;
    EndIf
    i++;
EndDo
If total.valid > 0
    Then average = sum / total.valid;
    Else average = -999
EndIf
End program
```

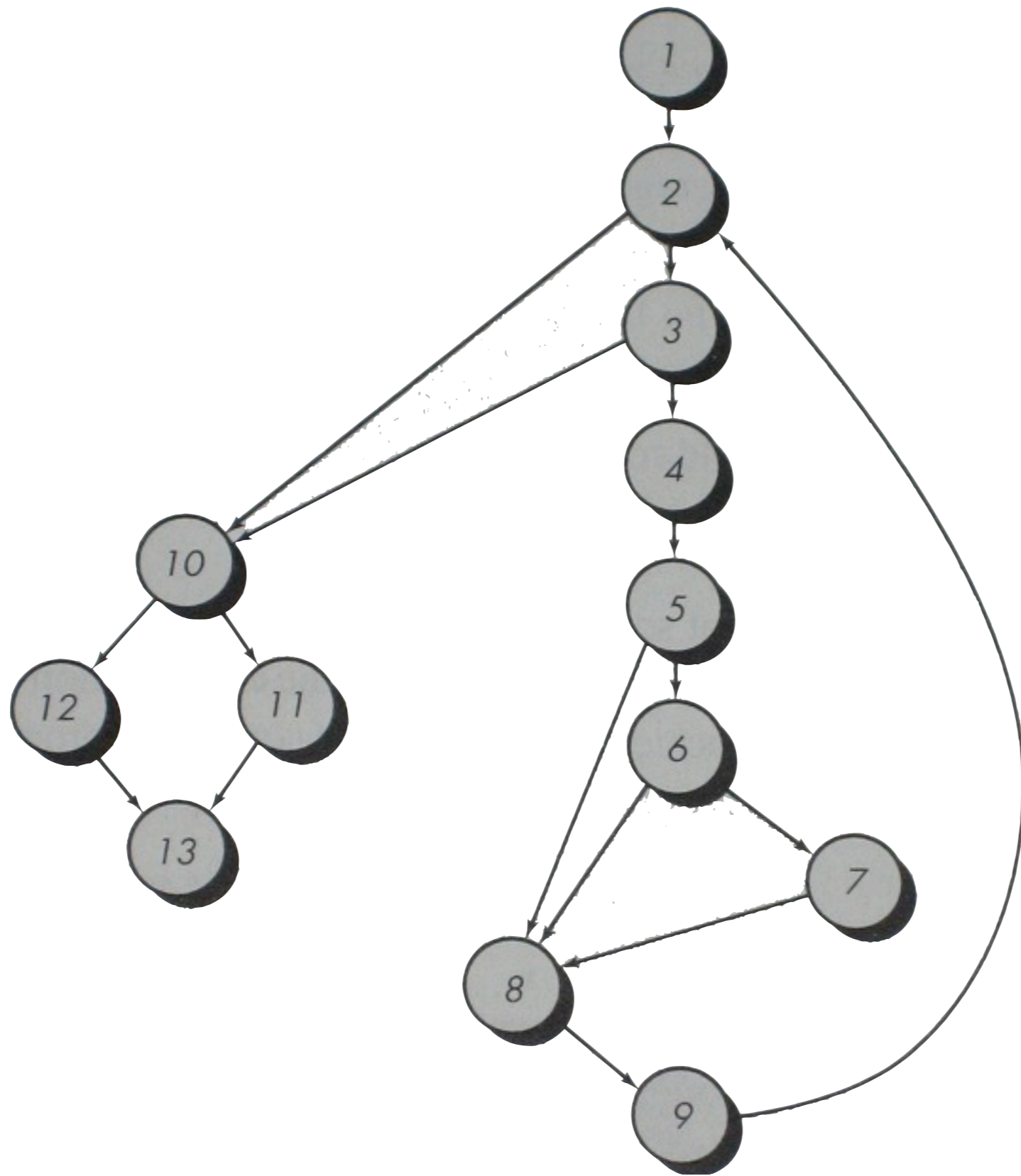
IF a OR b
then procedure x
else procedure y
ENDIF



Compound logic

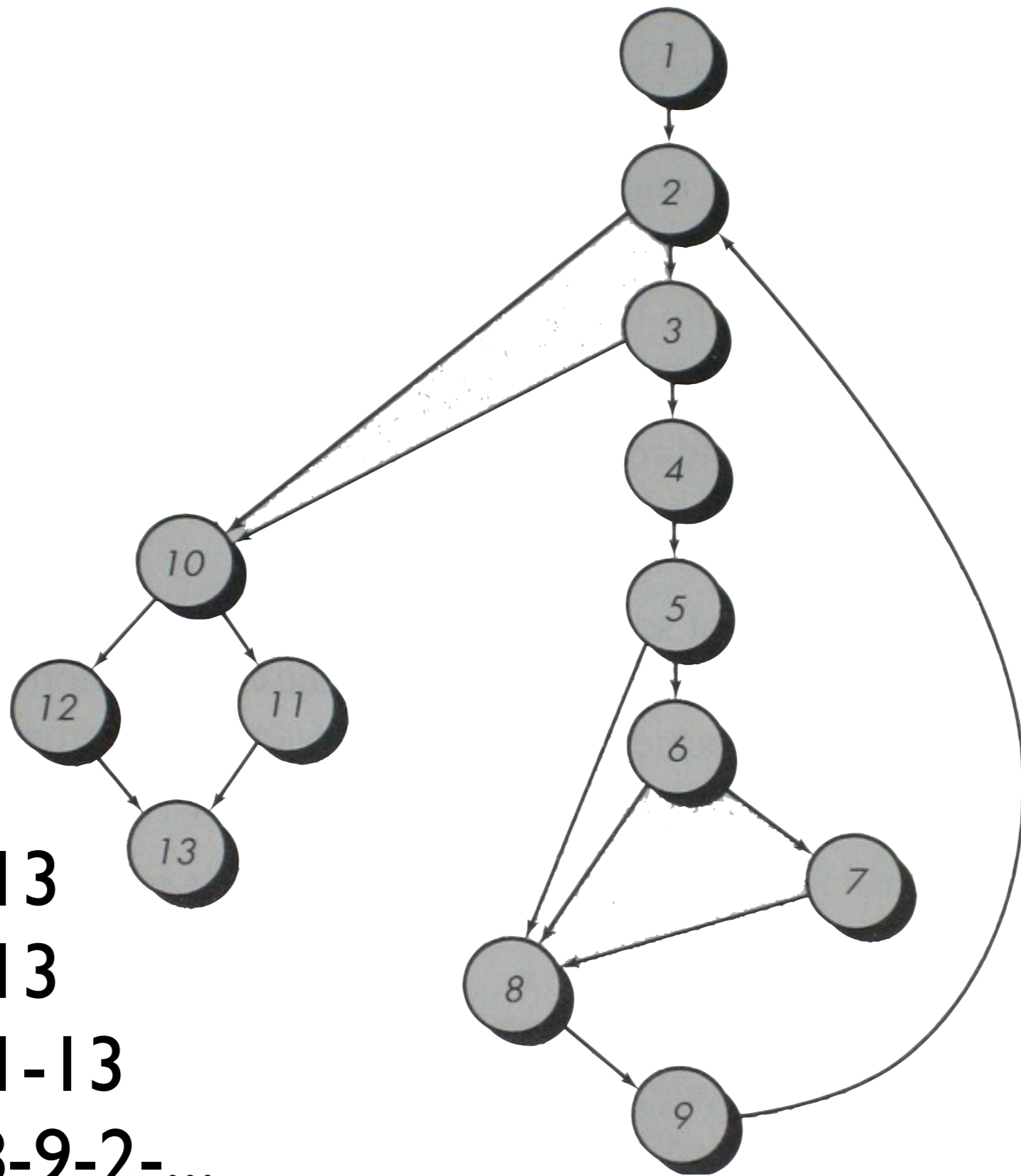
$$V(G) = ?$$

I
,



$V(G) = ?$

1
2



1-2-10-11-13

1-2-10-12-13

1-2-3-10-11-13

1-2-3-4-5-8-9-2-...

1-2-3-4-5-6-8-9-2-...

1-2-3-4-5-6-7-8-9-2-...

$V(G) = ?$