

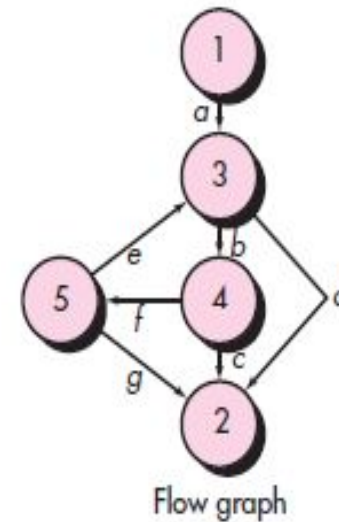
# GRAPH MATRICES



A graph matrix is a square matrix whose size (i.e., number of rows and columns) is equal to the number of nodes on the flow graph.



Each row and column corresponds to an identified node, and matrix entries correspond to connections (an edge) between nodes.



Connected to node		1	2	3	4	5
Node	1			a		
2						
3			d		b	
4			c			f
5			g	e		

Graph matrix

---

# LINK WEIGHT

- link weight : 1 - a connection exists  
: 0 - a connection does not exist
- Properties of link weight:
  - probability that a link (edge) will be execute.
  - processing time expended during traversal of a link
  - memory required during traversal of a link
  - resources required during traversal of a link.

# CONTROL STRUCTURE TESTING- CONDITION TESTING

---

- focuses on testing each condition in the program to ensure that it does not contain errors
- A simple condition is a Boolean variable or a relational expression, possibly preceded with one NOT ( $\neg$ ) operator. A relational expression takes the form  
E1 <relational-operator> E2  
where E1 and E2 are arithmetic expressions and <relational-operator>

# DATA FLOW TESTING

---

- The data flow testing method [Fra93] selects test paths of a program according to the locations of definitions and uses of variables in the program.
- $DEF(S) = \{X \mid \text{statement } S \text{ contains a definition of } X\}$
- $USE(S) = \{X \mid \text{statement } S \text{ contains a use of } X\}$
- It checks following a
  - If the variables are used without initialization.
  - If the initialized variables are not used at least once.

---

1. read x, y;

2. if(x>y)

3. a = x+1

else

4. a = y-1

5. print a;

variable	Defined	used
x	1	2, 3
y	1	2, 4
a	3,4	5

- Definition-Use chain of variable x [x, s, s']
- One simple data flow testing strategy is to require that every DU chain be covered at least once.



# LOOP TESTING

---

- The objective of Loop Testing is:
  - To fix the infinite loop repetition problem.
  - To know the performance.
  - To identify the loop initialization problems.
  - To determine the uninitialized variables.

# SIMPLE LOOPS

---

- Skip the loop entirely.
- Only one pass through the loop.
- Two passes through the loop.
- $m$  passes through the loop where  $m \leq n$ .
- $n - 1$ ,  $n$ ,  $n + 1$  passes through the loop.

# NESTED LOOPS

---

- Start at the innermost loop. Set all other loops to minimum values.
- Conduct simple loop tests for the innermost loop while holding the outer loops at their minimum iteration parameter (e.g., loop counter) values. Add other tests for out-of-range or excluded values.
- Work outward, conducting tests for the next loop, but keeping all other outer loops at minimum values and other nested loops to “typical” values.
- Continue until all loops have been tested.



# CONCATENATED LOOPS

---

- Use same approach for simple loop if loops are independent
- Use nested loop approach if loops are dependent

---

# **BLACK BOX TESTING**

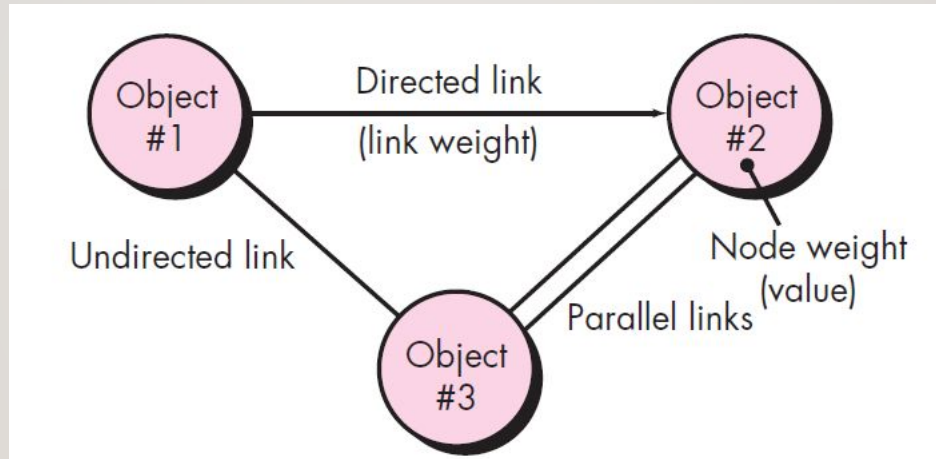
# BLACK-BOX TESTING

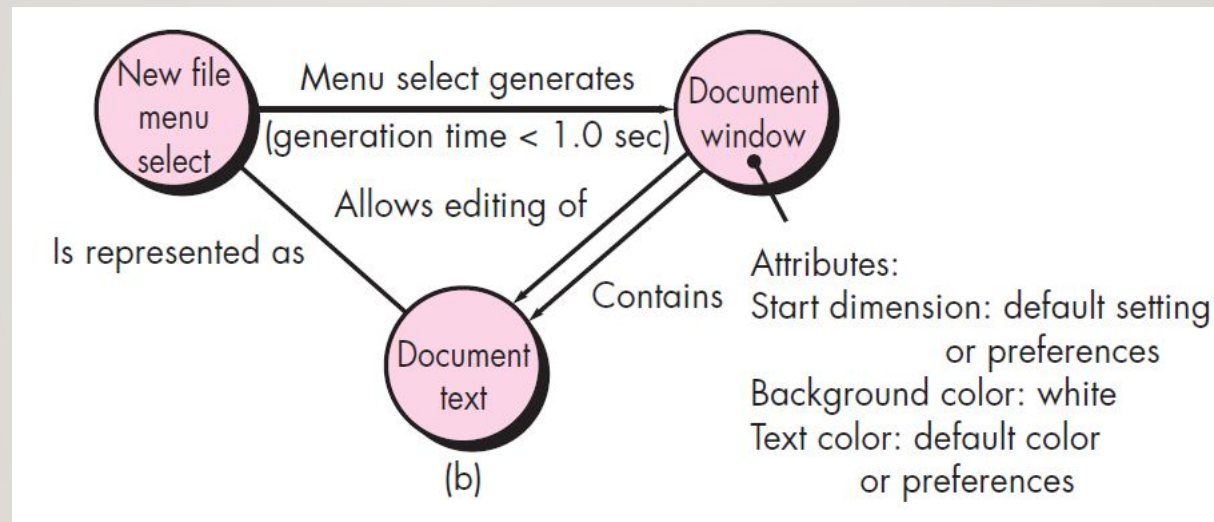
---

- Incorrect or missing functions
- interface errors
- errors in data structures or external database access
- behavior or performance errors
- initialization and termination errors

# GRAPH-BASED TESTING

---







# BEHAVIORAL TESTING METHODS

---

- **Transaction flow modelling**
- **Finite state modelling**
- **Data flow modelling**
- **Timing modeling**

# EQUIVALENCE PARTIONING

---

- It divides the input domain of a program into classes of data from which test cases can be derived.
- Guidelines
  - Range- 1 valid and 2 invalid classes
  - Specific values- 1 valid and 2 invalid classes
  - Member of set, 1 valid and 1 invalid class
  - Boolean, 1 valid and 1 invalid class

# BOUNDARY VALUE ANALYSIS

---

- leads to a selection of test cases that exercise bounding values.
- Guidelines
  - i/p specified as range bounded by values a and b- just above and below
  - Specified a number of values
  - Test cases should be designed to create an output report that produces the maximum and minimum allowable number of table entries.
  - Prescribed boundary, exercise that data structure at its boundary.

# ORTHOAGONAL ARRAY TESTING

---

- Black box testing techniques
- Test large numbers of possible input combinations
- Pair wise testing
- Any two columns give pair wise combinations
- 2-D array of numbers



	Top	Left	Right
TS1	Displayed	Displayed	Displayed
TS2	Displayed	Hidden	Hidden
TS3	Hidden	Displayed	Hidden
TS4	Hidden	Hidden	Displayed





- 
- No of variables-column
  - No of possible values-factors
  - Find OA with Smallest no of rows
  - Place the variable name in column
  - Given by Taguchi Designs (orthogonal array)

## L9 ORTHOGONAL ARRAY

---

Test case	Test parameters			
	P1	P2	P3	P4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1