/ ①

Lecture VI

DIRECTED Graphs

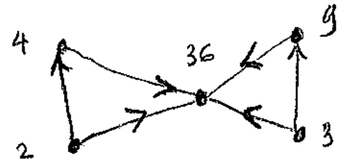
In chapter 5 we represented a relation R on a finite set A by digraph we represented the elements of A by a set of nodes and drew a directed edge from node a to node b.

6.1 Digraphs :

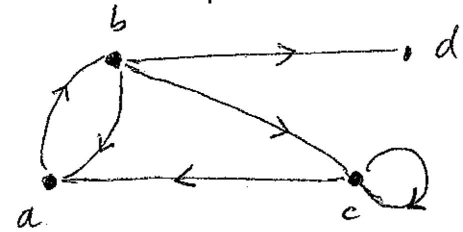
Definition: a digraph D is a finite set V (set of nodes or vertices of D) together with a subset E of $V \times V$. Each ordered pair of E is called a directed edge from a to b.

Exple: ① Let $V = \{2, 3, 4, 9, 36\}$, and define R on V as follows:

$x R y$ if $x \neq y$ and y divisible by x



② $V = \{(a,b), (b,a), (c,a), (c,c), (b,c), (b,d)\}$



Adjacency matrix :

	a	b	c	d
a	F	T	F	F
b	T	F	T	T
c	T	F	T	F
d	F	F	F	F

Definition: Let G be a digraph and let x be a node in G. The out-degree of x is the number of directed edges in G of the form (x, y), and indegree of x is the number of directed edges in G of the form (y, x). Total degree is equal to the sum of In and outdegree

Theorem 6-1: Let $G = (V, E)$ be a digraph. and M an $n \times n$ adjacency matrix for E

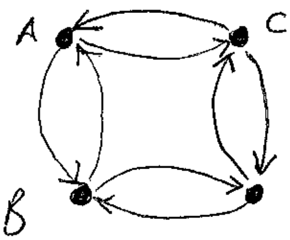
$$\sum_i \left[\text{Indegree}(v_i),\ v_i \in V\right] = \sum_j \left[\text{Out degree}(v_j),\ v_j \in V\right]$$

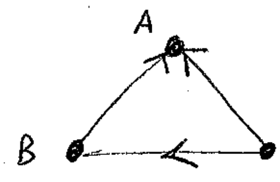
$$= \text{Total \# of edges of } G$$

A path is defined the same way as in undirected graph.
A cycle or circuit is a path from a vertex to itself.

Exple;

cyclic





Notcyclic
acyclic

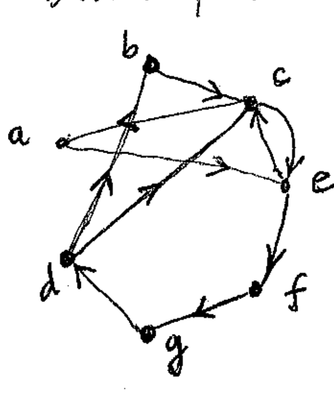
Acyclic digraph ⇒ existence of once source vertex.

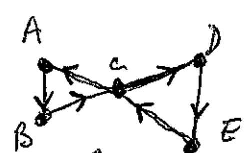
6.2 Path problems in Digraphs: By Modeling a real problem we face basically three difficulties, ① Existence of a path from between two vertices. ② finding the shortest path, and ③ counting the number of paths between 2 vertices.

Definition: ① Let A and B be two nodes in a digraph G. We say that B is reachable from A if there is a directed path from A to B of length at least 1.

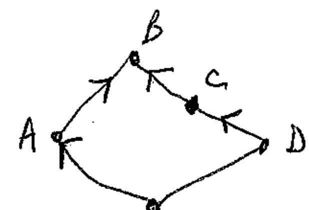
② A digraph G is connected if its underlying undirected graph is connected. It is strongly connected if, for any pair of distinct nodes A and B in G, there is a directed path from A to B

Exple:





strongly Connected



Theorem 6.2: Euler's Theorem for Digraphs.

Let G be a strongly connected digraph. Then G has a directed Euler circuit if and only if the in-degree of each node x in G is equal to its out degree.

Exple: from previous case ⓐ, ⓑ, ⓒ.

<u>Proposition-1</u>: Let $G = (V, E)$ be a digraph and suppose that the nodes of G are $\{v_1, v_2, v_3, \ldots, v_n\}$. Let M be the adjacency matrix of G and let v_j and v_k be nodes in G. Then there is a directed path of length q, $q \geq 1$, in G from v_j to v_k <u>iff</u> the $(i,j)^{th}$ entry of M^q is equal to 1.

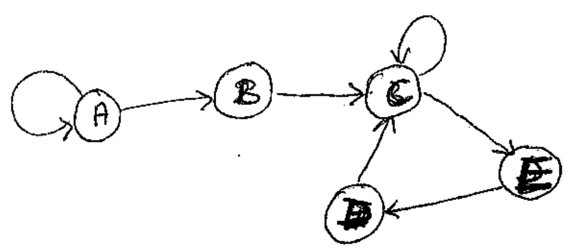
<u>Proof</u>: i) By Induction $q = 1$ by def. of adjacency matrix

Valid for q. — Show $q+1$.

$(\underbrace{a_j, \ldots a_m}, \underbrace{a_k})$ $\qquad M^{q+1} = M^q \cdot M$.

(ii) converse.

<u>Exple</u>:



5 nodes

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

We wish to know which pairs of nodes are connected by directed paths of length exactly 3. To do this we compute M^3

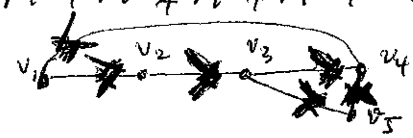
$$M^2 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad M^3 = \begin{pmatrix} 1 & 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

The <u>Reachability</u> <u>matrix</u> of G is the $n \times n$ matrix M^R, where the ~~with~~ entry $(j,k)^{th}$ entry is 1 if a_k from a_j and 0 otherwise. Thus the $(j,k)^{th}$ position of M^R is iff the $(j,k)^{th}$ position is 1 in one of matrices M^q for $1 \leq q \leq n$. Thus we have.

$$M^R = M + M^2 + \cdots + M^n$$

Exple :

$$M = \begin{array}{c|c} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}, \quad M^R = M^1 + M^2 + M^3 + M^4 + M^5$$



$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$+ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Proposition 2: A digraph G with two or more nodes is strongly connected iff each of the entries in its reachability matrix is equal to 1.

computing powers of matrix is sometimes a torture, we may have an Algorithm that by-passes this procedure, which is Warshall's Algorithm that is efficient and faster.

Warshall's Algorithm: Let $G = (V, E)$ be a digraph. suppose that $V = \{v_1, v_2, \cdots, v_n\}$ is the set of nodes, and M is the adjacency matrix of G. We construct a sequence of matrices $W_0, W_1, W_2, \cdots, W_n$ inductively as follows:

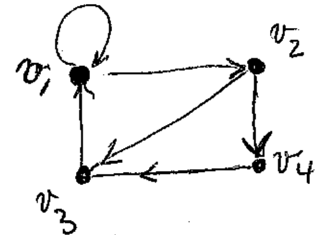
(a) $W_0 = M$

(b) Suppose W_K has been constructed and $0 \leq K < n$.
Let ${}_K W_{h,j}$ be the $(h, j)^{th}$ entry of W_K. To find W_{K+1}, set ${}_{K+1} W_{h,j} = {}_K W_{h,j}$ or $({}_K W_{h,K+1}$ and ${}_K W_{K+1,j})$

(c) The reachability matrix matrix of D is W_n

Exple: we compute W_1 through W_4 for the digraph G whose adjacency matrix is as follows:

$$W_0 = M = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

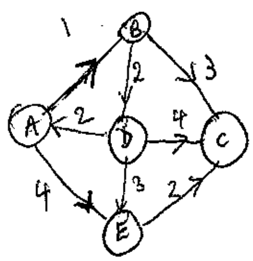


$$W_1 = \begin{pmatrix} {}_0W_{11} \text{ or } ({}_0W_{11} \text{ and } {}_0W_{11}) = 1 & {}_0W_{12} \text{ or } ({}_0W_{11} \text{ and } {}_0W_{12}) = 1 & {}_0W_{13} \text{ or } ({}_0W_{11} \text{ and } {}_0W_{13}) = 0 \\ {}_0W_{21} \text{ or } ({}_0W_{21} \text{ and } {}_0W_{11}) = 0 & {}_0W_{22} \text{ or } ({}_0W_{21} \text{ and } {}_0W_{12}) = 0 & {}_0W_{23} \text{ or } ({}_0W_{21} \text{ and } {}_0W_{13}) = 1 \\ {}_0W_{31} \text{ or } ({}_0W_{31} \text{ and } {}_0W_{11}) = 1 & {}_0W_{32} \text{ or } ({}_0W_{31} \text{ and } {}_0W_{12}) = 1 & {}_0W_{33} \text{ or } ({}_0W_{31} \text{ and } {}_0W_{13}) = 0 \\ {}_0W_{41} \text{ or } ({}_0W_{41} \text{ and } {}_0W_{11}) = 0 & {}_0W_{42} \text{ or } ({}_0W_{41} \text{ and } {}_0W_{12}) = 0 & {}_0W_{43} \text{ or } ({}_0W_{41} \text{ and } {}_0W_{13}) = 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad W_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad W_3 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = W_4$$

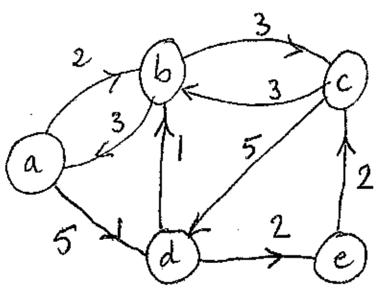
6-3- **Weighted Digraph:** A digraph is weighted if each ~~directed edge~~ ordered pair (x, y), directed edge, is assigned a positive number n called its weight and denoted by $\omega(x, y)$. We can also talk about the weight of a Path P.

Exple: (1)



$P = (a, b, d, c)$
weight of $P = 7$

(2) Weighted adjacency matrix

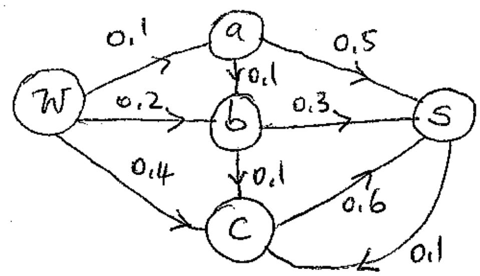


	a	b	c	d	e
a	∞	2	∞	5	∞
b	3	∞	3	∞	∞
c	∞	3	∞	5	∞
d	∞	1	∞	∞	2
e	∞	∞	2	∞	∞

③ A delivery truck transports goods from a warehouse W to a store located at S. The graph shows the length (in miles) and the directions of the streets of the town. The truck must obey the one way signs. In order to economize on fuel, the driver must find the shortest route from W to S. The legal routes from W to S are as follows

- -1- (W, a, S); length 0.6
- -2- (W, a, b, S); " 0.5
- -3- (W, a, b, c, S); " 0.9
- -4- (W, b, S) length 0.5
- -5- (W, b, c, S) " 0.9
- -6- (W, c, S) " 1.0

the best is route #2.



Next we investigate on an Algorithm that gives the path of the ~~least~~ weight or the shortest path.

Dykstra's Algorithm Let $PL(x)$ and $TL(x)$ denote Permanent and temporary label of x.

Find the shortest path from a to Z

Step-1- (Initial assignment of labels)

- Assign node a the P.L. zero, $PL(a) = 0$
- All other nodes x assigned ~~TL~~ ∞, $TL(x) = \infty$
- Set $V = a$.

Step-2- (Assignment of new Temporary labels).

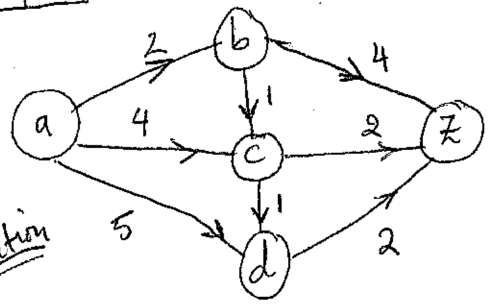
- To each node x without a P.L. compute a new $T.L(x)$ the following way $TL(x) = \min\{\text{old } TL(x), PL(V) + w(V, x)\}$

Step-3- (Assignment of a permanent label)

- Let q be a node with the smallest TL, that is not ∞. If no q can be found, stop: No directed graph from a to Z

$PL(Z)$ is the length of the shortest path from a to Z. Otherwise set $V = q$ and return to the beginning of step 2.

Exple:

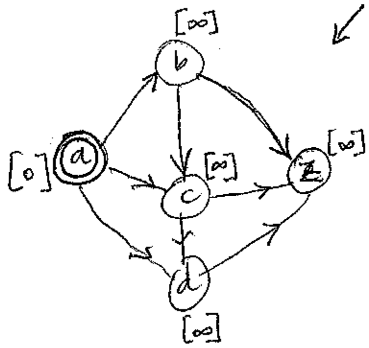


find the shortest path from a to Z

double circle the PL's

1st Iteration

Step-1- set $PL(a) = 0$, other node x $TL(x) = \infty$
set $V = a$.



step-2-

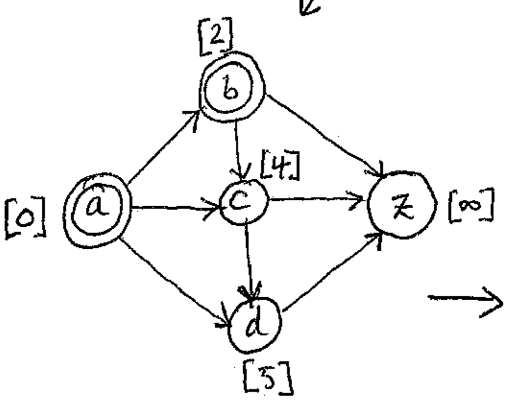
$TL(b) = \min\{\infty, 0+2\} = 2$

$TL(c) = \min\{\infty, 0+4\} = 4$

$TL(d) = \min\{\infty, 0+5\} = 5$

$TL(Z) = \min\{\infty, \infty\} = \infty$

step-3- Node b has the smallest new temporary label. Set $PL(b) = 2$ because $b \neq Z$, set $V = b$ and return to step -2-



2nd iteration

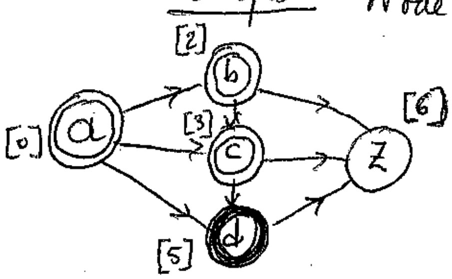
step-2-

$TL(c) = \min\{4, 2+1\} = 3$

$TL(d) = \min\{5, \infty\} = 5$

$TL(Z) = \min\{\infty, 2+4\} = 6$

step3- Node c has the smallest new temporary label. Set $PL(c) = 3$ since $c \neq Z$ set $V = c$. Return to step 2.



Iteration 3

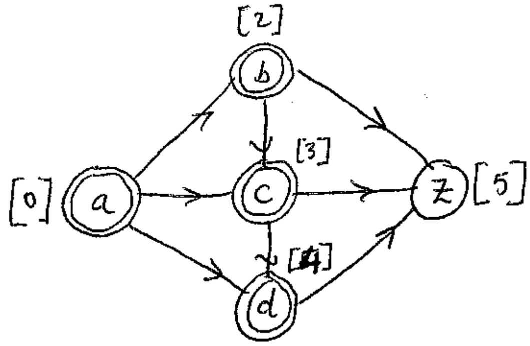
step-2-

$TL(d) = \min\{5, 3+1\} = 4$

$TL(z) = \min\{6, 3+2\} = 5$

step -3- Node d has the smallest new temporary label set $PL(d) = 4$. Because $d \neq z$ set $V = d$ and return to step -2-.

Iteration 4



step-2-

$TL(z) = \min\{5, 4+2\} = 5$

step 3. Set $PL(z) = 5$ and stop. The weight for the shortest path from a to z is 5

This Algorithm gives us the weight of the shortest path from a to z, not the path itself. We can modify step 1 and step 2 of the Algorithm so that at each iteration we also label node x with the shortest path $SP(x)$ from a to x that passes through only permanently labeled nodes.

If no such path exist, set $SP(x) = \phi$ if $P = (a, v_1, v_2, \dots, V)$ and $P'(a, v_1, v_2, \dots, V, x)$ obtained by adjoining the edge (V, x) to P, we shall denote $P' = (P, x)$. The modification is as follows:

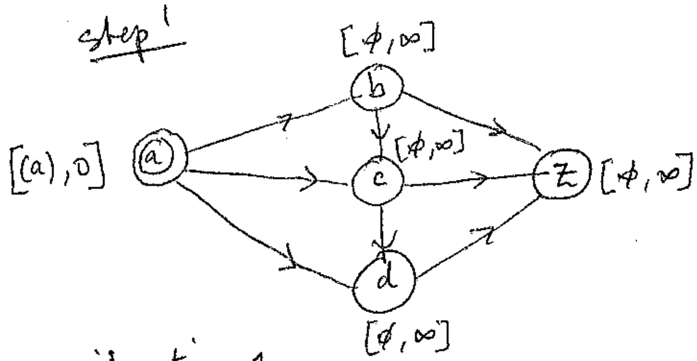
step-1- Also label node a with $SP(a) = (a)$. And every other node $SP(x) = \phi$

step-2- (We change $SP(x)$ only when we change the temporary label of x.) If the old label of x is not equal to its new label, set

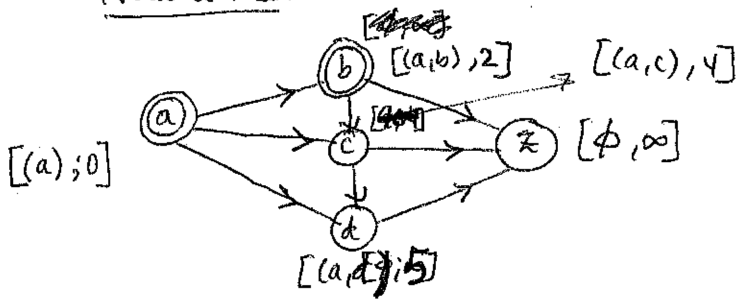
$$SP(x) = (SP(V), x)$$

Exple: consider the same graph as before, the path from a to x is listed before the $TL(x)$ of $PL(x)$ in the square brackets

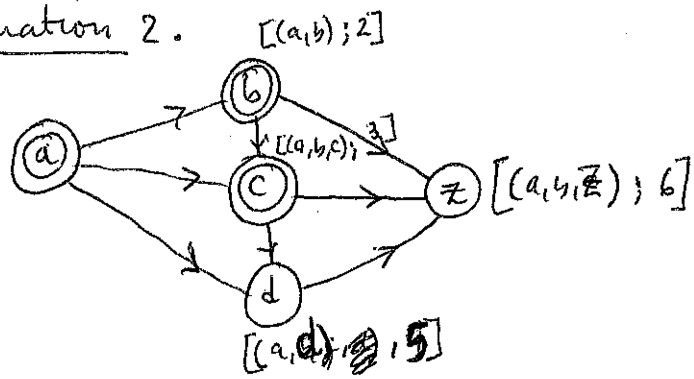
9 step 1



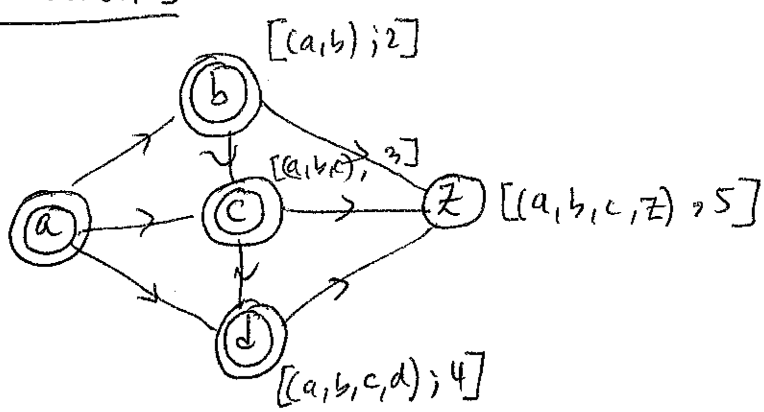
iteration 1.



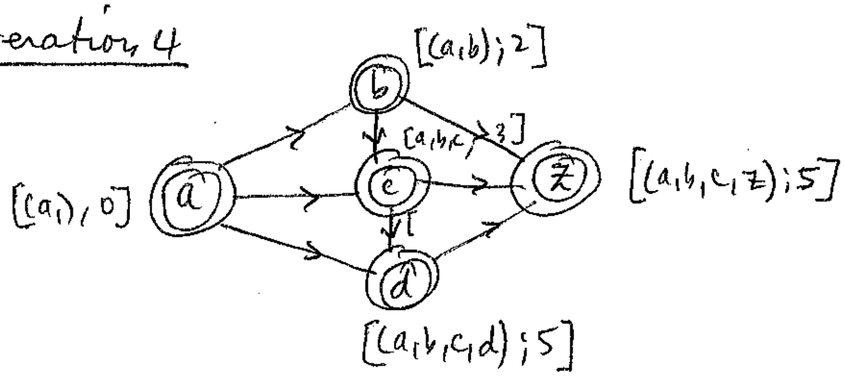
Iteration 2.



Iteration 3



Iteration 4



HW#5 Assgnt.

Page 204 # 2, # 3, # ~~5~~ 8

page 205: #12,

page 206: #14, # 18

page 245: # 5 → # of figure is confusing

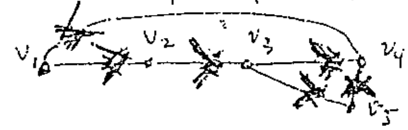
page 246: # 7, #8 optional

page 247 #11 @

page 248 # 16

Exple :

$$M = \begin{array}{c|ccccc} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline v_1 & 0 & 1 & 0 & 0 & 0 \\ v_2 & 0 & 0 & 1 & 0 & 0 \\ v_3 & 0 & 0 & 0 & 1 & 1 \\ v_4 & 1 & 0 & 0 & 0 & 0 \\ v_5 & 0 & 0 & 0 & 1 & 0 \end{array}, \quad M^R = M^1 + M^2 + M^3 + M^4 + M^5$$



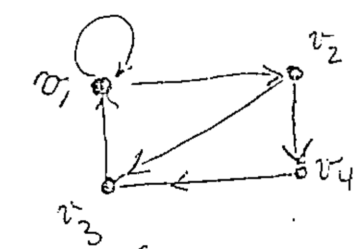
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$+ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Exple : . v_1 v_2 v_3 v_4 v_5

Exple : we compute W_1 through W_4 for the digraph G whose adjacency matrix is as follows:

$$W_0 = M = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$



$$W_1 = \begin{pmatrix} {}_0W_{11} \text{ or } ({}_0W_{11} \text{ and } {}_0W_{11}) = 1 & {}_0W_{12} \text{ or } ({}_0W_{11} \text{ and } {}_0W_{12}) = 1 & {}_0W_{13} \text{ or } ({}_0W_{11} \text{ and } {}_0W_{13}) = 0 \\ {}_0W_{21} \text{ or } ({}_0W_{21} \text{ and } {}_0W_{11}) = 0 & {}_0W_{22} \text{ or } ({}_0W_{21} \text{ and } {}_0W_{12}) = 0 & {}_0W_{23} \text{ or } ({}_0W_{21} \text{ and } {}_0W_{13}) = \\ {}_0W_{31} \text{ or } ({}_0W_{31} \text{ and } {}_0W_{11}) = 1 & {}_0W_{32} \text{ or } ({}_0W_{31} \text{ and } {}_0W_{12}) = 1 & {}_0W_{33} \text{ or } ({}_0W_{31} \text{ and } {}_0W_{13}) = \\ {}_0W_{41} \text{ or } ({}_0W_{41} \text{ and } {}_0W_{11}) = 0 & {}_0W_{42} \text{ or } ({}_0W_{41} \text{ and } {}_0W_{12}) = 0 & {}_0W_{43} \text{ or } ({}_0W_{41} \text{ and } {}_0W_{13}) = \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad W_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad W_3 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = W_4$$