Connections to Matrices and Relations

Written by Rachel Singh, last updated Sep 30, 2018

About

Trees are a handy structure in Data Structures, and are also a part of Graph Theory.

Topics

1. Adjacency Matrices

2. Matrix Multiplication

Adjacency Matrices

1. Adjacency Matrices

With an Adjacency Matrix, we can keep a list of the edges between vertices in a clean, organized format.

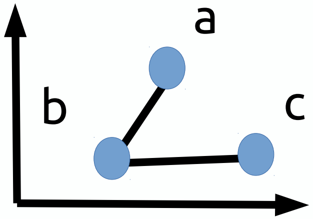
If we simply list out “1 to 5”, “1 to 2”, “2 to 4”, and so on, it would be difficult to read for lots of numbers. With a matrix, we can interpret it more easily and in an ordered format.

Notes

1. Adjacency Matrices

If we were storing a graph's data in a computer, how would we store it?

Does the (x,y) coordinates of those nodes matter?
Not really…



The data that matters is what vertices we have,
V = { a, b, c }
and what edges we have,
E = { [a, b], [b, c], … }

Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices

In a matrix, we have a set of rows and columns. Along each of these, we list out all the vertices in the graph.

We can reconstruct a graph from this matrix, knowing that ***0*** means no edge between two nodes, ***1*** means one edge between two nodes, ***2*** means two edges between, and so on…

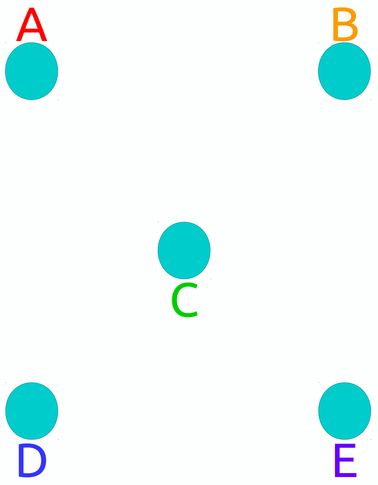
	A	B	C	D	E
A	0	1	1	1	0
B	1	1	1	0	1
C	1	0	0	0	0
D	1	0	0	0	2
E	0	1	0	2	1

Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices

When recreating the graph, we know which nodes we have – these are the row & column headers.



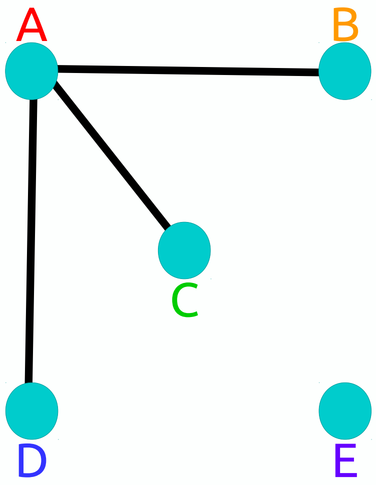
	A	B	C	D	E
A	0	1	1	1	0
B	1	1	1	0	1
C	1	1	0	0	0
D	1	0	0	0	2
E	0	1	0	2	1

Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices

When, we can look at one row at a time and begin drawing in our lines.



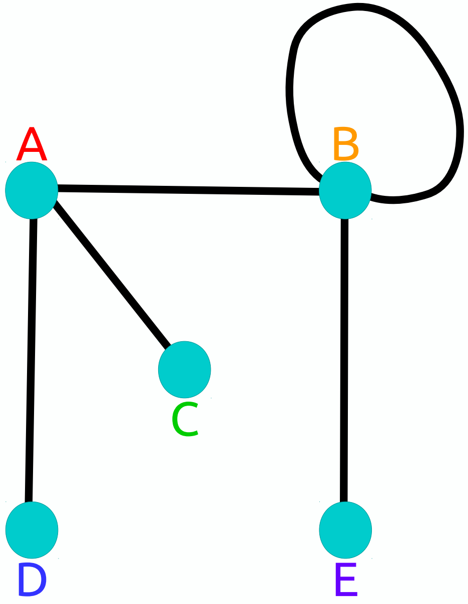
	A	B	C	D	E
A	0	1	1	1	0
B	1	1	1	0	1
C	1	1	0	0	0
D	1	0	0	0	2
E	0	1	0	2	1

Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices

When, we can look at one row at a time and begin drawing in our lines.



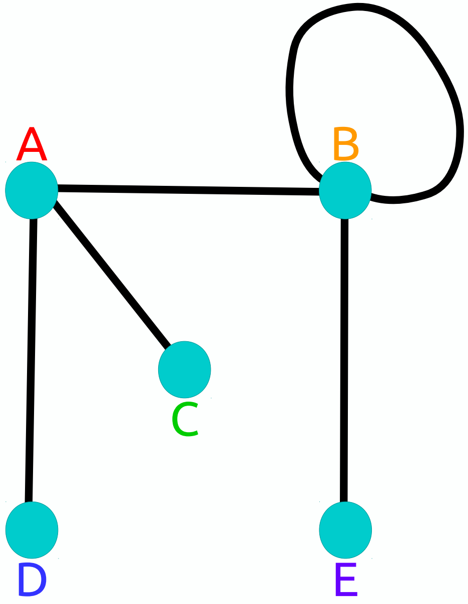
	A	B	C	D	E
A	0	1	1	1	0
B	1	1	1	0	1
C	1	1	0	0	0
D	1	0	0	0	2
E	0	1	0	2	1

Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices

When, we can look at one row at a time and begin drawing in our lines.



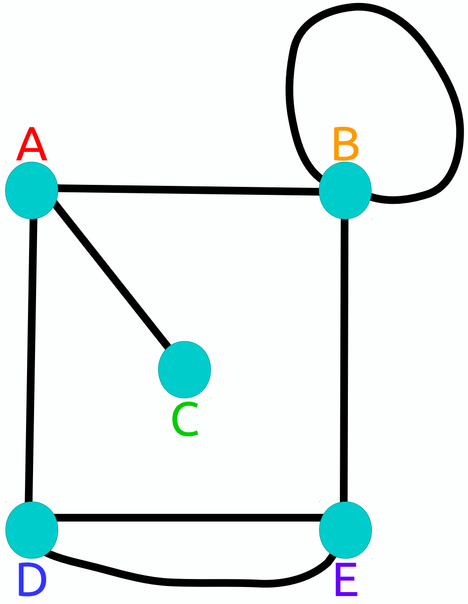
	A	B	C	D	E
A	0	1	1	1	0
B	1	1	1	0	1
C	1	1	0	0	0
D	1	0	0	0	2
E	0	1	0	2	1

Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices

When, we can look at one row at a time and begin drawing in our lines.



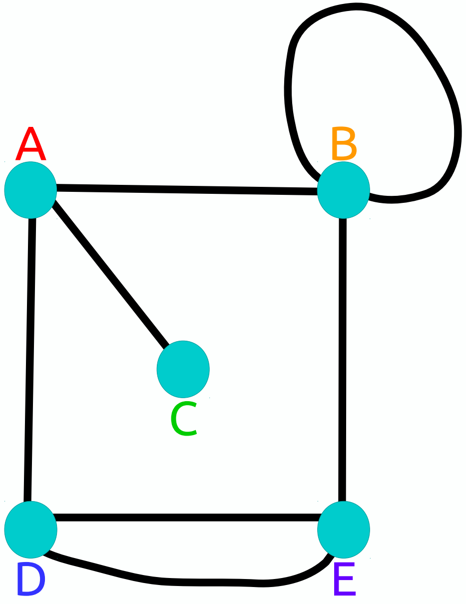
	A	B	C	D	E
A	0	1	1	1	0
B	1	1	1	0	1
C	1	1	0	0	0
D	1	0	0	0	2
E	0	1	0	2	1

Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices

When, we can look at one row at a time and begin drawing in our lines.



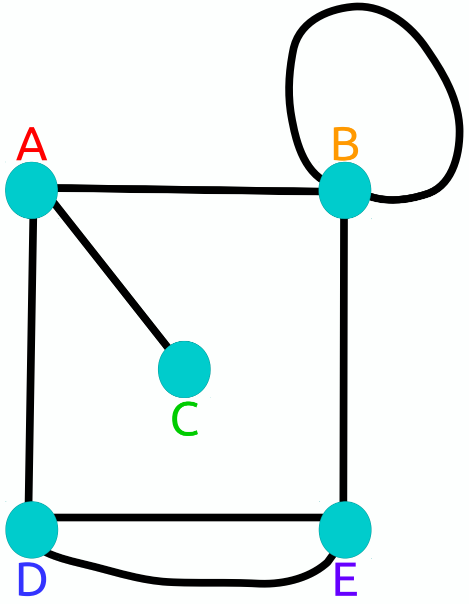
	A	B	C	D	E
A	0	1	1	1	0
B	1	1	1	0	1
C	1	1	0	0	0
D	1	0	0	0	2
E	0	1	0	2	1

Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices

Each cell represents the **amount of paths** between the node in ***row i*** and the node in ***column j***.



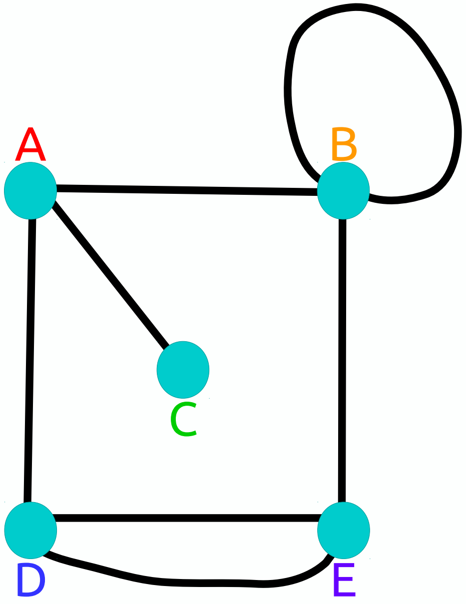
	A	B	C	D	E
A	0	1	1	1	0
B	1	1	1	0	1
C	1	1	0	0	0
D	1	0	0	0	2
E	0	1	0	2	1

Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices

Note that, with this **undirected graph**, there is a symmetry along the diagonal...



	A	B	C	D	E
A	0	1	1	1	0
B	1	1	1	0	1
C	1	1	0	0	0
D	1	0	0	0	2
E	0	1	0	2	1

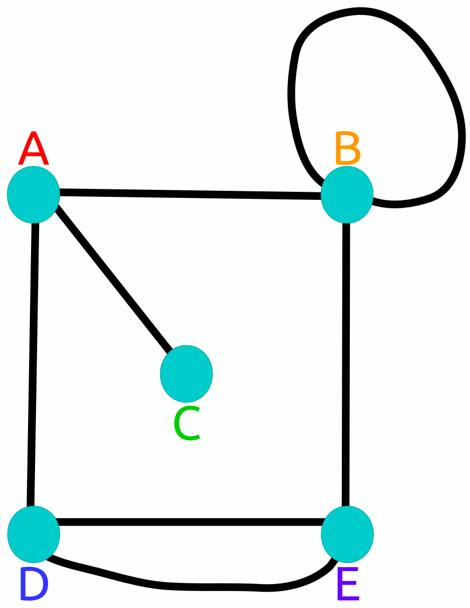
Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices

Formally, we can say:

$$M_{ij} = \text{the \# of edges that connect } v_i \text{ to } v_j \, .$$



	A	B	C	D	E
A	0	1	1	1	0
B	1	1	1	0	1
C	1	1	0	0	0
D	1	0	0	0	2
E	0	1	0	2	1

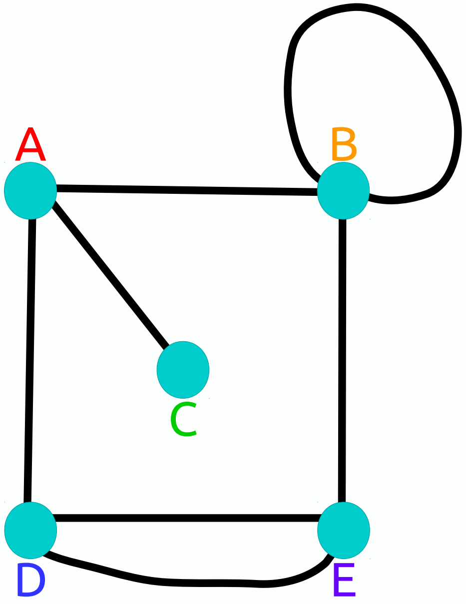
Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices

Take note:

- Two nodes that have parallel lines (e.g., D → E) will have a “2” in the cell for these – there are two edges.
- A node with a loop (e.g., B → B) will have a “1” - it is only one edge.

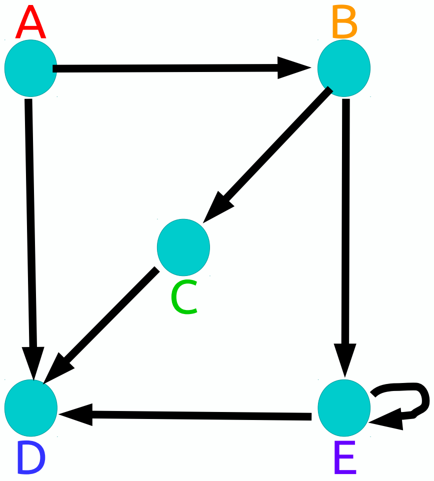


	A	B	C	D	E
A	0	1	1	1	0
B	1	1	1	0	1
C	1	1	0	0	0
D	1	0	0	0	2
E	0	1	0	2	1

Notes

We use adjacency matrices to store **edge** information between nodes.

1. Adjacency Matrices



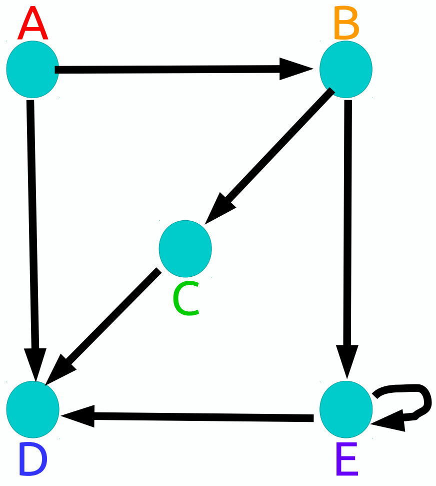
Now let's look at a **directed graph** and its adjacency matrix.

	A	B	C	D	E
A					
B					
C					
D					
E					

Notes

In a directed graph's adjacency matrix, the cell at **row i, column j** represents the amount of paths ***from i to j.***

1. Adjacency Matrices



The **rows** will be read as the “from” (starting point), and the **columns** will be read as the “to” (ending point).

There will be a “1” in the cell $\mathbf{M}_{A,B}$ to represent one arrow going from A to B.

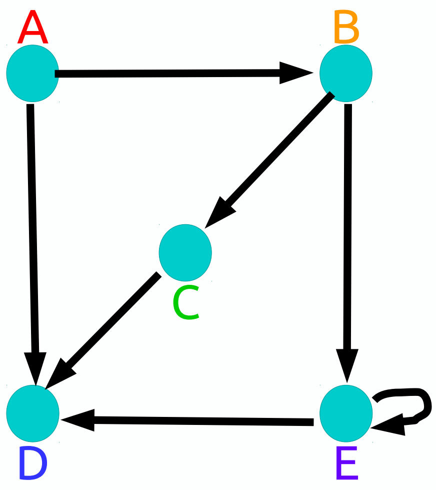
But, there will be a “0” in the cell $\mathbf{M}_{B,A}$ because no path exists going the opposite direction.

	A	B	C	D	E
A		1			
B	0				
C					
D					
E					

Notes

In a directed graph’s adjacency matrix, the cell at **row i, column j** represents the amount of paths ***from i to j.***

1. Adjacency Matrices



So let's fill out one row at a time.

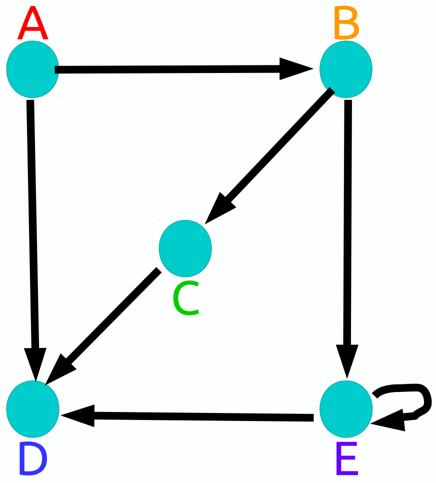
A has paths from A → B, and A → D:

	A	B	C	D	E
A	0	1	0	1	0
B	0				
C					
D					
E					

Notes

In a directed graph's adjacency matrix, the cell at **row i, column j** represents the amount of paths ***from i to j.***

1. Adjacency Matrices



So let's fill out one row at a time.

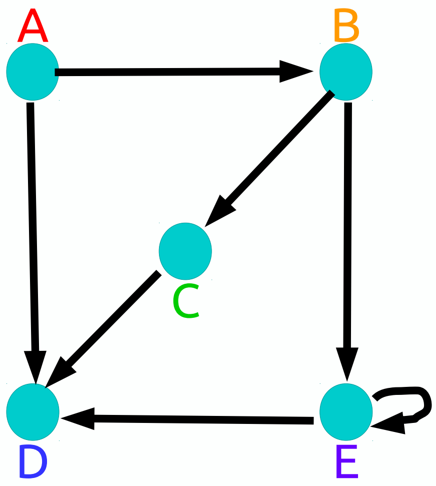
B has paths B → C and B → E

	A	B	C	D	E
A	0	1	0	1	0
B	0	0	1	0	1
C					
D					
E					

Notes

In a directed graph's adjacency matrix, the cell at **row i, column j** represents the amount of paths ***from i to j.***

1. Adjacency Matrices



So let's fill out one row at a time.

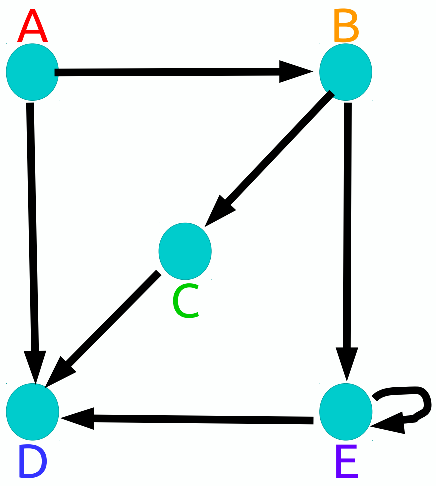
C only has one path from C → D.

	A	B	C	D	E
A	0	1	0	1	0
B	0	0	1	0	1
C	0	0	0	1	0
D					
E					

Notes

In a directed graph's adjacency matrix, the cell at **row i, column j** represents the amount of paths ***from i to j.***

1. Adjacency Matrices



So let's fill out one row at a time.

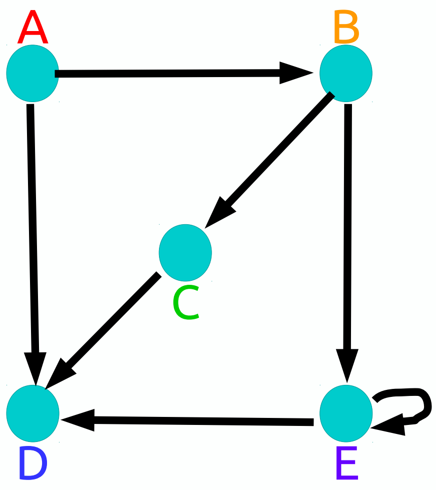
D has no outgoing paths, so the D row is all 0's.

	A	B	C	D	E
A	0	1	0	1	0
B	0	0	1	0	1
C	0	0	0	1	0
D	0	0	0	0	0
E					

Notes

In a directed graph's adjacency matrix, the cell at **row i, column j** represents the amount of paths ***from i to j.***

1. Adjacency Matrices



So let's fill out one row at a time.

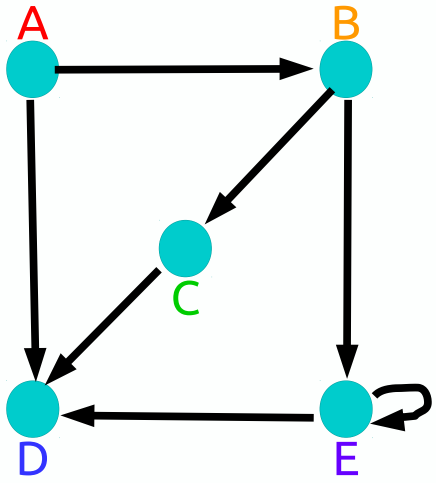
E has a path from E → D, and from E → E. This loop only counts once.

	A	B	C	D	E
A	0	1	0	1	0
B	0	0	1	0	1
C	0	0	0	1	0
D	0	0	0	0	0
E	0	0	0	1	1

Notes

In a directed graph's adjacency matrix, the cell at **row i, column j** represents the amount of paths ***from i to j.***

1. Adjacency Matrices



Notice that A has no **inputs**, so its column is all 0's.

D has no **outputs**, so its row is all 0's.

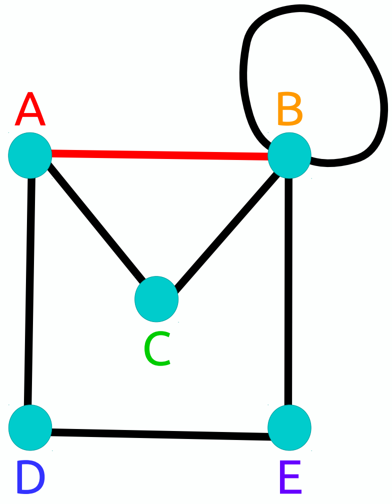
	A	B	C	D	E
A	0	1	0	1	0
B	0	0	1	0	1
C	0	0	0	1	0
D	0	0	0	0	0
E	0	0	0	1	1

Notes

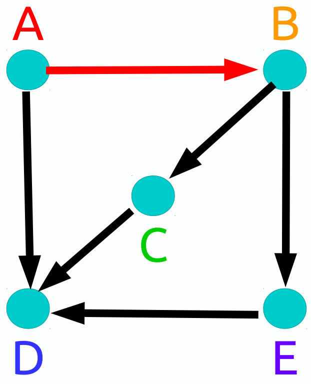
In a directed graph's adjacency matrix, the cell at **row i, column j** represents the amount of paths ***from i to j.***

1. Adjacency Matrices

Writing edges: Remember that, for an undirected graph, we specify the path between two nodes with square brackets. For directed graphs, we surround the node pair with parentheses.



[A,B]



(A,B)

Notes

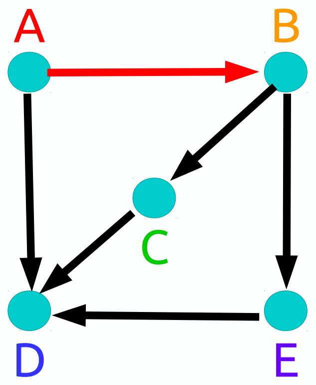
Matrix Multiplication

2. Matrix Multiplication

Theorem 1: Let M be the adjacency matrix of a directed graph G with vertex set { 1, 2, 3, …, n }. The row i, column j entry of M^k counts the number of k-step walks from node i to node j in the graph G.

The directed graph's matrix on its own represents the amount of one-edge walks between two nodes.

If we multiply the matrix to itself k times, we get a matrix for walks between nodes with k edges.



(A,B)

From Discrete Mathematics, Ensley & Crawley, page 551

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

2. Matrix Multiplication

Definition: Given matrices M and N where the number of entries in the rows of M is the same as the number of entries in the columns of N, we define the product $M \cdot N$ to mean the new matrix P so that the entry row i, column j of P is the row-column product of row i from M and column j from N. Formally, we write $P = M \cdot N$ to mean that

$$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + M_{i,3} \cdot N_{3,j} + \ldots$$

From Discrete Mathematics, Ensley & Crawley, page 484

Notes

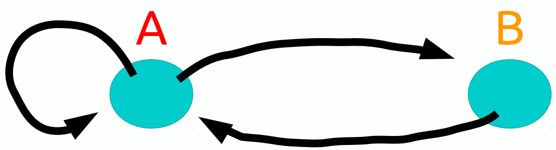
M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:
$\mathbf{P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \ldots}$

2. Matrix Multiplication

Let's look at this graph with two nodes. There are paths from A to A, A to B, and B to A, but none from B to B.



G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

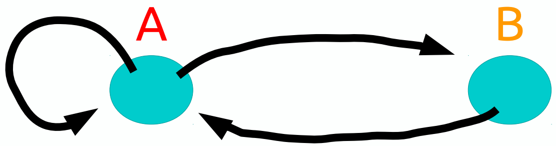
Multiplying M by N:

$\mathbf{P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \ \ldots}$

2. Matrix Multiplication

All length-1 paths of G:

1) A → A
2) A → B
3) B → A



G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

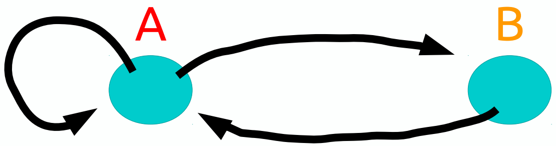
M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:

$\mathbf{P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \ \ldots}$

2. Matrix Multiplication

All length-1 paths of G:



G

1) A → A
2) A → B
3) B → A

Building out the Adjacency Matrix communicates this same information...

	A	**B**
A	1	1
B	1	0

G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

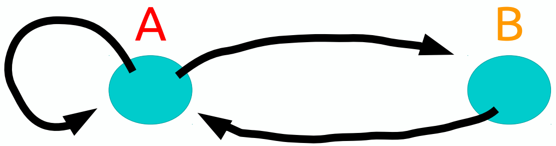
Multiplying M by N:

$$\mathbf{P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \dots}$$

2. Matrix Multiplication

All length-1 paths of G:

1) A → A
2) A → B
3) B → A



G

Building out the Adjacency Matrix communicates this same information...

- There is **one path** from A to A…

	A	B
A	1	1
B	1	0

G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

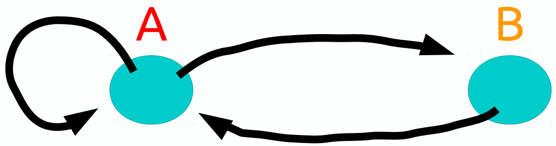
Multiplying M by N:

$$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \dots$$

2. Matrix Multiplication

All length-1 paths of G:

1) A → A
2) A → B
3) B → A



G

Building out the Adjacency Matrix communicates this same information...

- There is **one path** from A to A…
- There is **one path** from A to B…

	A	B
A	1	1
B	1	0

G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

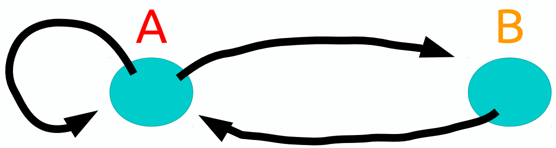
Multiplying M by N:

$$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \dots$$

2. Matrix Multiplication

All length-1 paths of G:

1) A → A
2) A → B
3) B → A



G

Building out the Adjacency Matrix communicates this same information...

- There is **one path** from A to A…
- There is **one path** from A to B…
- There is **one path** from B to A…

	A	B
A	1	1
B	1	0

G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

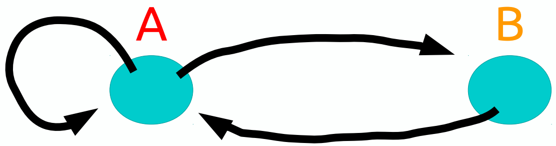
M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:

$\mathbf{P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \ldots}$

2. Matrix Multiplication

All length-1 paths of G:



G

1) A → A
2) A → B
3) B → A

Building out the Adjacency Matrix communicates this same information...

- There is **one path** from A to A…
- There is **one path** from A to B…
- There is **one path** from B to A…
- There is **no path** from B to B…

	A	B
A	1	1
B	1	0

G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

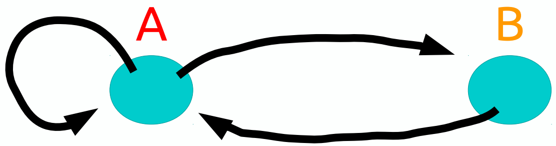
Multiplying M by N:

$$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \ldots$$

2. Matrix Multiplication

All length-1 paths of G:

1) A → A
2) A → B
3) B → A



G

The matrix for G, or G^1, indicates how many paths there are of length 1 between any two nodes.

	A	B
A	1	1
B	1	0

G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

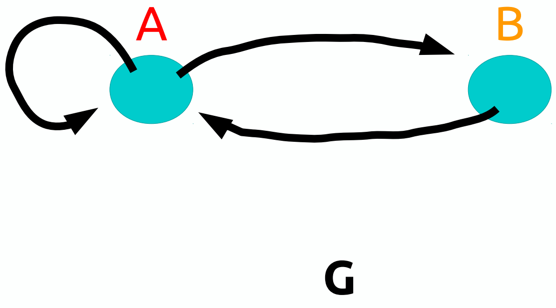
M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:

$$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \dots$$

2. Matrix Multiplication

Now, let's look at where we can go in two steps (length-2 paths):



1) A → A → A
2) A → B → A
3) A → A → B
4) B → A → A
5) B → A → B

Notes

M: The matrix showing the # of length-1 paths between two nodes.

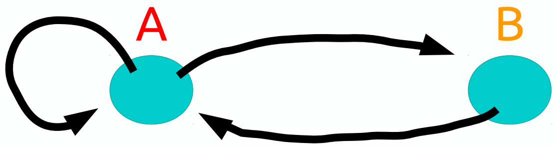
M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:

$$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \dots$$

2. Matrix Multiplication

Now, let's look at where we can go in two steps (length-2 paths):



G

1) A → A → A   From A to A
2) A → B → A   From A to A
3) A → A → B   From A to B
4) B → A → A   From B to A
5) B → A → B   From B to B

If we look at the endpoints, we can see that there are **two ways** to get from A to A **(in two steps)**, and each other path (A→B, B→A, B→B) only has one way **(in two steps)**.

Notes

M: The matrix showing the # of length-1 paths between two nodes.

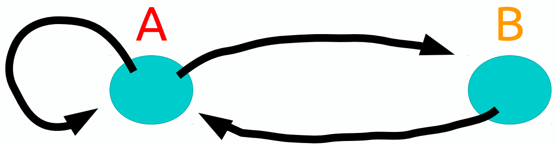
M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:

$$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \dots$$

2. Matrix Multiplication

Now, let's look at where we can go in two steps (length-2 paths):



G

1) A → A → A From A to A
2) A → B → A From A to A
3) A → A → B From A to B
4) B → A → A From B to A
5) B → A → B From B to B

If we multiply our G matrix to itself, getting G^2, we will get the adjacency matrix with this information – How many paths are there between any two nodes, **in two steps?**

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:
$\mathbf{P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \dots}$

2. Matrix Multiplication

Example: Calculate A · B

	A	B
A	1	1
B	1	0

G

·

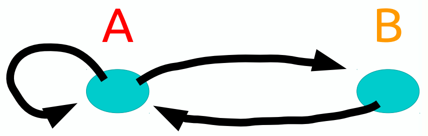
	A	B
A	1	1
B	1	0

G

=

	A	B
A		
B		

G^2



G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:

$$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \ldots$$

2. Matrix Multiplication

Example: Calculate A · B

	A	B
A	1	1
B	1	0

G

·

	A	B
A	1	1
B	1	0

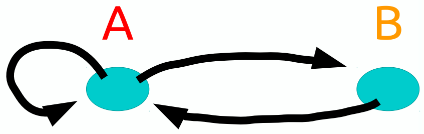
G

=

	A	B
A	2	
B		

$\mathbf{G^2}$

$$G^2_{1,1} = G_{1,1} \cdot G_{1,1} + G_{1,2} \cdot G_{2,1}$$
$$= 1 \cdot 1 + 1 \cdot 1$$
$$= 2$$



G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:
$\mathbf{P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \ldots}$

2. Matrix Multiplication

Example: Calculate A · B

	A	B
A	1	1
B	1	0

G

·

	A	B
A	1	1
B	1	0

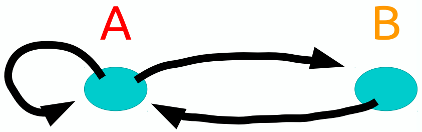
G

=

	A	B
A	2	1
B		

G^2

$$G^2_{1,2} = G_{1,1} \cdot G_{1,2} + G_{1,2} \cdot G_{2,2}$$
$$= 1 \cdot 1 + 1 \cdot 0$$
$$= 1$$



G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:

$$\mathbf{P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \ldots}$$

2. Matrix Multiplication

Example: Calculate A · B

	A	B
A	1	1
B	1	0

G

·

	A	B
A	1	1
B	1	0

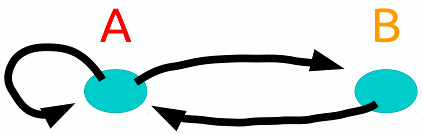
G

=

	A	B
A	2	1
B	1	

G^2

$$
\begin{aligned}
G^2_{2,1} &= G_{2,1} \cdot G_{1,1} + G_{2,2} \cdot G_{2,1} \\
&= 1 \cdot 1 + 0 \cdot 1 \\
&= 1
\end{aligned}
$$



G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:
$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \ldots$

2. Matrix Multiplication

Example: Calculate A · B

	A	B
A	1	1
B	1	0

G

·

	A	B
A	1	1
B	1	0

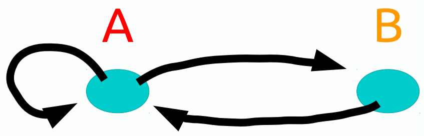
G

=

	A	B
A	2	1
B	1	1

G^2

$$G^2_{2,2} = G_{2,1} \cdot G_{1,2} + G_{2,2} \cdot G_{2,1}$$
$$= 1 \cdot 1 + 0 \cdot 0$$
$$= 1$$



G

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:
$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \ldots$

2. Matrix Multiplication

Example: Calculate A · B

	A	B
A	1	1
B	1	0

G

·

	A	B
A	1	1
B	1	0

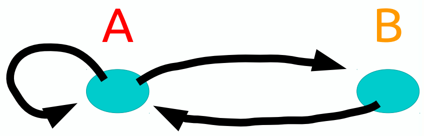
G

=

	A	B
A	2	1
B	1	1

G^2

So while *G* shows the paths we can go between two nodes in one “move”, G^2 shows us the paths we can go between two nodes in two “moves”.



G

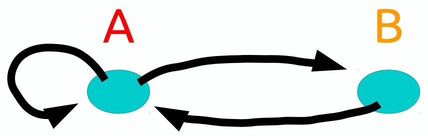
Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:
$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \ldots$

2. Matrix Multiplication



G

And as we continue multiplying the adjacency matrix, we will get G^k, the amount of k-length paths between any two points.

Length-1 paths:

	A	B
A	1	1
B	1	0

G

1) A → A
2) A → B
3) B → A

Length-2 paths:

	A	B
A	2	1
B	1	0

G^2

1) A → A → A — From A to A
2) A → B → A — From A to A
3) A → A → B — From A to B
4) B → A → A — From B to A
5) B → A → B — From B to B

Notes

M: The matrix showing the # of length-1 paths between two nodes.

M^2: The matrix showing the # of length-2 paths between two nodes.

Multiplying M by N:

$$P_{i,j} = M_{i,1} \cdot N_{1,j} + M_{i,2} \cdot N_{2,j} + \dots$$

Conclusion

Matrices can be really interesting to work with, but doing the multiplication by hand is tedious. Luckily, computers are really good at working with matrices. ;)