EE1004

Fndt'n of IS & Data Anlys

Assessments:

- Examination : 50%
- Tests and quizzes : 30% (Two Test:20% and 3-4 quizzes:10%)
- Assignments: 20 % (3-4 Assignments: 8 % and 3 Labs: 12%)

Lab: Week 6, Week 9, week 12

Reference:

Stephen Boyd and Lieven Vandenberghe, Introduction to Applied Linear Algebra Vectors, Matrices, and Least Squares

To pass the course, at least 30% in Exam and at least 30% in CourseWork.

(Note that if total is still 3x%, you may fail in the subject. Better score is 40+)

Chapter One

- What is data?
- What is big data?
- The ways to represent data in engineering
- Graph

What are data?

- Data **are** raw facts and figures that on their own have no meaning
- Data are gathered or captured from activity of people, places and things.
- These can be any alphanumeric characters i.e. text, numbers, symbols

What is data?

- Yes, Yes, No, Yes, No, Yes, No, Yes
- 42, 63, 96, 74, 56, 86
- 111192, 111234
- None of the above have any meaning until they are given a **CONTEXT** and **PROCESSED** into a usable form

What is the value of 1100 ??

The correct answer is “I **do not know”** because no context is given.

You want to be good in study. Please do not be loose-loose.

Context

Example 1: My Dog

Qualitative:

- He is white and brown
- He has short hair
- His breed is corgi

Quantitative:

- Discrete:
    - Legs : 4
    - Eyes : 2
    - He has 2 sisters
- Continuous:
    - He weighs 5.2 kg
    - He is 20 cm

Context

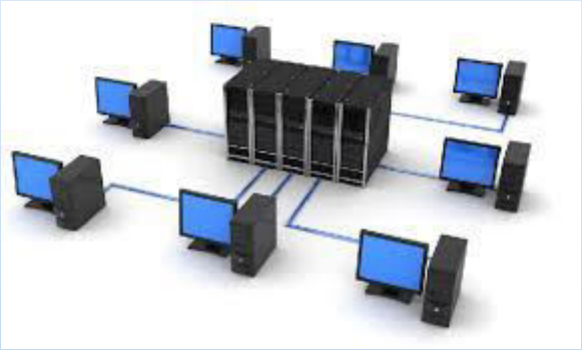
Example 2: My network

Qualitative:

- Color of desktop : black
- Network protocol: TCP
- Structure : Star

Quantitative:

- Discrete:
    - Number of desktop : 8
    - Servers : 1
- Continuous:
    - Power consumption: 20.28 KW
    - Total Cable length: 200.25 m



Qualitative and Quantitative

Qualitative Data	Quantitative Data
• Deals with descriptions • Data can be observed • Colors, texture type, smells, beauty, etc.	• Result of counting or measuring attributes of a population. • Deals with numbers • Data which can be measured. • Length, height, area, power consumption

Qualitative and Quantitative

- Qualitative is a categorical measurement (or observation) not in terms of number.
- Through a natural language description.

Examples:

- Hair color of my dog is brown and white
- Breed of my dog is corgi
- Structure of my network is star

Qualitative and Quantitative

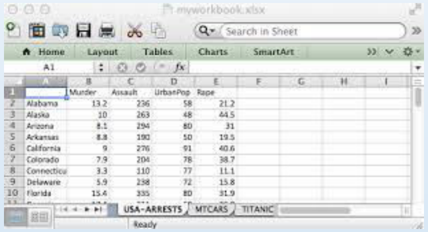
- Quantitative data a numerical measurement not in terms of natural language description
- Through numbers

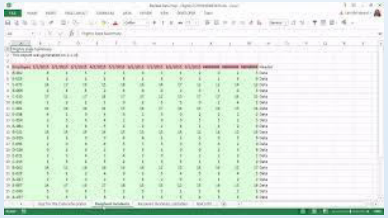
Example:

- My income is HKD 20,220.
- Your GPA is 3.40.
- I have two dogs
- The power consumption of my computer is 2kW.

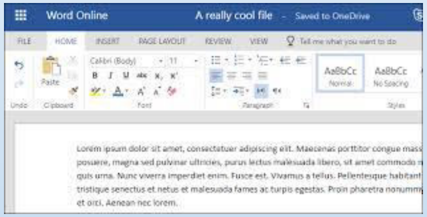
Data examples:

Excel Files:



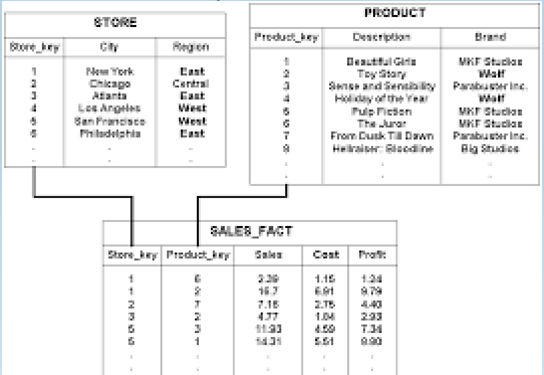


Word Files:



Images

Tables in a database system



STORE

Store_key	City	Region
1	New York	East
2	Chicago	Central
3	Atlanta	East
4	Los Angeles	West
5	San Francisco	West
6	Philadelphia	East
.	.	.

PRODUCT

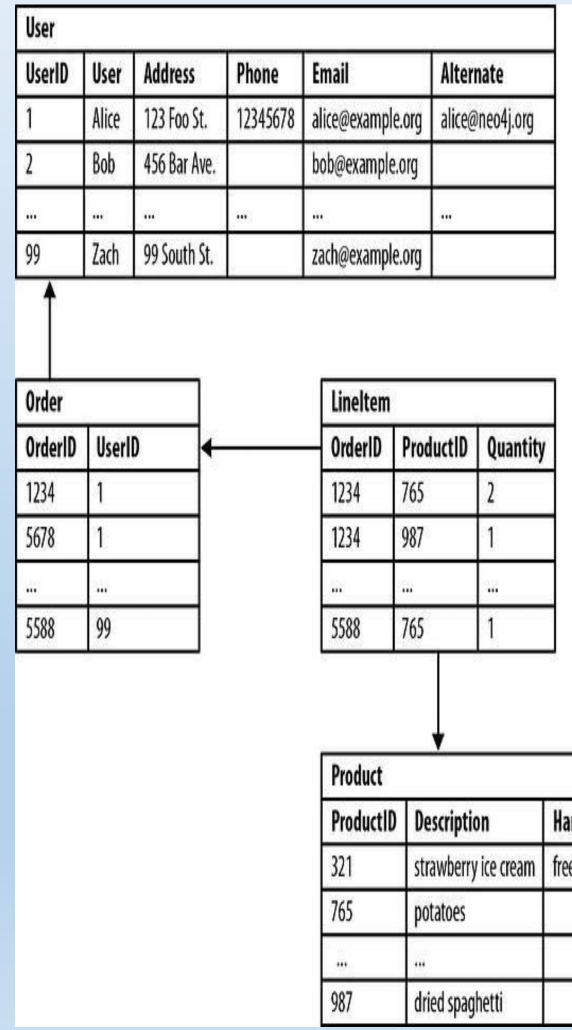
Product_key	Description	Brand
1	Beautiful Girls	MKF Studios
2	Toy Story	Wolf
3	Sense and Sensibility	Parabuster Inc.
4	Holiday of the Year	Wolf
5	Pulp Fiction	MKF Studios
6	The Juror	MKF Studios
7	From Dusk Till Dawn	Parabuster Inc.
8	Hellraiser: Bloodline	Big Studios
.	.	.

SALES_FACT

Store_key	Product_key	Sales	Cost	Profit
1	6	2.39	1.15	1.24
1	2	16.7	6.91	9.79
2	7	7.16	2.75	4.40
3	2	4.77	1.04	2.93
5	3	11.93	4.59	7.34
5	1	14.31	5.51	8.80
.	.	.	.	.

Structured Data and Unstructured Data

- Structured data is the type of data most of us are used to working with. Think of data that fits neatly within fixed fields and columns in relational databases and spreadsheets. Hence structured data the patterns of structured data make them easily searchable.
- Fields store length-delineated data phone numbers, Social Security numbers, or ZIP codes. Even text strings of variable length like names are contained in records, making it a simple matter to search.
- Note that in a database, the format of a field is well defined.
- Data may be human- or machine-generated.
- Their formats allow us to search with human generated queries and via algorithms using type of data and field names, such as alphabetical or numeric, currency or date.



User

UserID	User	Address	Phone	Email	Alternate
1	Alice	123 Foo St.	12345678	alice@example.org	alice@neo4j.org
2	Bob	456 Bar Ave.		bob@example.org	
...	...	...	...	...	...
99	Zach	99 South St.		zach@example.org	

Order

OrderID	UserID
1234	1
5678	1
...	...
5588	99

LineItem

OrderID	ProductID	Quantity
1234	765	2
1234	987	1
...	...	...
5588	765	1

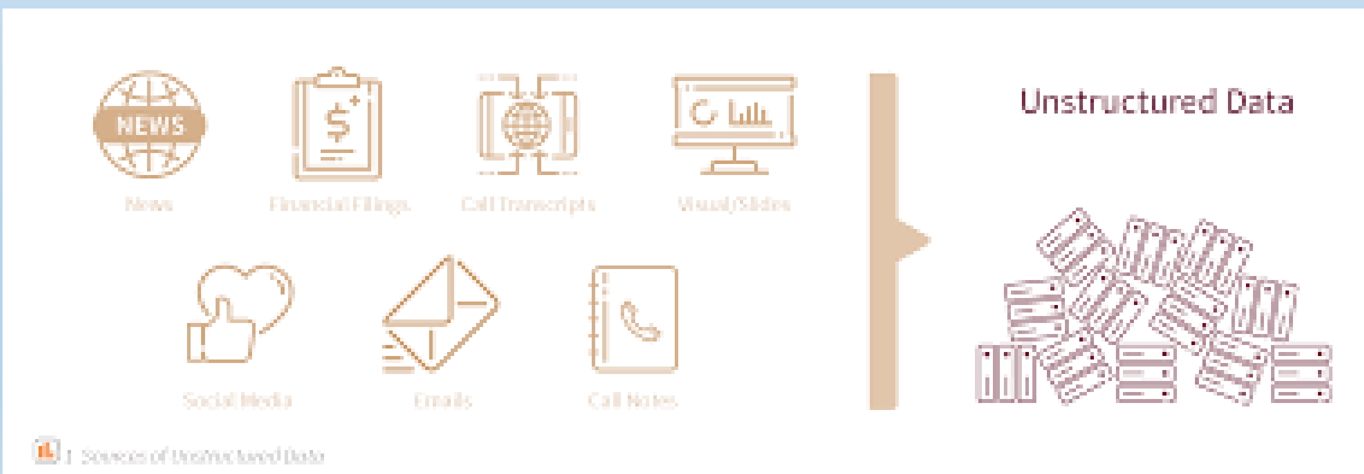
Product

ProductID	Description	Ha
321	strawberry ice cream	fre
765	potatoes	
...	...	
987	dried spaghetti	

Structured Data and Unstructured Data

Unstructured data is “everything else” – is comprised of data that is usually not as easily searchable, such as a collection of audio files, video files, and social media postings, formats of data should be different.

Unstructured data has internal structure but is not structured via pre-defined data models or schema. It may be textual or non-textual, and human- or machine-generated. It may also be stored within a non-relational database like NoSQL.



Structured Data and Unstructured Data

Typical human-generated unstructured data includes:

Text files: Word processing, spreadsheets, presentations, email, logs.

Emails: Email has some internal structure thanks to its metadata, and we sometimes refer to it as semi-structured. However, its message field is unstructured and traditional analytics tools cannot parse it.

Social Media files: Data from Facebook, Twitter, LinkedIn.

Websites: YouTube, Instagram, photo sharing sites.

Typical machine-generated unstructured data includes:

Satellite imagery: Weather data, land forms, military movements.

Scientific data: Oil and gas exploration, space exploration, seismic imagery, atmospheric data.

Sensor data: Traffic, weather, oceanographic sensors.

Big Data

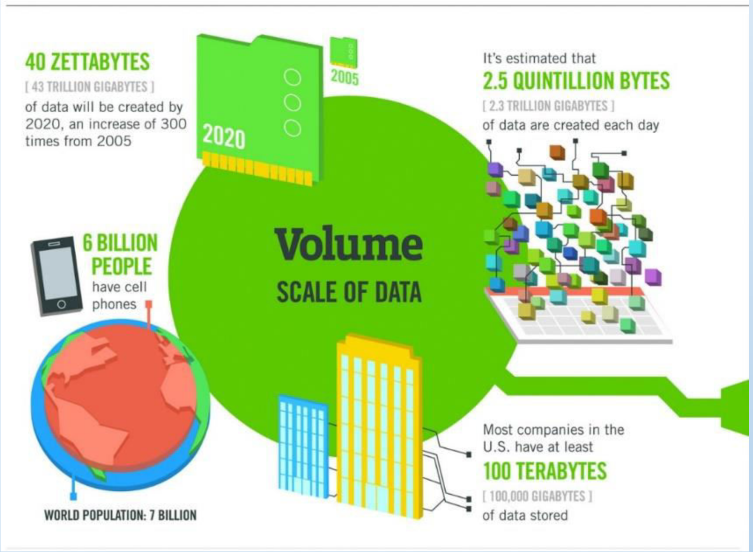
Big Data is typically characterized by the following four V's:

Volume:

- The amount and scale of data being created every day is vast compared to traditional data sources we had in the past.

  facebook posts, and Youtube

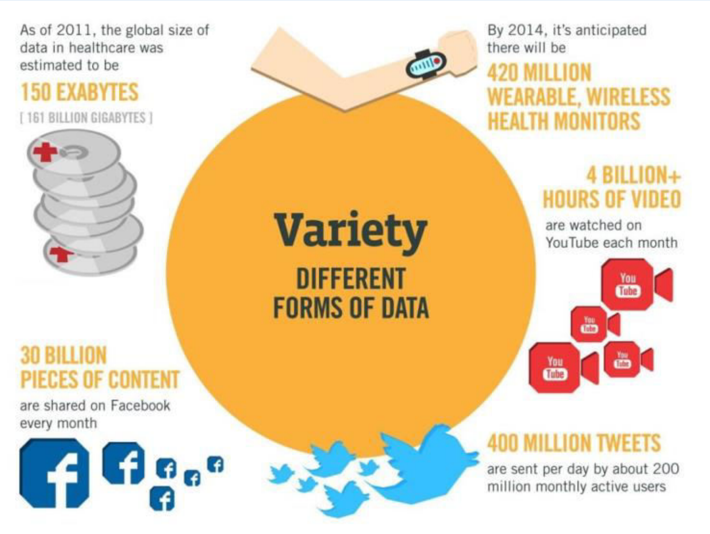
- Usually greater than terabytes and petabytes
- The data sets in Big Data are too large to process with a regular laptop or desktop processor.



Big Data

Variety

- The data comes from different sources in different structures and forms. Data are created not only by human but also by machines.
- Three types: structured, semi structured and unstructured data.
- The variety in data types implies that we need particular processing capabilities and specialist algorithms.



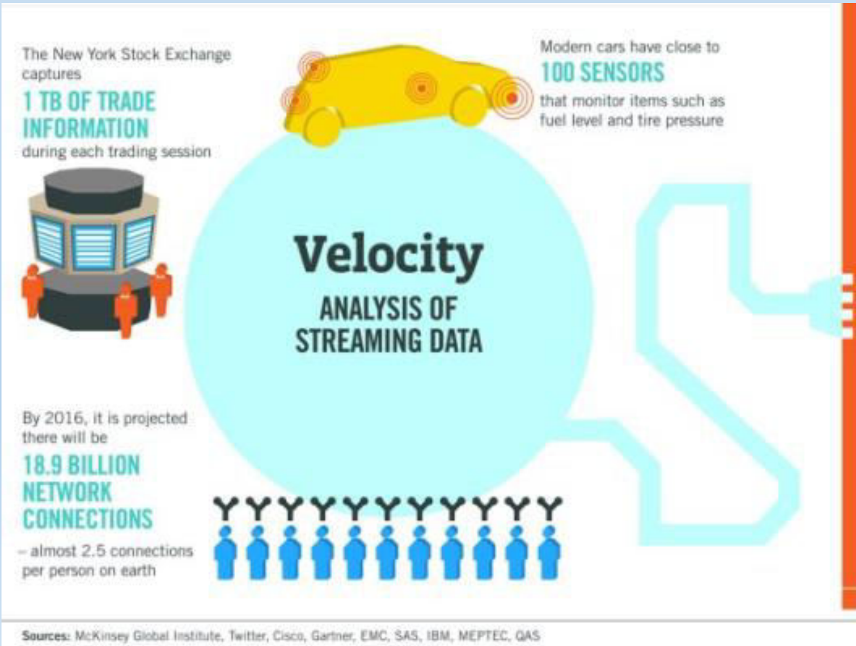
CCTV audio and video files at various locations in a city.

Facebook images and videos

Big Data

Velocity:

- Data are generated extremely fast, the process that never stops even while we sleep. Storing them never stops too.
- Single instance of high-velocity data is Twitter where over 350,000 tweets are now sent worldwide per minute, equating to 500 million tweets per day.



Big Data (Quality of Data)

Veracity:

- Big Data is sources from many different places.
- We need to test the quality of the data.
- High veracity data are valuable to analyze and that contribute in a meaningful way to the overall results.
- Low veracity data contains a high percentage of meaningless data.

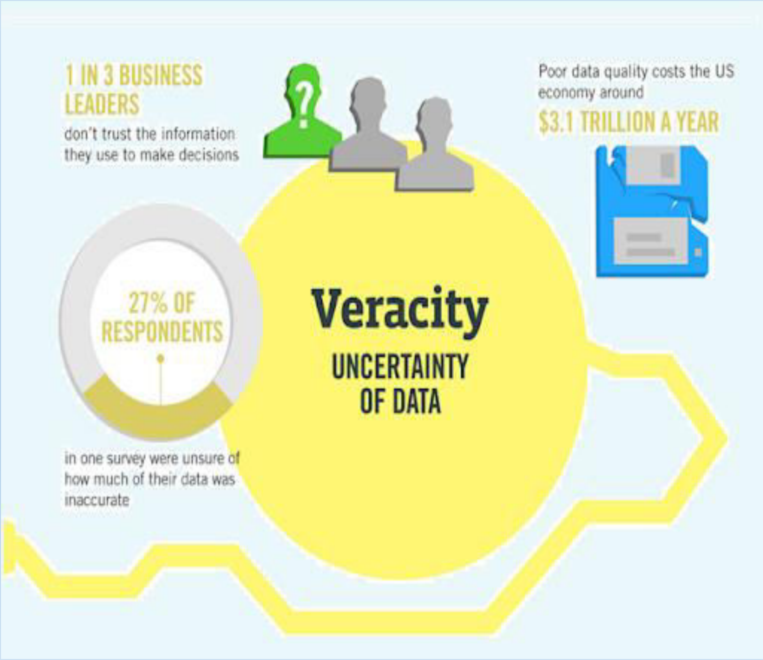
Also,

The bus GPS data

(about ten minutes ago)=> quality ??

The realtime bus GPS data

may not be usable in some situations??



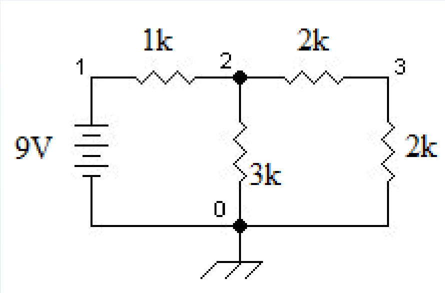
Representation of Data in EE

Circuits:

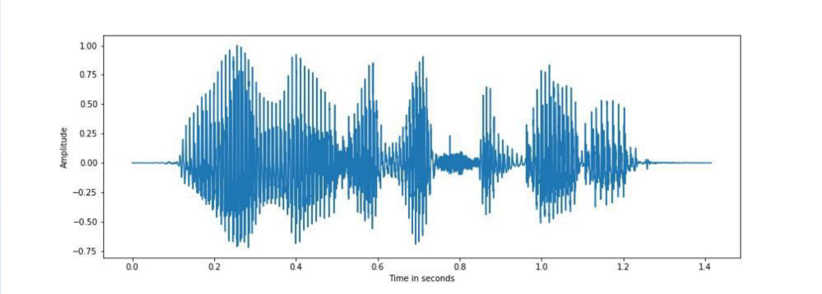
$$9 = 1000 \times i_1 + 4000 \times i_2$$

$$9 = 1000 \times i_1 + 3000 \times i_3$$

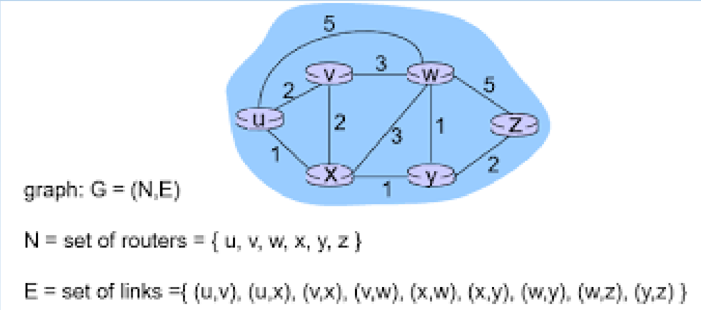
$$0 = i_1 - i_2 - i_3$$



Signals: Speech signal, image

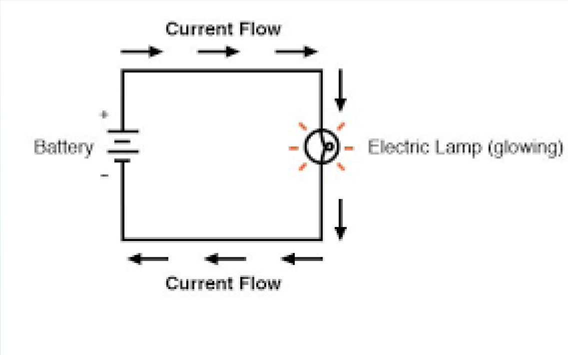


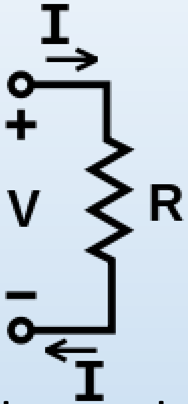
Network:



Representation of Data in EE

Circuit:





Ohm's law :

Ohm's law states that the current through a conductor between two points is directly proportional to the voltage across the two points.

Voltage=Current × Resistance

$$V=IR$$

Voltage: electric potential difference.

Current: the rate of flow of electric charge (electrons) past a point.

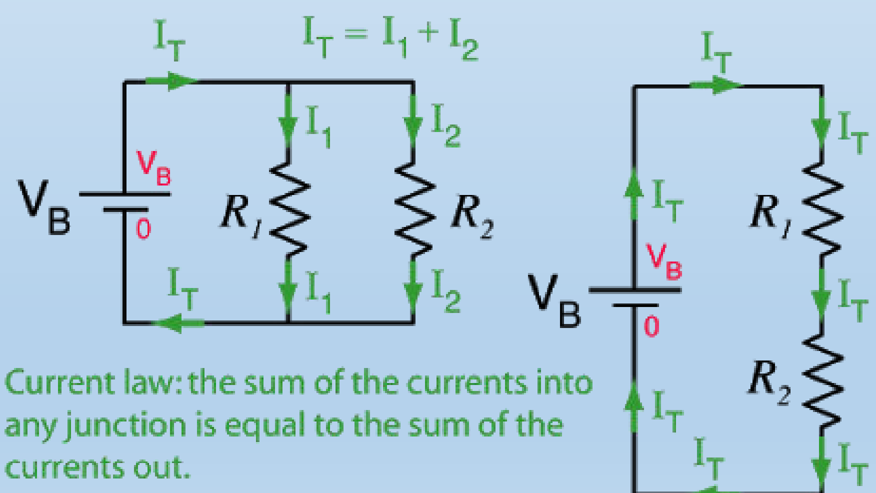
Resistance: a measure of its opposition to the flow of electric current.

Representation of Data in EE

Circuit:

Notations: Circuit Diagrams

- Node: Point of connection between elements.
- Branch : Connection between two nodes.
- Loop: Closed loops in circuit diagram.



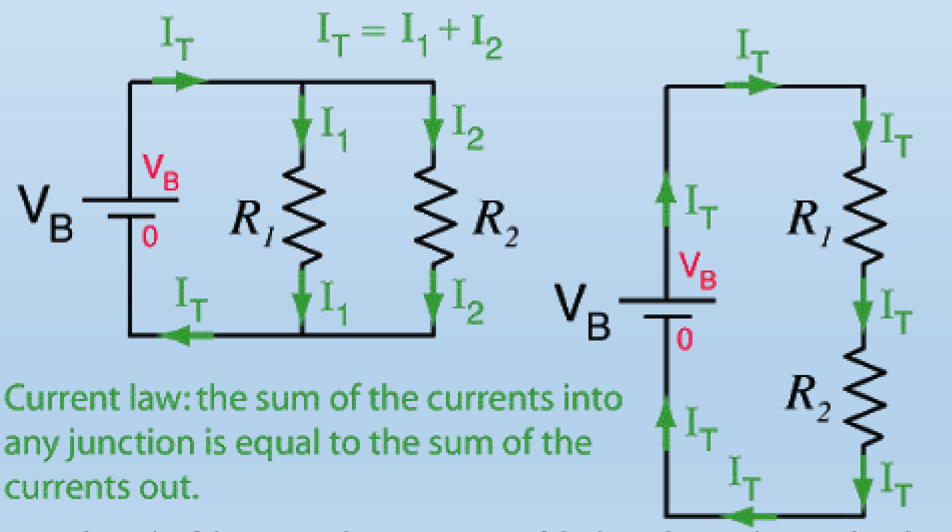
For any branch of the circuit, the current out of the branch must be equal to the current into the branch. This is required by the conservation of electric charge. Any cross-section of the circuit must carry the total current. For a series circuit, the current is the same at any point in the circuit.

Representation of Data in EE

Kirchhoff's Current Law

Algebraic sum of currents entering a closed a node (or a loop) is zero. (note that currents have directions)

$$\sum i_n = 0$$



For any branch of the circuit, the current out of the branch must be equal to the current into the branch. This is required by the conservation of electric charge. Any cross-section of the circuit must carry the total current. For a series circuit, the current is the same at any point in the circuit.

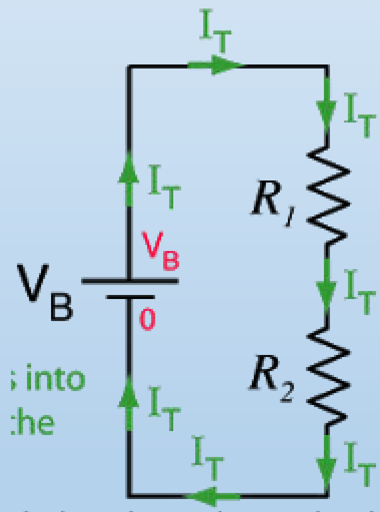
Representation of Data in EE

Kirchhoff's Voltage Law

Algebraic sum of all voltages around a closed path (or loop) is zero.

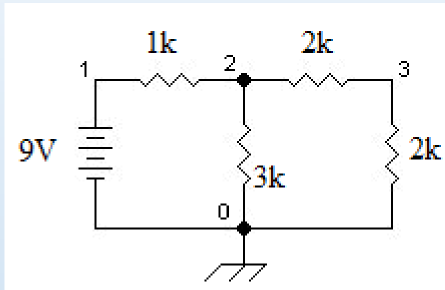
$$\sum V_n = 0$$

$$V_B - I_T R_1 - I_T R_2 = 0$$



Representation of Data in EE

Circuits:



$$9 = 1000 \times i_1 + 4000 \times i_2$$

$$9 = 1000 \times i_1 + 3000 \times i_3$$

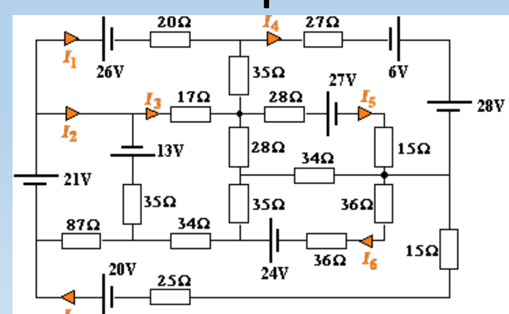
$$0 = i_1 - i_2 - i_3$$

Many students may set up inappropriate equations. three cases: wrong equations, not enough equations, some equations are not linear independent

We can say that the set of **linear** equations represent the circuit .

Of course, we can solve the above set of **linear** equations by hand.

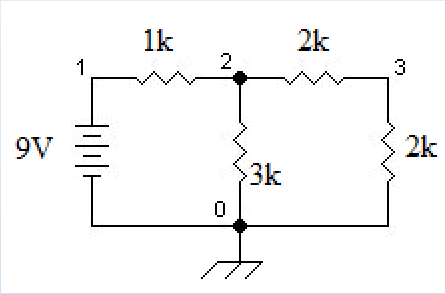
However, complicated circuit???



Representation of Data in EE

Circuits:

$$9 = 1000 \times i_1 + 4000 \times i_2$$
$$9 = 1000 \times i_1 + 3000 \times i_3$$
$$0 = i_1 - i_2 - i_3$$



In vector-matrix form:

$$\begin{pmatrix} 9 \\ 9 \\ 0 \end{pmatrix} = \begin{pmatrix} 1000 & 4000 & 0 \\ 1000 & 0 & 3000 \\ 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix}$$

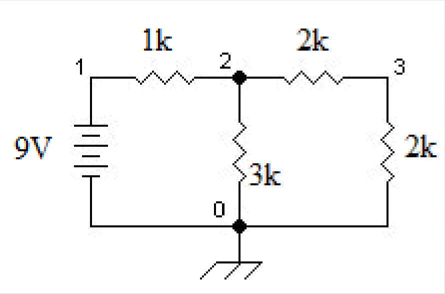
We can say that above vector-matrix equation represents the circuit .

$$\begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} = \begin{pmatrix} 1000 & 4000 & 0 \\ 1000 & 0 & 3000 \\ 1 & -1 & -1 \end{pmatrix}^{-1} \begin{pmatrix} 9 \\ 9 \\ 0 \end{pmatrix}$$

Representation of Data in EE

Not only represent a circuit with a particular set values.

But also, a particular structure.



$$V_B = R_1 \times i_1 + (R_{2a} + R_{2b}) \times i_2$$

$$V_B = R_1 \times i_1 + R_3 \times i_3$$

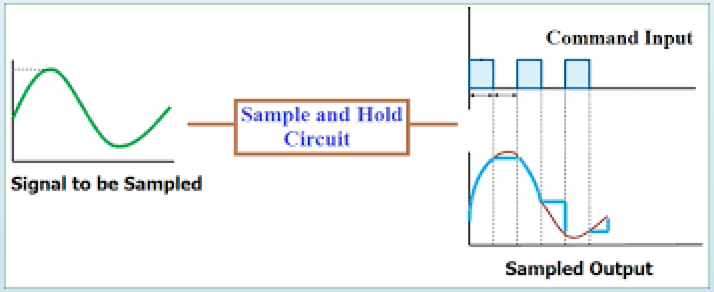
$$0 = i_1 - i_2 - i_3$$

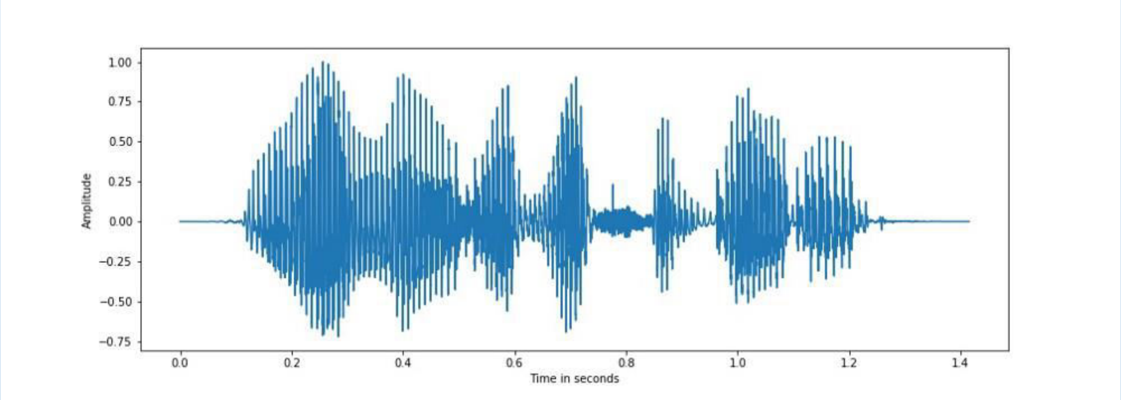
$$\begin{pmatrix} V_B \\ V_B \\ 0 \end{pmatrix} = \begin{pmatrix} R_1 & R_{2a} + R_{2b} & 0 \\ R_1 & 0 & R_3 \\ 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix}$$

$$\begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} = \begin{pmatrix} R_1 & R_{2a} + R_{2b} & 0 \\ R_1 & 0 & R_3 \\ 1 & -1 & -1 \end{pmatrix}^{-1} \begin{pmatrix} V_B \\ V_B \\ 0 \end{pmatrix}$$

Representation of Data in EE

Speech



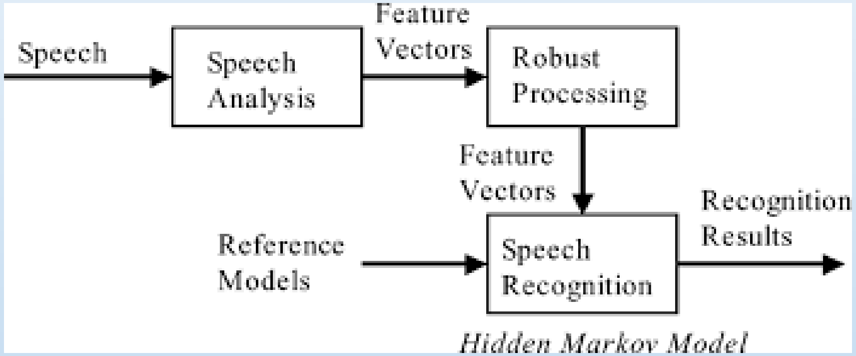


It can be seen from the above figure that speech can be represented as a variation of amplitude with time, saying a sequence of real numbers, $\{a(1), a(2), \dots \dots\}$.

The amplitude is normalised such that the maximum value is *1*.

Representation of Data in EE

Speech recognition

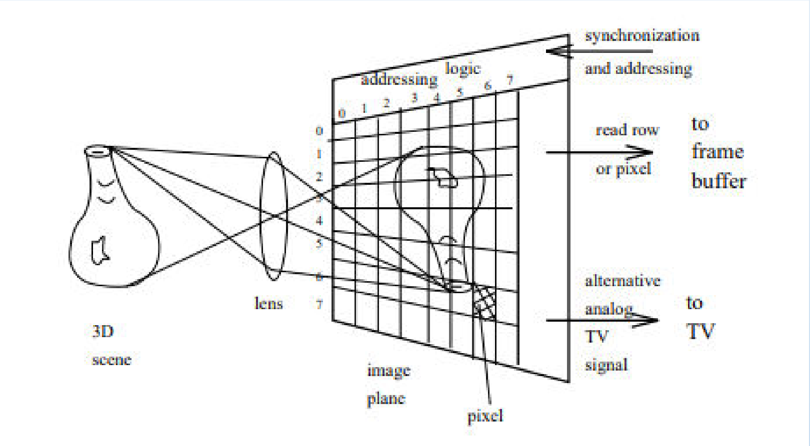


Convert the sequence to a number of features. The collection of features from a feature vector.

In Speech Analysis, Robust Processing, Speech Recognition, a lot mathematics skills are involved, such as
linear algebra (matrix and vector), and probability.

Representation of Data in EE

Image



- A grey scale image is a monochrome digital image Im[indx,indy] with one intensity value per pixel. The indices indx and indy indicate the positions of pixels.
- A color image is a 2D image M[indx,indy,c], i.e., 2D array of 3D vectors, where c=1 to 3. Or say 3D array.
- In image processing and pattern recognition, we use a lot of mathematics tools in linear algebra to handle those arrays.

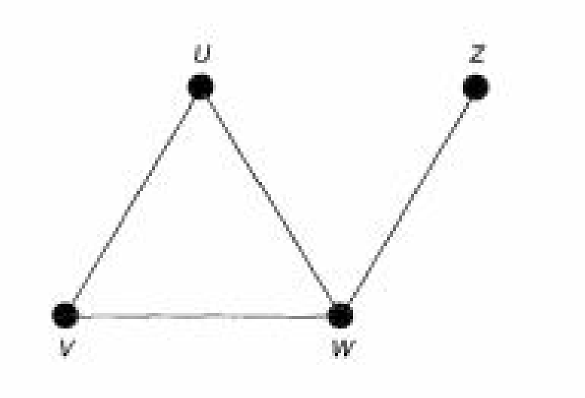
Graph

A simple graph G={V,E} consists of a non-empty finite set V(G) of elements called vertices (or nodes), and a finite set E(G) of distinct unordered pairs of distinct elements of V(G) called edges.

V(G): vertex set

E(G): the edge set.

An edge {v, w} is said to join the vertices v and w, and is usually abbreviated to vw.



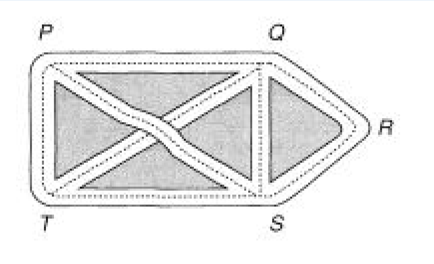
Vertex set = {w, v, w, z},

Edge set = {uv, uw, vw, wz}

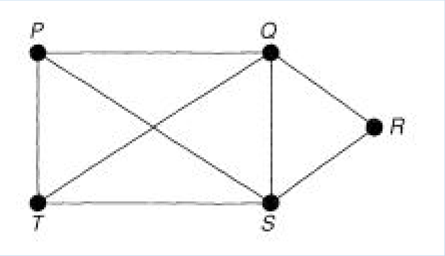
Brief Introduction of Graph

Representation example:

A road map :



The junction points P, Q, R, S and T => vertices, the roads => edges, and the whole diagram is called a graph.

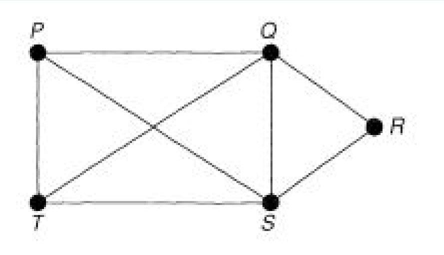


Vertex set = {P, Q, R, S,T},

Edge set = { PQ, QR, RS, ST, PT, PS, QT, QS}

Note that the intersection of the lines PS and QT is not a vertex, since it does not correspond to a cross-roads

Brief Introduction of Graph (order and size)



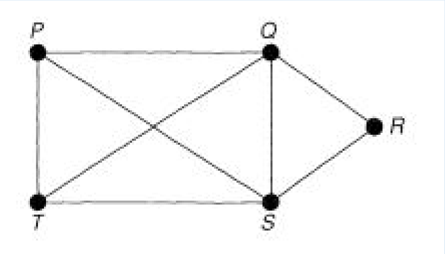
- Given a G={V,E}
- The order of G, denoted by $|G|$, is the number of vertices of G, i.e., $|G| = |V|$.

  Here $|\cdot|$ denotes the number of elements in a set.
- The size of G, denoted by $||G||$, is the number of edges of G, i.e., $||G|| = |E|$.
- Note that if the order of G is n, then the size of G is

  between 0 and $C_2^n = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2}$
- In our example, $|G| = 5$ and $\|G\| = 8$

Brief Introduction of Graph (degree)

The degree of a vertex is the number of edges with that vertex as an end-point;

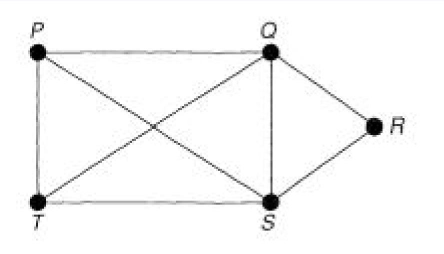


The degree of Q=4.

The degree of R= ?

:

Brief Introduction of Graph (neighbourhood)



- Given a graph $G = \{V, E\}$
- The set of all neighbors of a vertex v of $G = \{V, E\}$, denoted by $N(v)$, is called the neighborhood of v.
- The above definition does not include v itself, and is more specifically the **open neighbourhood** of v.
- We can define a neighbourhood in which v itself is included. In this case, we the **closed neighbourhood**.
- If A is a subset of V, we denote by $N(A)$ the set of all vertices in G that are adjacent to at least one vertex in A.

Open neighbourhood of R are $\{Q, S\}$

Closed neighbourhood of R are $\{R, Q, S\}$

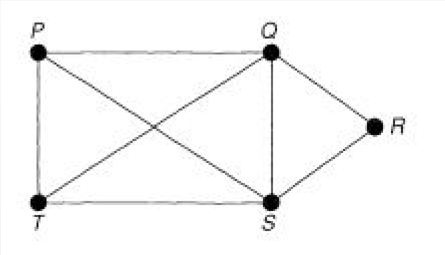
Let $A = \{Q, R\}$

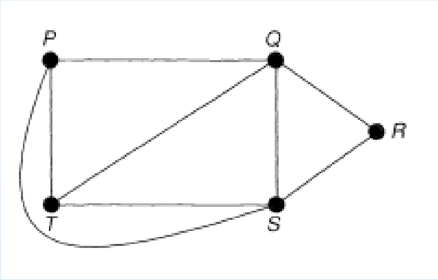
Open neighbourhood of A are $\{P, T, S\}$

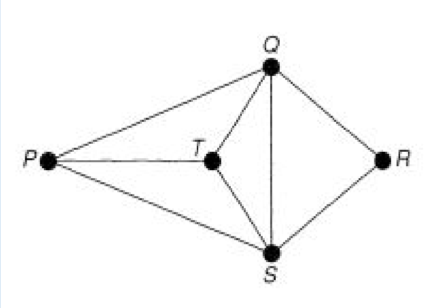
Closed neighbourhood of are $\{P, S, T, Q, R\}$

Brief Introduction of Graph (identical and isomorphic)

- Remove the 'crossing' of the lines PS and QT by drawing the line PS outside the rectangle PQST.
- The resulting graph still tells us whether there is a direct road from one intersection to another.
- A graph is a representation of a set of points and of how they are joined up.
- The way that we draw is irrelevant.
- All the graphs are the same.

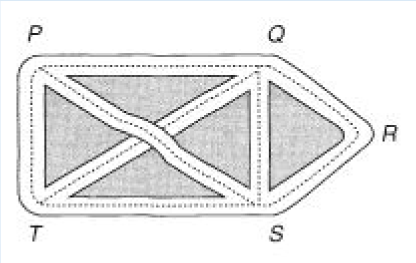


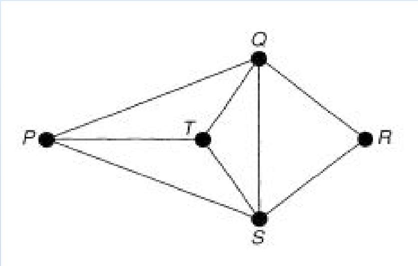


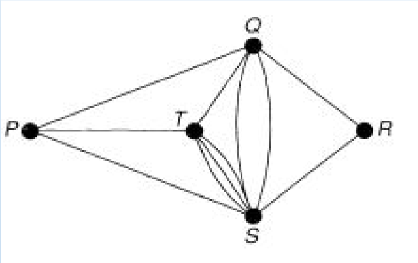


Brief Introduction of Graph (multiple edges and loop)

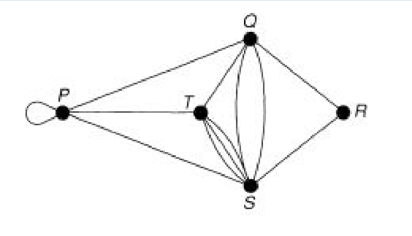
- Suppose that the roads joining *Q* and S, and *S* and *T,* have too much traffic to carry.
- Build extra roads joining these points.
- The edges joining *Q* and *S,* or *S* and *T,* are called **multiple edges.**







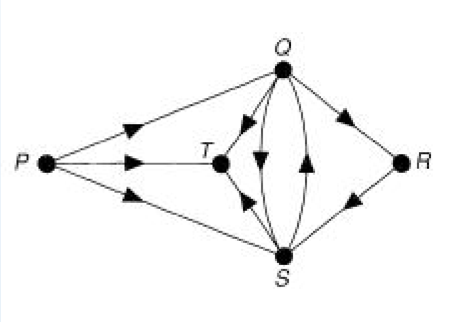
- Now we need a car park at P, then we indicate this by drawing an edge from *P* to itself, called a **loop.**



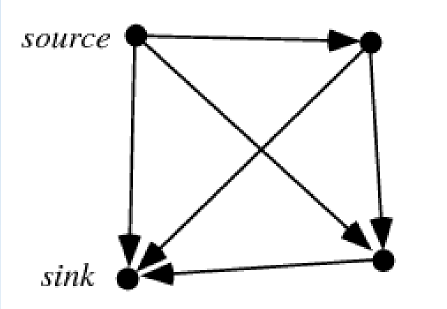
- In general, a graph may contain loops and multiple edges.
- Graphs with no loops or multiple edges are called **simple graphs.**

Brief Introduction of Graph (directed graphs)

- The study of **directed graphs** (or **digraphs,** as we abbreviate them) arises from making the roads into one-way streets.

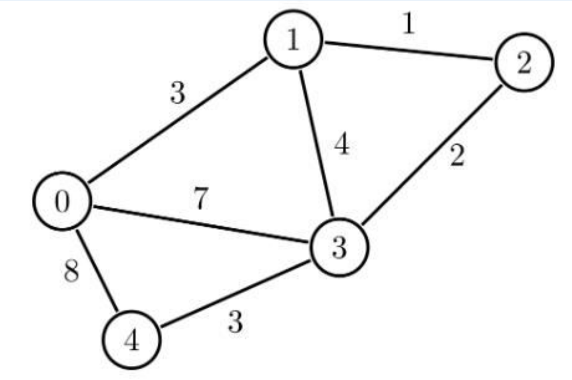


- *V* is a set whose elements are called *vertices*, *nodes*, or *points*;
- *A* is a set of ordered pairs of vertices, called *arrows*, *directed edges* (sometimes simply *edges* with the corresponding set named *E* instead of *A*), *directed arcs*, or *directed lines*.
- For example, an arc (*x*, *y*) is considered to be directed from *x* to *y*.



Brief Introduction of Graph (weighted graphs)

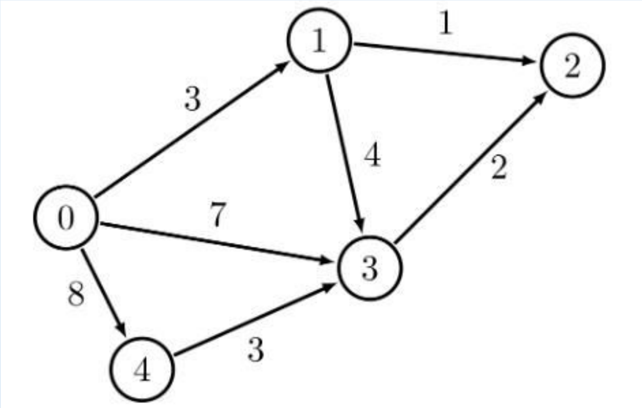
- A **weighted graph** is a **graph** in which each branch is given a numerical **weight**.
- A **weighted graph** is therefore a special type of labeled **graph** in which the labels are numbers (which are usually taken to be positive).



- Weighted graphs are often used to model real objects and processes. For example, the graph above can be considered as a map, where the nodes are cities and the edges are roads. The weight of each edge is the distance between two cities.

Brief Introduction of Graph (weighted graphs)

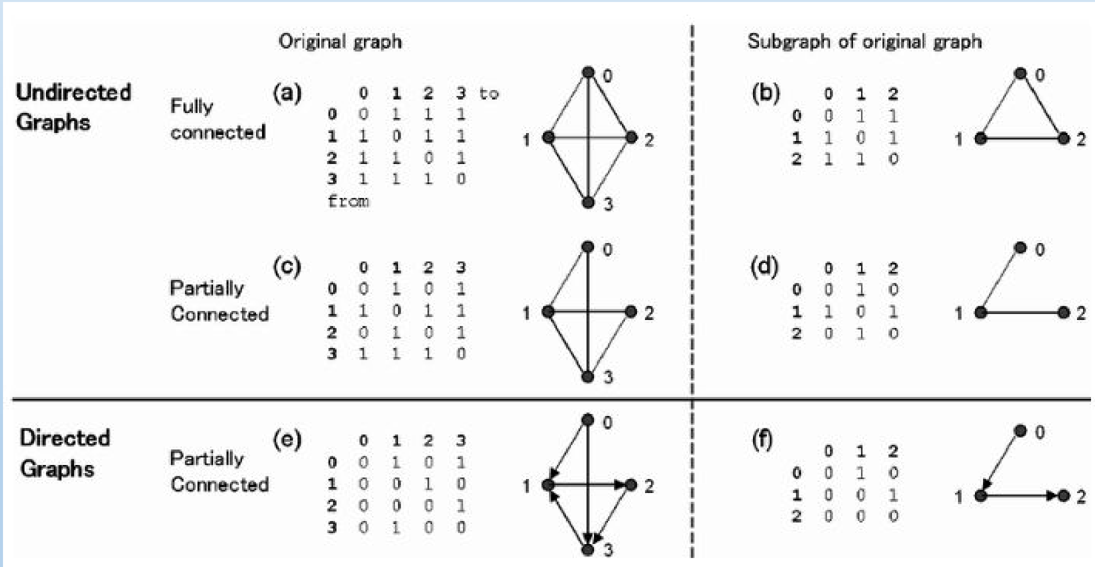
- A **directed weighted graph** is a **directed graph** in which each branch is given a numerical **weight**.



- Weighted directed graphs also can be used to model some real processes.
- In our example, Node 0 can be considered as a storage place, from where we need to transfer some resources to the destination point, say to Node 2.
- The remaining nodes can be considered as intermediate places.
- Each edge shows a direction in which the resources can be transferred.
- The weight of an edge shows the maximum amount of resources that can be sent through the edge.

Brief Introduction of Graph (adjacency array (matrix) and incidence array (matrix))

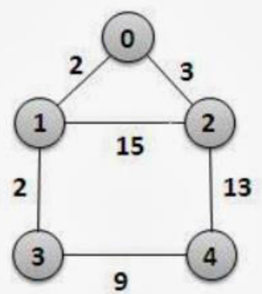
- If G is a graph with vertices labelled $\{1,2, \dots \dots, n\}$, its **adjacency array** $\boldsymbol{A}$ is the *n* x *n array* whose ij-th entry is the number of edges (or the weight of the edge) joining vertex i and vertex j.



Adjacency matrix and graph presentations of different undirected and directed graphs.

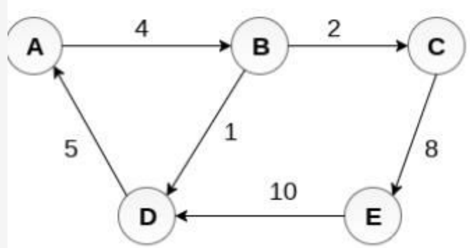
Brief Introduction of Graph
(adjacency array (matrix) and incidence array (matrix))

- If G is a graph with vertices labelled $\{1,2, \dots \dots, n\}$, its **adjacency array** A is the n x n *array* whose ij-th entry is the number of edges (or the weight of the edge) joining vertex i and vertex j.



	0	1	2	3	4
0	0	2	3	0	0
1	2	0	15	2	0
2	3	15	0	0	13
3	0	2	0	0	9
4	0	0	13	9	0

Adjacency Matrix Representation of Weighted Graph



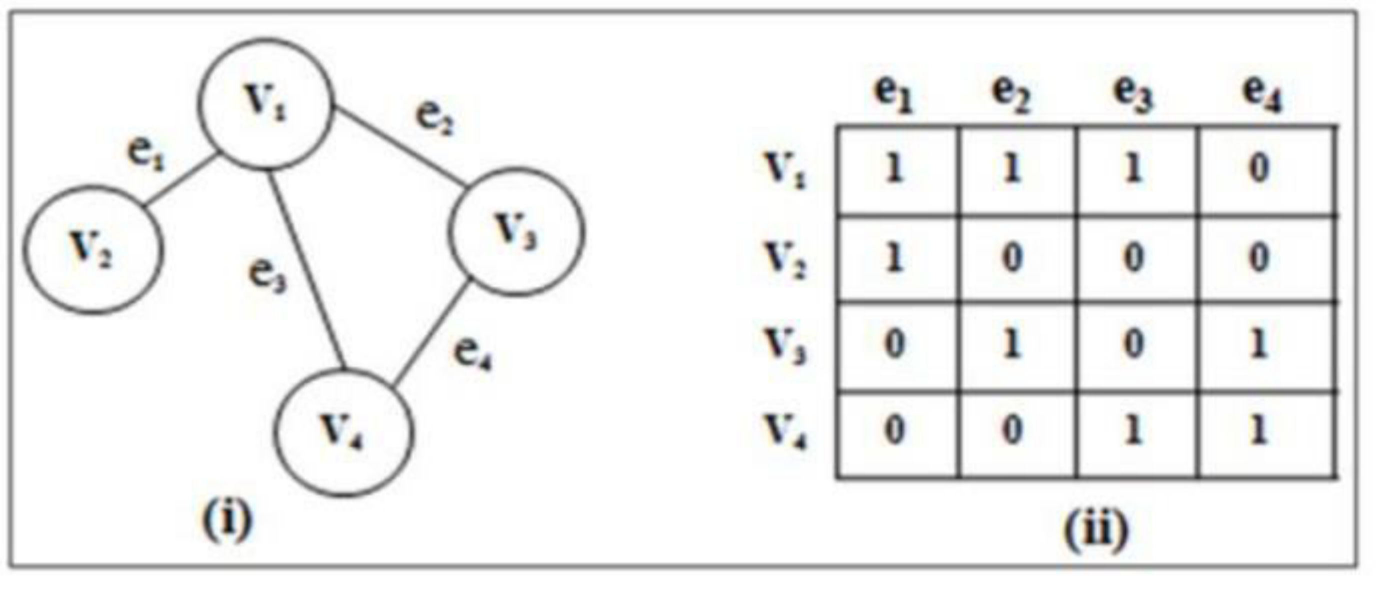
Weighted Directed Graph

	A	B	C	D	E
A	0	4	0	0	0
B	0	0	2	1	0
C	0	0	0	0	8
D	5	0	0	0	0
E	0	0	0	10	0

Adjacency Matrix

Brief Introduction of Graph
(adjacency array (matrix) and incidence array (matrix))

- n nodes and m edges
- Its **incidence array $\boldsymbol{M}$** is the *n* x *m array* whose ij-*th* entry is 1 if vertex i is incident to edge; and 0 otherwise.



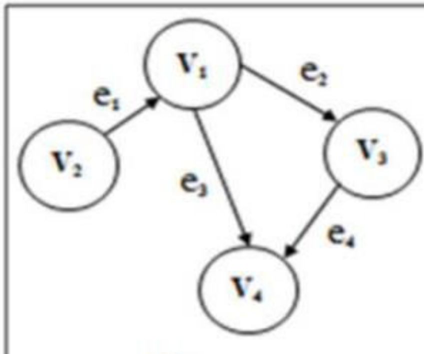
	e_1	e_2	e_3	e_4
V_1	1	1	1	0
V_2	1	0	0	0
V_3	0	1	0	1
V_4	0	0	1	1

(ii)

Brief Introduction of Graph (adjacency array (matrix) and incidence array (matrix))

- n nodes and m edges
- Its **incidence array $\boldsymbol{M}$** is a n × m 2D array
- The ij-*th* entry of $\boldsymbol{M}$ is 1, the j-th edge leaves the i-th node.
- The ij-*th* entry of $\boldsymbol{M}$ is -1, the j-th edge entries the i-th node.

(Note that some authors use opposite sign notation)

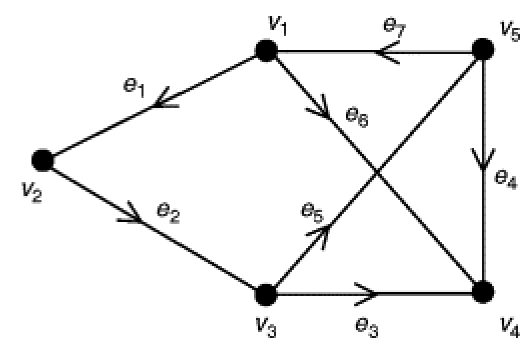


(i)

	e1	e2	e3	e4
V1	-1	1	1	0
V2	1	0	0	0
V3	0	-1	0	1
V4	0	0	-1	-1

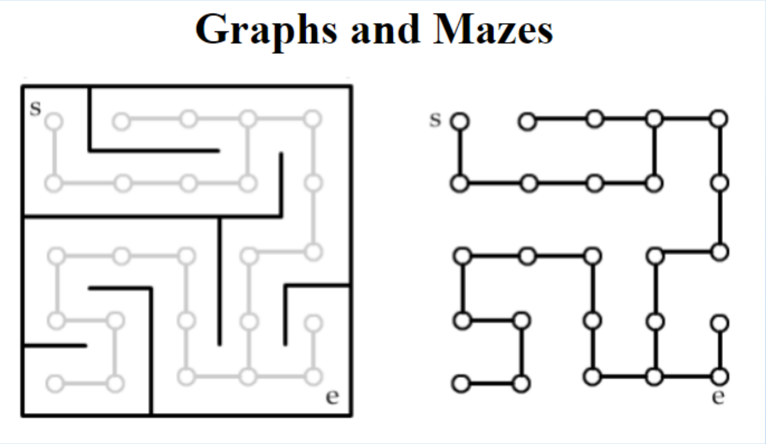
(ii)

$$\begin{array}{c|ccccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \\ \hline v_1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 \\ v_2 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_3 & 0 & -1 & 1 & 0 & 1 & 0 & 0 \\ v_4 & 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ v_5 & 0 & 0 & 0 & 1 & -1 & 0 & 1 \end{array}$$



Brief Introduction of Graph (Some problems and Solutions)

- A maze as a graph.
- Each room in the maze is viewed as a vertex.
- We add edges to the graph between adjacent rooms that are not blocked by a wall.
- Find a shortest path from s to w. ???
- It is the shortest path problem.



Brief Introduction of Graph (shortest path)

- Given a source vertex v_1 , find all the shortest paths to all other nodes.
- One algorithm is Dijkstra's Algorithm

Let $\mathbb{S}$ be the set of vertices to which we have a shortest path.

1. Initialize the cost of the source vertex to 0, and all the rest of the nodes to ∞
2. Initialize set $\mathbb{S}$ to be empty (no element).
3. While $\mathbb{S}$ does not contain all vertices

   a. Select the node A with the lowest cost that is not in $\mathbb{S}$

   b. Put it in $\mathbb{S}$

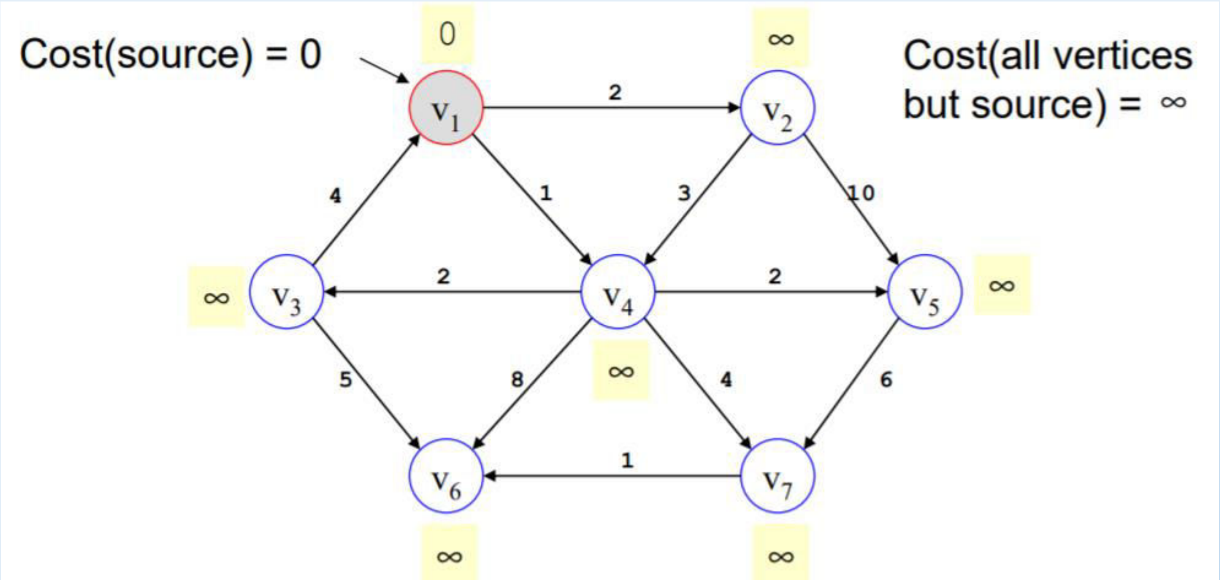
   c. For each node B adjacent to A

      if cost(A)+cost(A,B) < the current cost of B',

      set cost(B) = cost(A)+cost(A,B) and record

      previous(B) = A (so that we can remember the path)

Example

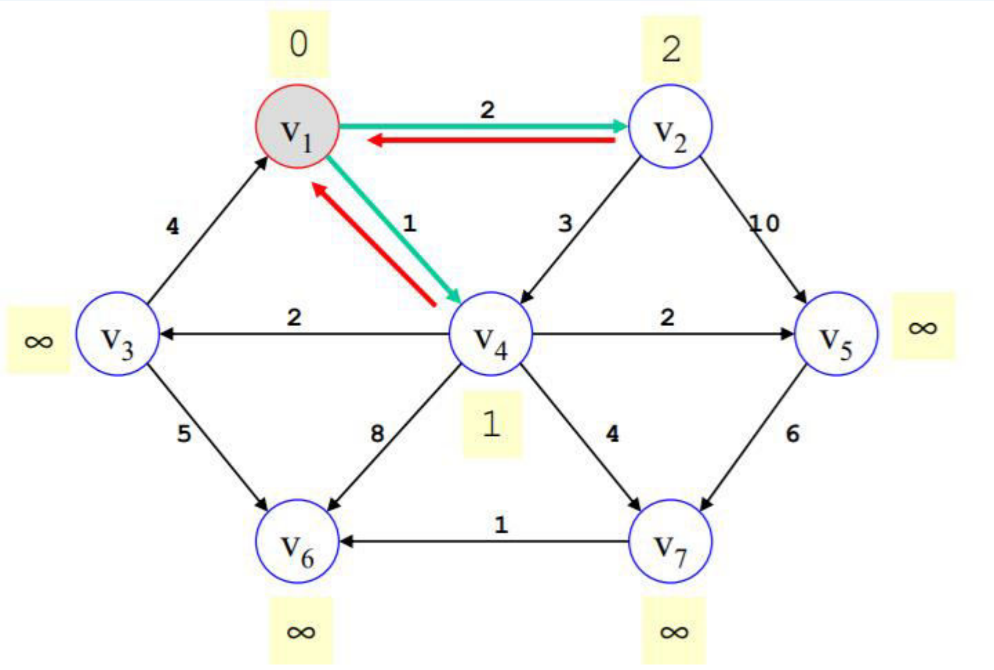


Since $C(v_1) = 0$, put v_1 to $\mathbb{S}$

$\mathbb{S} = \{v_1\}$, previous(v_1) = Nil

Brief Introduction of Graph (shortest path)

Example



$\mathbb{S} = \{v_1\}$

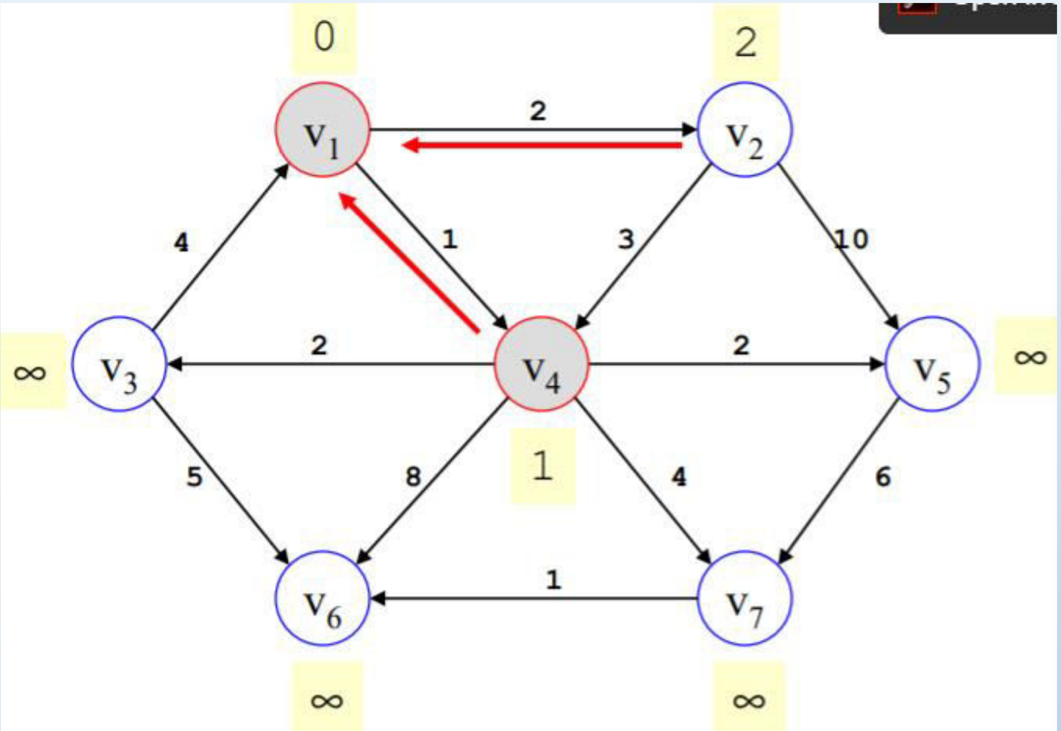
$C(v_2) = 0 + 2 = 2$, previous(v_2) = v_1

$C(v_4) = 0 + 1 = 1$, previous(v_4) = v_1

$\mathrm{C(others)} = \infty$

Brief Introduction of Graph (shortest path)

Example



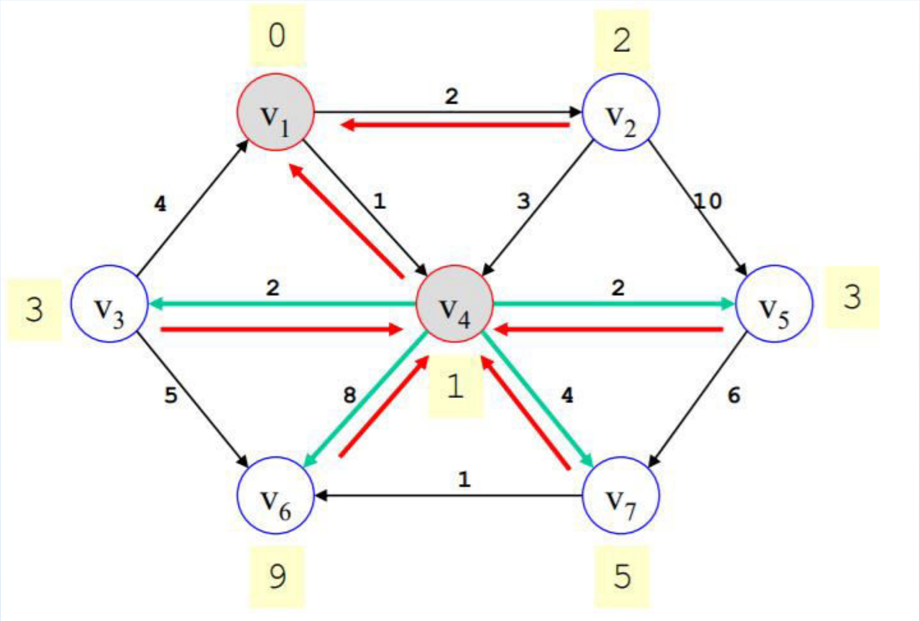
v_4 is with min cost

put v_4 to $\mathbb{S}$

$\mathbb{S} = \{v_1, v_4\}$

Brief Introduction of Graph (shortest path)

Example



$\mathbb{S} = \{v_1, v_4\}$

v_3, v_5, v_6, v_7 are neighbors to v_4

update

$C(v_3) = 1 + 2 = 3$, previous(v_3) = v_4

$C(v_5) = 1 + 2 = 3$, previous(v_5) = v_4

$C(v_6) = 1 + 8 = 9$, previous(v_6) = v_4

$C(v_7) = 1 + 4 = 5$, previous(v_7) = v_4

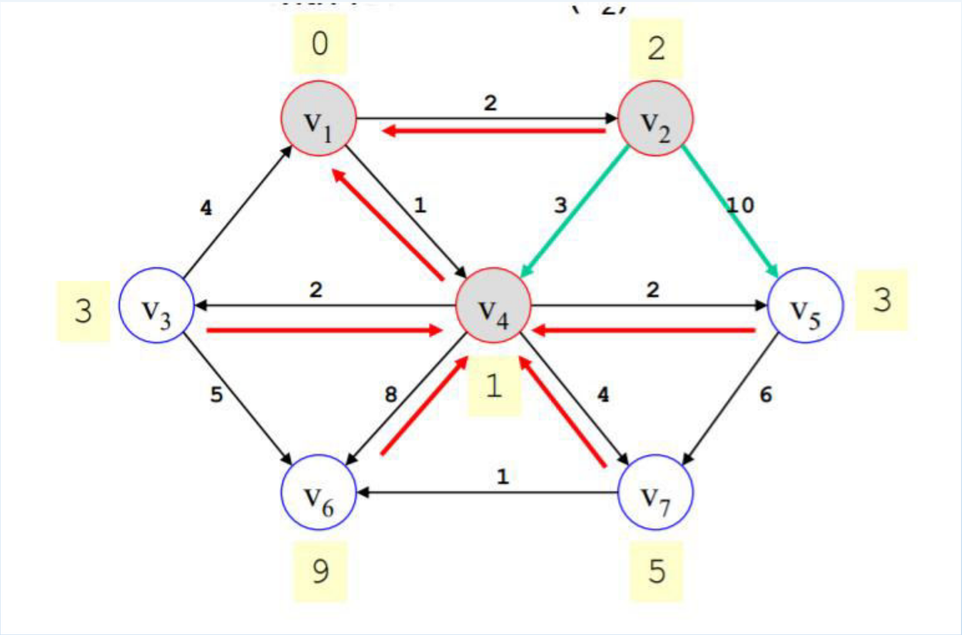
No update

$C(v_4) = 1$, previous(v_4) = v_1

$C(v_2) = 2$, previous(v_2) = v_1 (Note v_2 is with the min cost)

Brief Introduction of Graph (shortest path)

Example



put v_2 to $\mathbb{S}$

$\mathbb{S} = \{v_1, v_2, v_4\}$

v_5 is neighbor to v_2

No update why?

$Current\ C(v_5) < 10 + 2$

$C(v_2) = 2$, previous(v_2) = v_1

$C(v_3) = 3$, previous(v_3) = v_4

$C(v_4) = 1$, previous(v_4) = v_1

$C(v_5) = 3$, previous(v_5) = v_4

$C(v_6) = 9$, previous(v_6) = v_4

$C(v_7) = 5$, previous(v_7) = v_4

Brief Introduction of Graph (shortest path)

Example

Note v_5 is with the min cost)

put v_5 to $\mathbb{S}$

$\mathbb{S} = \{v_1, v_2, v_4, v_5\}$

v_7 is neighbor to v_5

No update why?

$Current\ C(v_7) < 3 + 6$

$C(v_2) = 2$, previous(v_2) = v_1

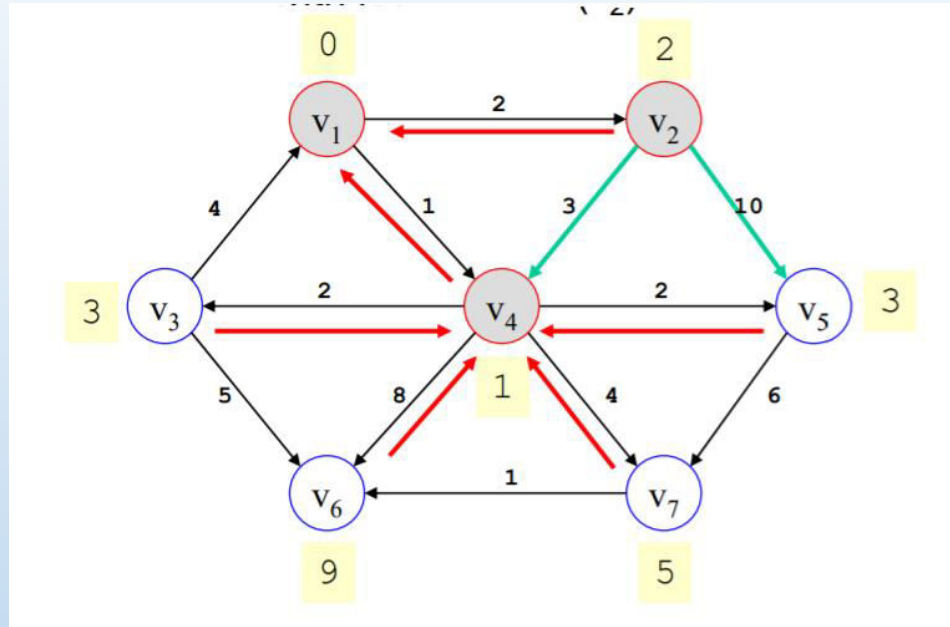
$C(v_3) = 3$, previous(v_3) = v_4

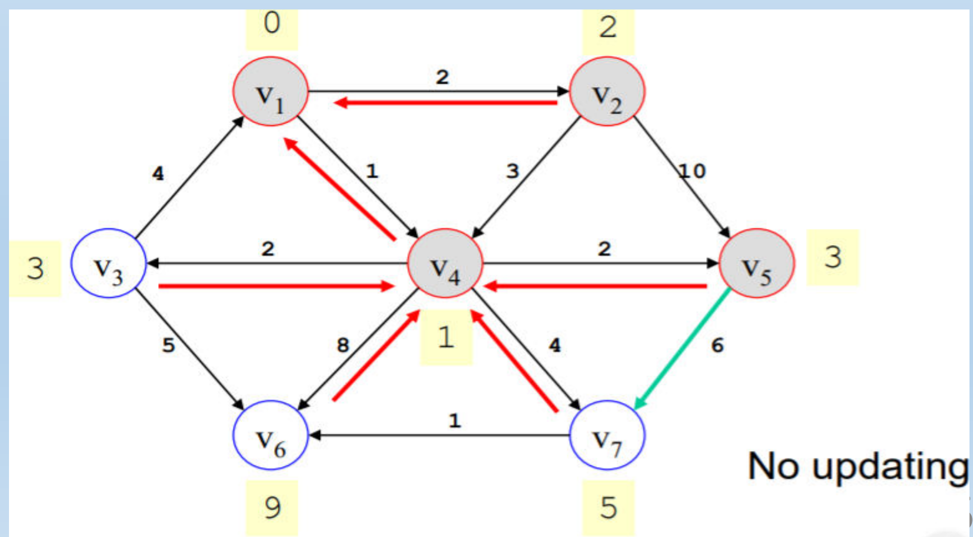
$C(v_4) = 1$, previous(v_4) = v_1

$C(v_5) = 3$, previous(v_5) = v_4

$C(v_6) = 9$, previous(v_6) = v_4

$C(v_7) = 5$, previous(v_7) = v_4





Brief Introduction of Graph (shortest path)

Note v_3 is with the min cost

Put v_3 to $\mathbb{S}$

$\mathbb{S} = \{v_1, v_2, v_3, v_4, v_5\}$

$v_6$is neighbor to v_3

$C(v_2) = 2$, previous(v_2) = v_1

$C(v_3) = 3$, previous(v_3) = v_4

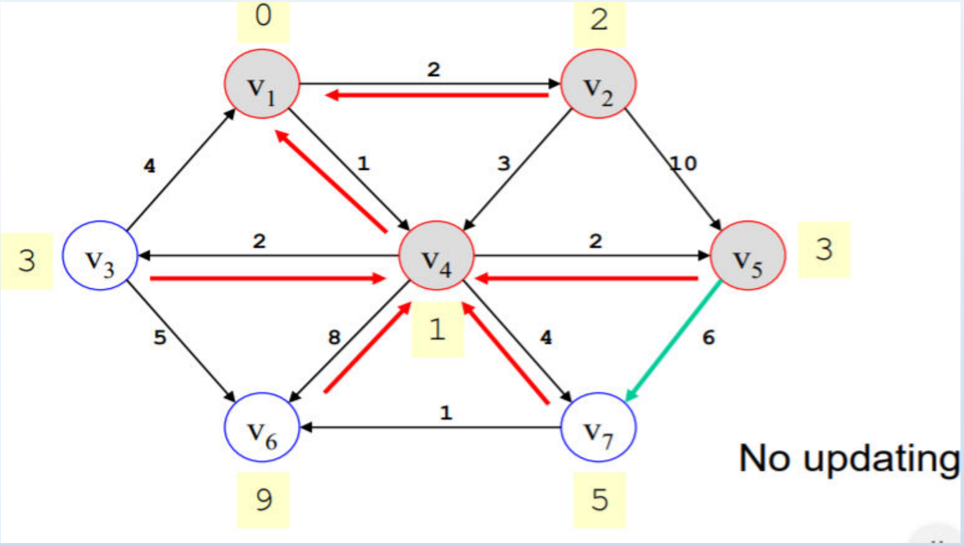
$C(v_4) = 1$, previous(v_4) = v_1

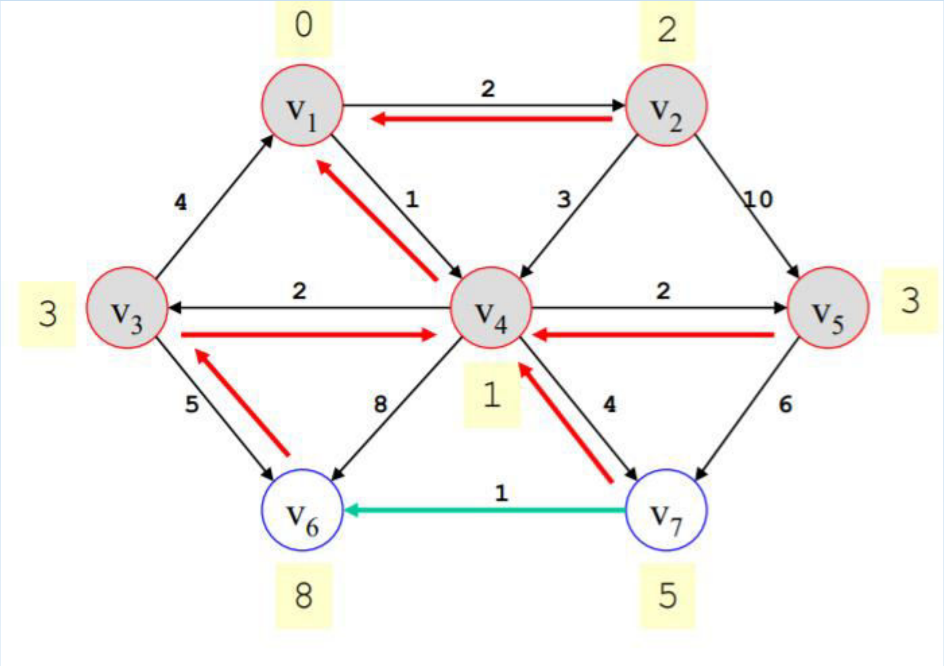
$C(v_5) = 3$, previous(v_5) = v_4

$C(v_6) = 3 + 5 = 8 < 9$ $(current\ cost)$

previous(v_6) = v_3

$C(v_7) = 5$, previous(v_7) = v_4





Brief Introduction of Graph (shortest path)

Note v_7 is with the min cost

Put v_7 to $\mathbb{S}$

$\mathbb{S} = \{v_1, v_2, v_3, v_4, v_5, v_7\}$

v_6 is neighbor to v_7

$C(v_2) = 2$, previous(v_2) = v_1

$C(v_3) = 3$, previous(v_3) = v_4

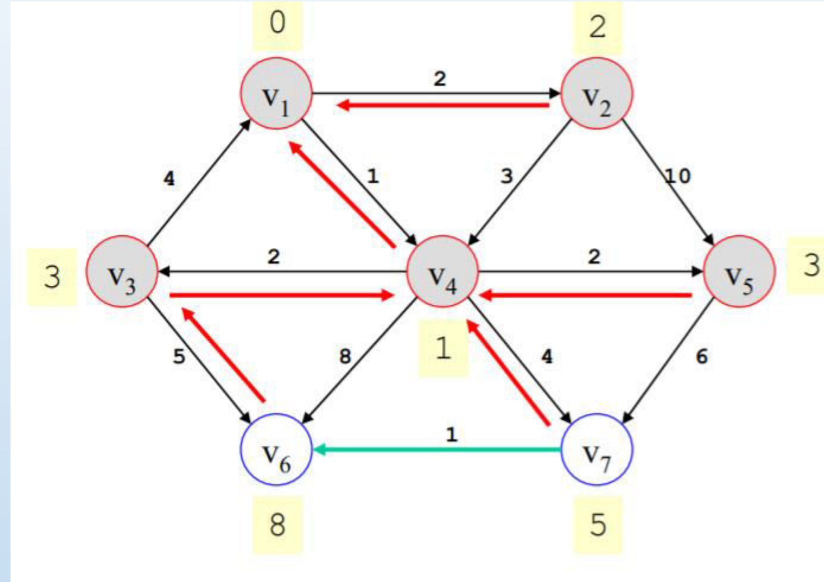
$C(v_4) = 1$, previous(v_4) = v_1

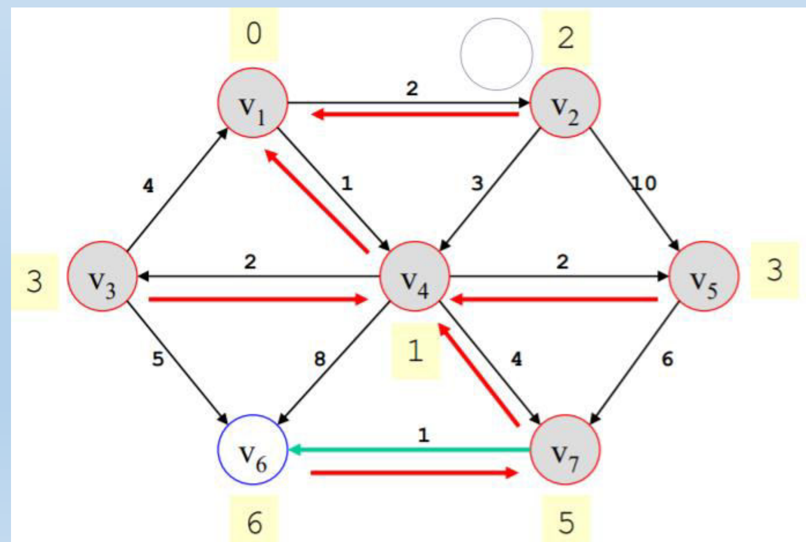
$C(v_5) = 3$, previous(v_5) = v_4

$C(v_6) = 1 + 5 = 6 < 8\ (current\ cost)$,

previous(v_6) = v_7

$C(v_7) = 5$, previous(v_7) = v_4





Brief Introduction of Graph (shortest path)

Put v_6 to $\mathbb{S}$

$\mathbb{S} = \{v_1, v_2, v_3, v_4, v_5, v_6 v_7\}$

$C(v_2) = 2$, previous(v_2) = v_1

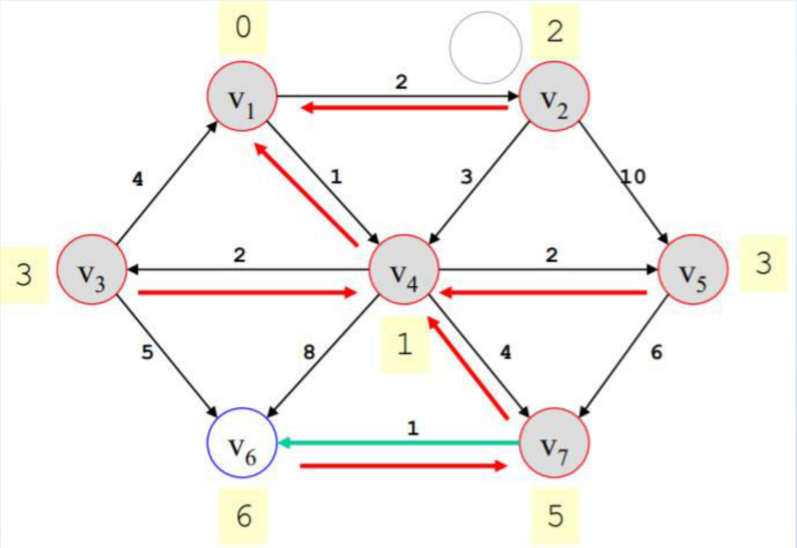
$C(v_3) = 3$, previous(v_3) = v_4

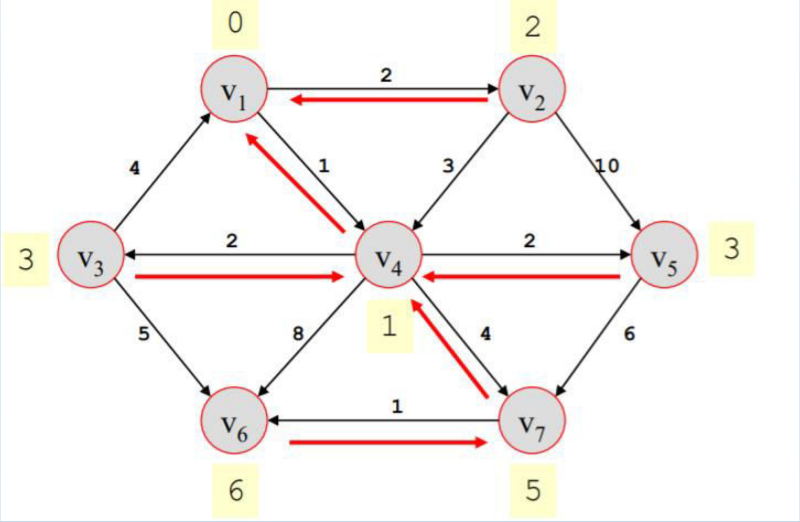
$C(v_4) = 1$, previous(v_4) = v_1

$C(v_5) = 3$, previous(v_5) = v_4

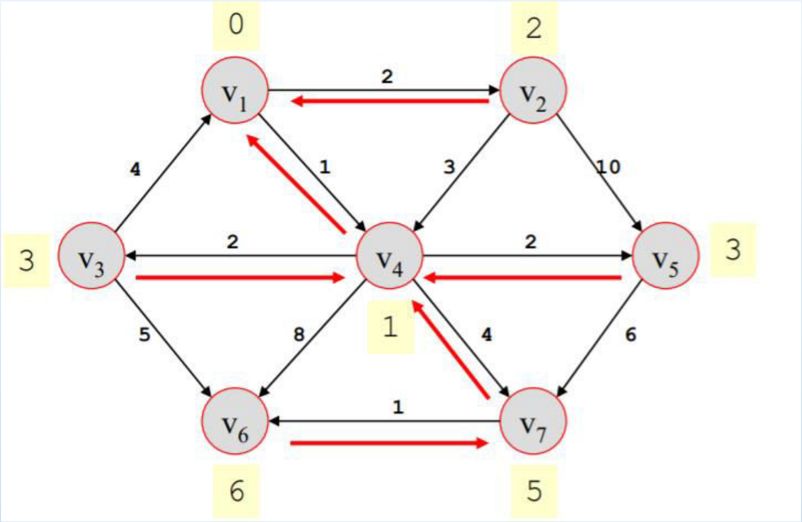
$C(v_6) = 6$, previous(v_6) = v_7

$C(v_7) = 5$, previous(v_7) = v_4





Brief Introduction of Graph (shortest path)



Put v_6 to $\mathbb{S}$

$\mathbb{S} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$

$C(v_2) = 2$, previous(v_2) = v_1

$C(v_3) = 3$, previous(v_3) = v_4

$C(v_4) = 1$, previous(v_4) = v_1

$C(v_5) = 3$, previous(v_5) = v_4

$C(v_6) = 6$, previous(v_6) = v_7

$C(v_7) = 5$, previous(v_7) = v_4

Using back track, we can find out the shortest path.

For example, what the shortest path from v_1 to v_7 ? cost of the path is 5, previous(v_7) = v_4 , and

previous(v_4) = v_1, so v_1->$v_4 \rightarrow v_7$

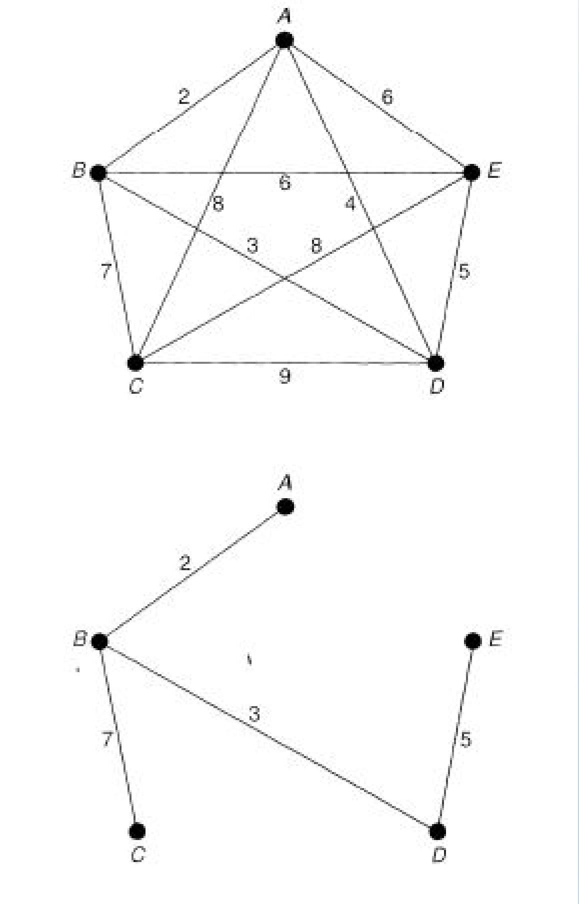
What the shortest path from v_1 to v_6 ? cost of the path is 6, previous(v_6) = v_7, previous(v_7) = v_4 , and previous(v_4) = v_1, so v_1->$v_4 \rightarrow v_7 \rightarrow v_6$

The minimum connector problem

- Find a sub-graph with the minimum total length from a connected graph.

Application:

- Laying cables or pipes among a number of sites.
- The weights of the edges are the cost to lay cable (pipe) between two sites.



*Brief Introduction of Graph (*Maximum Flow)

Maximum Flow

- A flow network is defined as a directed graph involving a source(S), a sink(T) and several other nodes connected with edges.
- Each edge has an individual capacity which is the maximum limit of flow that edge could allow.
- For any non-source and non-sink node, the input flow is equal to output flow.
- **Maximum Flow :** the maximum amount of flow that the network would allow to flow from source to sink.
- What is the corresponding flow paths?