Chapter 1. Charles

Basic Concepts of Graph

Definition

$G = \{V, E\}$ = Vertices + Edges

Example:

$e = (u, v) =$

u, v are ends of e.   or

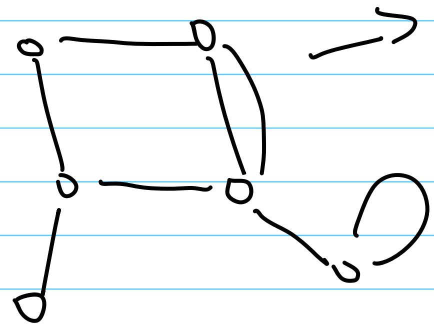
e join vertex u and v

Definition

ε = number of edges $= |V|$

ν = number of vertices $= |E|$

Example:



→ Multiedge 重边

→ loop 环边

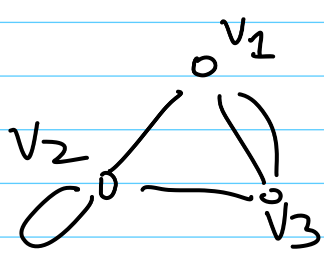
Definition

Dedgree = number of edges incident with vertex v : $d(v)$

$\Delta(G) = \max\{d_G(v) \mid v \in V(G)\}$

$\delta(G) = \min\{d_G(v) \mid v \in V(G)\}$

Example:



$d(v_1) = 3$

$d(v_2) = 4$ (loop = 2)

$d(v_3) = 3$

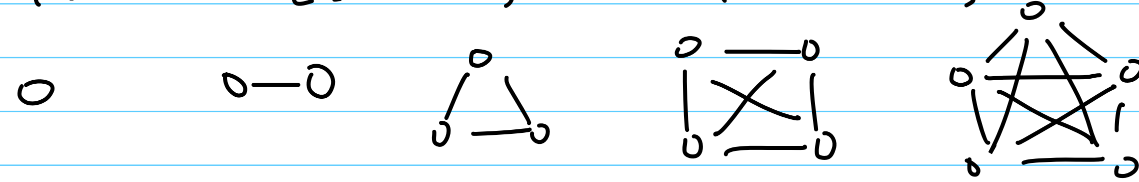
$\Delta(G) = \max\{d(v_1), d(v_2), d(v_3)\} = 4$

$\delta(G) = \min\{d(v_1), d(v_2), d(v_3)\} = 3$

Definition

Complete Graph: Any two vertex is adjacent with each other. (K_n)

K_1: K_2: K_3: K_4: K_5:



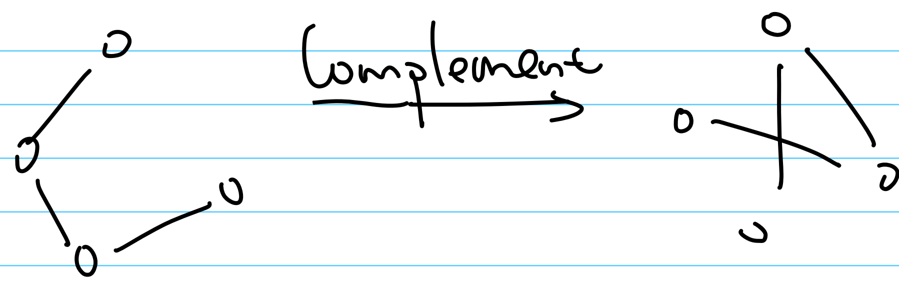
Regular Graph:

$\forall$ two vertex has same degree,

$k = d(v_i) = d(v_j)$ $\{v_i, v_j \in V(G)\}$

↳ called k-regular graph.

Complement 补圖 $\bar{G}$



Theorem:

$$\sum_{v \in V(G)} d(v) = 2e$$

Corollary:

Number of odd degree vertices is always even. (including 0)

Subgraph

Definition:

If $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$,

Then $H \subseteq G$.

Spanning Graph: $V(H) = V(G)$

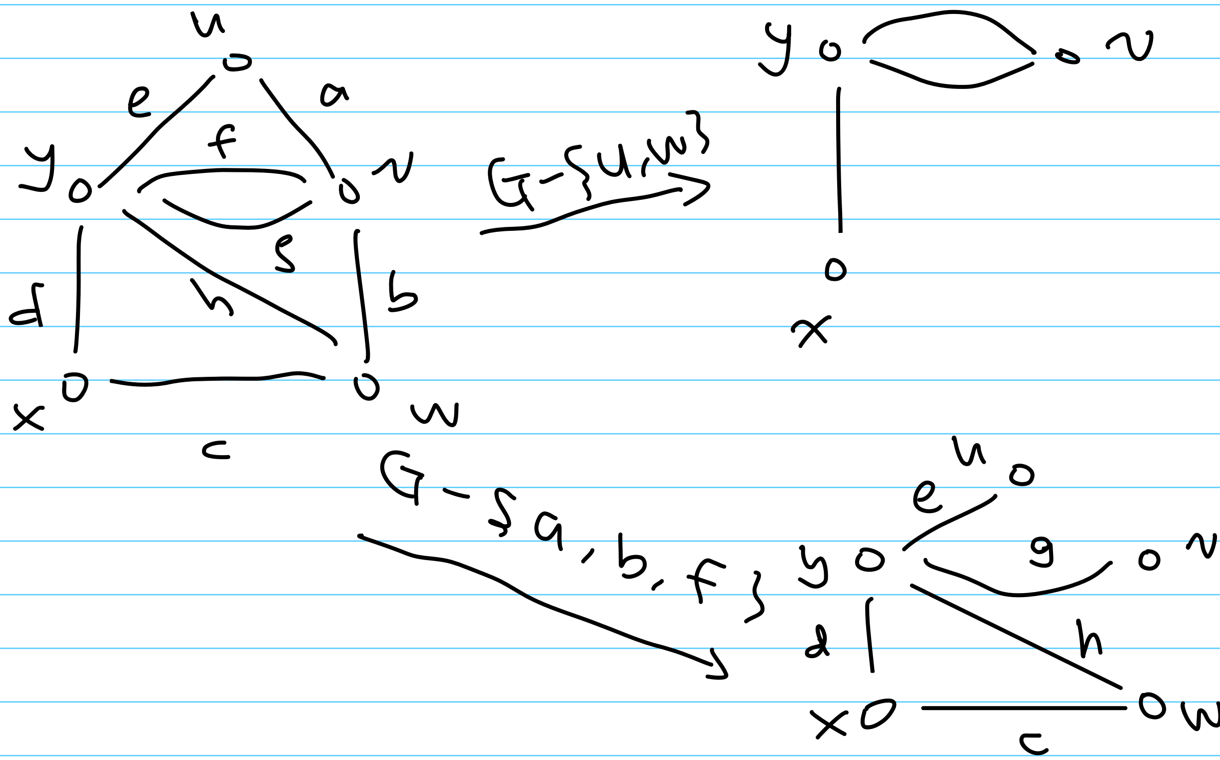
Induced subgraph:

$V' \subseteq V(G)$, The subgraph of G whose vertex set is V' and whose edge set is the set of those edges of G that have both ends in V' is called the subgraph of G induced by V' and is denoted by $G[V']$

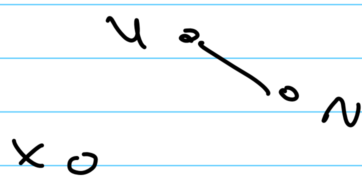
Edge-induced subgraph:

$E' \subseteq E(G) \rightarrow G[E']$

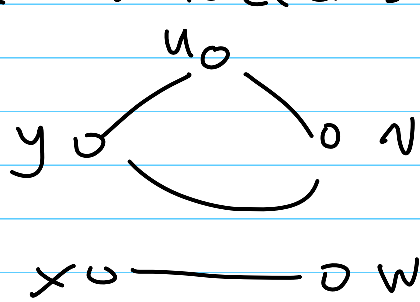
Example:



The Induced subgraph, $G[\{u,v,x\}]$



The Edge-induced subgraph, $G[\{a,c,e,g\}]$



※ Symmetric Difference

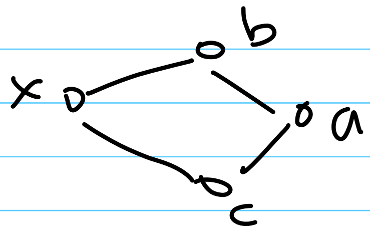
$$(A-B)\cup(B-A) = A \oplus B$$

$G_1 \oplus G_2 = \{V, E_1 \oplus E_2\}$

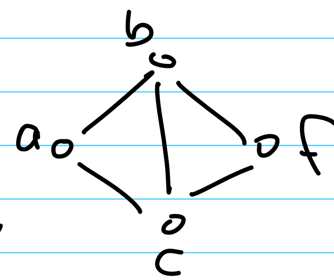
where $G_1 = \{V, E_1\}$. $G_2 = \{V, E_2\}$

!!! $V(G_1) = V(G_2)$

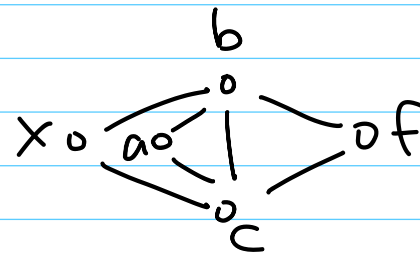
Example:



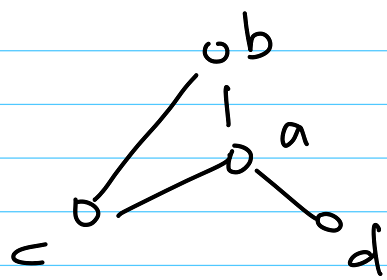
G_1



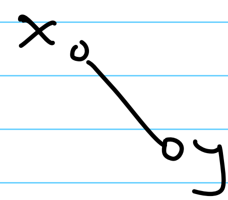
G_2



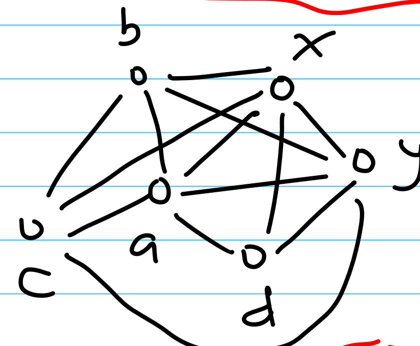
$G_1 \cup G_2$



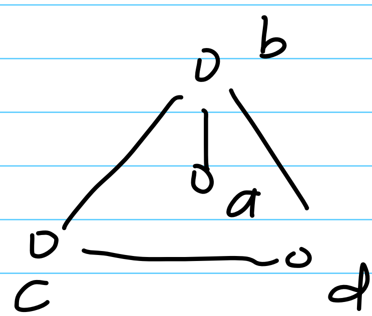
G_1



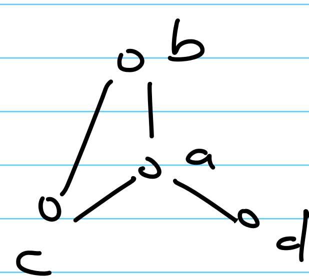
G_2



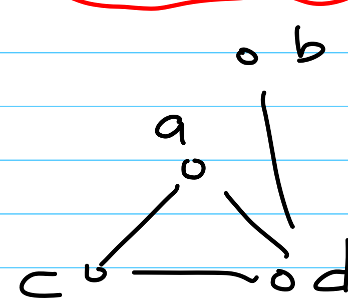
$G_1 \vee G_2$



G_1



G_2



$G_1 \oplus G_2$

Path and Cycle

$$\begin{tikzcd}
\text{Walk} \arrow[r, "\text{No Repeat}", "\text{Edges}"'] \arrow[d] & \text{Trail} \arrow[r, "\text{No repeat}", "\text{Vetrices}"'] \arrow[d] & \text{Path} \arrow[d] \\
\text{Closed Walk} \arrow[r] & \text{Closed Trail} \arrow[r] & \text{Cycle}
\end{tikzcd}$$

Number of edges is length

distance = length of shortest path

Corollary:

G is a simple Graph, $\delta(G) \geq 2$, then G must have cycle.

Corollary:

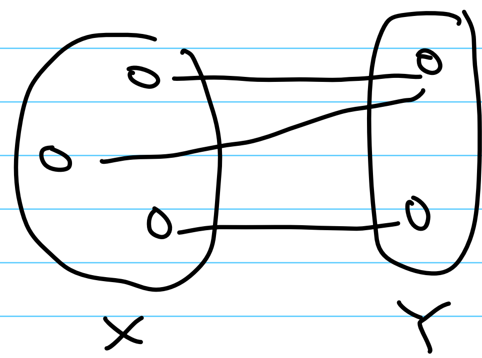
G is a simple Graph, $\delta(G) \geq 3$, then G must have even cycle.

Bipartite Graph

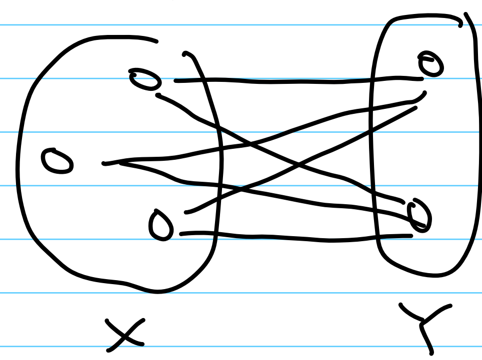
(二部图)

$G = (X \cup Y, E)$ or $G = (X, Y)$

Example:



Complet Bipartite graph:



Theorem.

Bipartite Graph if and only if No odd cycle.

Proof:

$\Rightarrow$ $C = v_0 v_1 v_2 \cdots v_k v_0$ is a cycle of $G = (X \cup Y, E)$, then let $v_0 \in X$.

So $v_1 \in Y$, $v_2 \in X$ $\cdots$ $v_k \in Y$

have $k+1$ edges, and k is odd, then an even cycle.

$\Leftarrow$ let G doesn't have odd cycle.

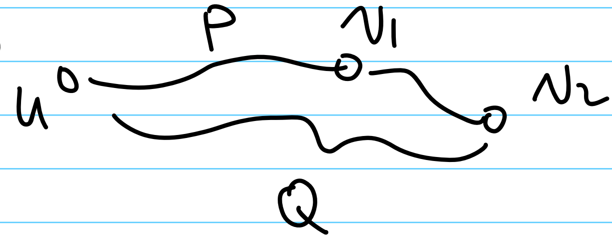
$\forall u \in V(G)$, Let

$X = \{v \in V(G) \mid d(u,v) \text{ is odd}\}$

$Y = \{v \in V(G) \mid d(u,v) \text{ is even}\}$

Let P, Q is shortest path from u to v_1, v_2, $\forall e = v_1 v_2 \in E(G)$

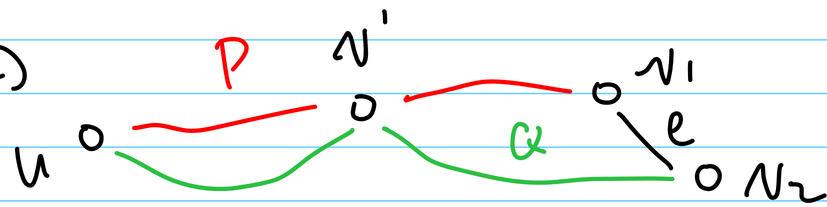
(1)



$\therefore$ $v_1 \in X$, $v_2 \in Y$ or

$v_1 \in Y$, $v_2 \in X$

(2)



$\because d(u,v')$ in $P = d(u,v')$ in Q

$\therefore$ If P and Q both odd or even

then $v'v_1v_2v'$ is odd cycle ✗

$\therefore v_1 \in X,\ v_2 \in Y$

$v_1 \in Y,\ v_2 \in X$ ■

Connection

Theorem:

If G is connected, then

$$\varepsilon(G) \geq \nu(G) - 1$$

Isomorphic 略

Shortest Path Problem

- Dijkstra Algorithm $O(\nu^2)$

Tree

Connected Graph without cycles.

Corollary:

Non-trivial tree has at least two leaves ($d(v)=1$)

Spanning Tree

Definition:

$T \subseteq G$, T is a tree. $V(T) \subseteq V(G)$

then T is a spanning tree of G

Minimum Spanning tree Problem

- Kruskal Algorithm
- Prime Algorithm

中心与中位点、略.