9 Graphs

9.1 Graphs and Graph Models

Definition *1 An **undirected graph**(**ugraph**, **simple graph**) G = (V, E) where edge is a collection of **two elements subsets**(**unordered pair**) of V.*

$\{u, v\} \in E$ vs. $(u, v), (v, u) \in E$.

*Edge {u, v} **connectes** two **end points** u, v.*

Multigraph *G = (V, E, f)* ***Edge weighted graph***

$f: E \to \mathbf{N}$

*Egde {u, v} of **multiplicity** m, if* $f(\{u, v\}) = m \in \mathbf{N}$.

Pseudograph *G = (V, E)*

*Edge is a collection of **one or two elements** subsets of V.*

$\{u, v\} \in E$ connects u and v

$\{u\} \in E$ self-loop

Definition *2 A* ***directed graph(digraph***, ***graph****) G on the set of* ***vertices*** *V is, G = (V, E) where E ⊆ V × V. A pair (u, v) ∈ E is called* ***edge(arc)*** *and said to* ***starts*** *at the* ***vertex(node)*** *a and* ***ends*** *at the* ***vertex*** *b.*

Edges *of* ***digraph*** *is a binary* ***relation*** *on V.*

Reflexivity	*All vertices have a self-loop*
Irreflexivity	*No verteces have a self-loop*
Symmetry	*All edges are bidirectional*
Antisymmetry	*All edges are unidirectional, self-loop is allowded*
Asymetry	*All edges are unidirectional, self-loop is* ***not*** *allowded*
Transitivity	*All paths should have an (****extra****) edge*

Digraph simulates ugraph and pseudograph

{u, v} in ugraph	*(u, v), (v, u) in digraph*	*symmetricity*
{u} in digraph	*(u, u) in digraph*	*reflexivity*

Definition *2.1 Let G = (V, E) be a digraph. Then a* ***path*** *is defined as a* ***sequence*** *of vertices* $(a_0, a_1, \ldots, a_n)$ *with* $n \geq 0$ *such that*

$(a_i, a_{i+1}) \in E, 0 \leq \forall i < n$. *The path* ***starts*** a_o *and* ***ends*** a_n.

The ***length*** *of the path is defined to be n.*

$(a) \in V$ *path of length 0;* $n = 0, a_0 = a$.

A path is a ***cycle*** *if* $a_0 = a_n$.

$a\,E^* b$ *there is a path from a to b*

$a\,E^+ b$ *there is a positive length path from a to b*

Definition *2.2 A* ***cycle*** *is a positive lengh path that begins and ends* ***same*** *vertex. A* ***directed acyclic graph****(DAG) is a directed graph with* ***no cycle****.*

Lemma *If G is a DAG,* G^+ *is a* ***irreflexive partial order****.*

9.2 Graph Terminology and Special Types of Graphs

Consider simple graph(ugraph)

Definition *1* ***Adjacency***

edge {u, v} ∈ E

u, v are ***adjacent****(****neighbors****,* ***connected****)*

Edge {u, v} ***connects****(is* ***incedent*** *with) vertices u and v.*

Vertices u and v are ***endopints*** *of edge {u, v}.*

Definition *2* ***Degree*** *of Vertex, v ∈ V, deg(v) ∈* **N**.

the number edges incedent with it(except self-loop counts twice)

If deg(v) = 0, v is called ***isolated****.*

If deg(v) = 1, v is called ***pendant****.*

Theorem *1 The* ***Handshaking Theorem***

$$\sum_{v \in V} deg(v) = 2|E|.$$

proof *Every edge contibutes two to the sum of the degrees.*

Theorem *2 Any undirected graph has* ***even*** *number of* ***vertices*** *of* ***odd*** *degree.*

Some Special Simple Graphs

Complete graph, *for* $n \geq 1$, *n vertices* K_n, *has*

an edge between ***every two vertices***. $n(n-1)/2$ *edges*

Empty *graph has* ***no edge*** *at all.*

Line, L_n: $n+1$ *vertices and n edges. A* ***path*** *of length n.*

Cycle, *for* $n \geq 3$, C_n: *n vertices and n edges. A* ***cycle*** *of length n.*

Wheel, *for* $n \geq 3$, W_n: C_n *and n edges from a* ***hub*** *vertex and to* C_n.

$n+1$ *vertices,* $2n$ *edges.*

n-cube, *For* $n \geq 0$, Q_n, *with* 2^n *vertices and .*

$Q_0 = (\{v_0\}, \{\})$.

Let $Q_n = (V, E)$ *where* $V = \{v_0, v_1, \ldots v_{2^n-1}\}$ *where k = and*

$Q_n' = (V', E')$ *where* $V' = \{u_0, u_1, \ldots u_{2^n-1}\}$ *and* $V \cap V' = \emptyset$. *Then*

$Q_{n+1} = (V \cup V', E \cup E' \cup \{\{v_0, u_0\}, \{v_1, u_1\}, \ldots, \{v_{2^n-1}, u_{2^n-1}\}\})$

Bipartite Graph

$G = (V, E)$ *is bipartite, iff* $V = V_1 \cup V_2$ *where* $V_1 \cap V_2 = \emptyset$ *and*

$\forall \{v_1, v_2\} \in E, v_1 \in V_1, v_2 \in V_2$.

Partition $\{V_1, V_2\}$ *is called* ***bipartition***.

Complete Bipartite Graph, $K_{n+m} = (V_1 \cup V_2, E)$ *where* $V_1 \cap V_2 = \emptyset$ *and*

$|V_1| = n$, $|V_2| = m$, *and* $E = \{\{v_1, v_2\} | v_1 \in V_1, v_2 \in V_2\}$

Definition *6 A* ***subgraph*** *of graph $G = (V, E)$ is $H = (V', E')$, if $V' \subseteq V$ and $E' \subseteq E$.*

Let $G' = (V', E')$ is a subgraph of $G = (V, E)$. Then G is called ***spanning subgraph*** *of G, if $V' = V$.*

Let $G' = (V', E')$ is a subgraph of $G = (V, E)$. Then $G'' = (V'', E'')$ is called the ***complement*** *of G' with respect to G where $E'' = E - E'$, and $V'' = \{a \in V \mid \{a, b\} \in E''\}$*

The ***complement*** *of ugraph $G = (V, E)$ with respect to K_n, denoted $\overline{G}$, is the (**absolute**)* ***complement*** *of G.*

Definition *7 The union of two graph $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is $G_{G_1 \cup G_2} = (V_1 \cup V_2, E_1 \cup E_2)$.*

9.3 Representing Graphs and Graph Isomorphism

Isomorphism of Graphs

Definition 1 The graph $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are **isomorphic**,
iff $\exists$ a bijection .∍. $\forall u \forall v \in V_1$: $\{u, v\} \in E_1$ iff $\{f(u), f(v)\} \in E_2$.

Let $Q = (V, E)$ a graph. A **bijection** $f: V \to V'$ where $V \cap V' = \emptyset$. Then
$Q' = (\{f(v) \in V' | v \in V\}, \{\{f(v), f(u)\} | \{v, u\} \in E\})$
We may extend f domain and codomain of f from vertices to edges.
$f(\{v, u\}) = \{f(v), f(u)\}$.
$Q' = (f(V), f(E))$

Q and Q' are isomorphic.
Q' is called an isomorphic image of Q.

n-cube, *revisited.*

i) Q = ({v}, { }) be a ***0-cube***.

ii) Let Q be an ***n-cude***. *Then*

$Q' = (V \cup f(V), E \cup f(E) \cup \{V, f(V)\}$ *is an (n+1)-cube.*

Graph isomorphism

Example 8, 9, 10

Difficult to check!

Necessary *conditions for graph* ***isomophisms***

$|V_1| = |V_2|$ *and* $|E_1| = |E_2|$.

The number of vertices with degree n is same in both graphs.

Subgraph H of one graph is isomorphic to a subgraph of the other

isomorphic simple cycle

9.4 Connectivity

Definition *1 Let G = (V, E) be a ugraph. Then a* ***path*** *is from u to v is defined as a* ***sequence*** *of vertices* $(v_o, v_1, \ldots, v_n)$ *with* $n \geq 0$ *such that*

$\{v_i, v_{i+1}\} \in E, 0 \leq \forall i < n, v_o = u, v_n = v.$

The ***length*** *of the path is defined to be n(≥0).(number of edges)*

Note that u is a path of length 0 but not an edge.

A path is a ***cycle*** *if* $a_o = a_n, n \geq 1$.

A path ***traverses*** *the vertices along it.*

A path is ***simple****, if it contains no* ***vertex*** *more than once.*

A graph, C, is ***simple cycle*** *of* ***length*** *n, iff it is isomorphic to* C_n *for some* $n \geq 3$.

Definition *3 Two vertices u and v are to said to be* ***connected****,*
if there is a path between them.
A graph is said to be ***connected****,*
if there ***every*** *pair of vertices are connected.*
We write u connecs v or u is connected to v, if u and v are connected.

$$connected \subseteq V \times V \qquad \text{binary relation on } V$$

Lemma *0.1 The binary relation* ***connected*** *on V is an* ***equivalence*** *relation*

Definition *3.2 The* ***quivalence class*** *defined by the equivalence relation* ***connected*** *is called* ***connected component****.*

$P = \{[v]_{connected} \mid v \in V\}$ ***partition*** *of V.*

$|P|$ *= number of conncted components*

Definition *3.3 Two vertices in a graph are* ***k-connected*** *if they* ***remain connected*** *in any subgraph by* ***deleting*** *k–1 edges.*

A graph is ***k-connected*** *if every pair of vertices are k-connected.*

simple cycle	*2-connected*
K_n	*(k–1)-connected*

Theorem *1 If vertex v is connected to vertex u in a graph,*

there is a ***simple path*** *from u to v.*

proof *Consider* ***minimum*** *length path from u to v.*

$v_o, v_1, \ldots, v_k$ *where* $v_o=u$, $v_k=v$ *with* $k \geq 2$*(if* $k \leq 1$*, simple) is simple.*

Assume it is not simple, $0 \leq \exists i < \exists j < n$ *.϶.* $v_i = v_j$*. Not minimum path!*

Corollary *1.1 For any path of* ***length*** *k in graph, there is a* ***simple*** *path of length at* ***most*** *k with the same endpoints.*

Theorem *1.1 Every graph G = (V, E) has* ***at least*** *|V| – |E|* ***connected components****.*

proof *P(n): G = (V, E) with |E| = n has at least |V| – n C. C.*

base*: |E| = 0, |V| connected components.*

induction*: Consider G = (V, E) with n+1 edges.*

Remove *an edge {u, v} and call the resulting graph G'.*

G' has at least |V| – n connected components.(I. H.)

Add back *the edge {u, v}.*

case 1: u and v are in the ***same*** *C. C.* ***Same*** *number of components*

G has at least |V| – n > |V| – (n+1) components.

case 2: u and v are in the ***diffrent*** *C. C. One* ***less*** *component*

G has at least |V| – n – 1 = |V| – (n+1) components.

Corollary *1.2 The* ***connected*** *graph with n vertices has* ***at least*** *n-1 edges.*

proof $1 \geq |V| - |E|$ $|E| \geq |V| - 1$

10.1 Introduction to Trees

Definition *10.1 A **tree** is an **acyclic connected** graph.*

*A vertex of **degree one** is called **leaf**.*

*A **forest** is a set of trees.*

Theorem *Every **tree** has following properties:*

*1. Any **connected subgraph** is a **tree**.*

***2**. There is a **unique (simple) path** between **every pair of vetices**.*

*3. **Adding** an **edge** between two vertices create a **(simple) cycle**.*

*4. **Removing** any **edge disconnects** the tree.*

*5. If tree has at **least two vertices**, then it has **at least two leaves**.*

***6**. The number of **vertices** is **one larger** than that of **edges**.*

proof *1. Any subgraph of acyclic graph subgraph is also **acyclic**.*

*2. There is **at least one** (simple) path, since connected and acyclic.*

*Assume **two different** simple paths from u to v.*

Assume x be the first vertex where the path ***diverge****,*

y be the next vertex they ***share****.*

There is a ***cycle*** *from x to y and then y to x.*

3. *Additional edge* $\{u, v\} \cup$ *(simple)* ***path*** *from u to v = (simple)* ***cycle***

4. *Remove* $\{u, v\}$. ***Unique*** *simple path was (u, v).* $\therefore$ *Not connected*

5. *Let* $(v_1, \ldots, v_m)$ *be the* ***longest*** *simple path in the tree. Then* $m \geq 2$.

$2 < \forall i \leq m, \{v_1, v_i\} \notin E$, *since* $(v_1, \ldots, v_i, v_1)$ *is a* ***cycle****.*

$\{u, v_1\} \notin E$ *where u is not in the path, since* $(v_1, \ldots, v_m)$ *is the* ***longest****.*

$\therefore$ v_1 *is a* ***leaf****. By* ***symmetric*** *argumant* v_m *is a* ***second*** *leaf.*

6. *Induction on number of vertices.*

n=1, no edge. 0 + 1 = 1. O.K.

Consider (n+1)-vertex tree T and let v be a ***leaf*** *of T.*

Deleting v and its incident edge gives a smaller ***tree****.*

$$(|E| - 1) = (|V| - 1) + 1$$

Adding v and its incident edge gives a larger ***tree****.* $\therefore$ $|E| = |V| + 1$.

Theorem *Every **connected graph** has **spanning tree**.*

proof *Let T be a **connected spanning** subgraph of G*
*with the **smallest** number of **edges**.*
Suppose T has a cycle $(v_0, v_1, \ldots, v_n, v_0)$.
Suppose we remove the edge $\{v_n, v_0\}$.
*If arbitrary vertices x and y has a **path not** containing the edge* $\{v_n, v_0\}$,
*x and y has a path containing **that path**.*
*If arbitrary vertices x and y has a path **containing** the edge* $\{v_n, v_0\}$,
*x and y has a path containing the **path*** $(v_0, v_1, \ldots, v_n)$.
*This is a **contradiction** that T has the **smallest** number of edges*
*and **connected**.*
$\therefore$ *T is acyclic.*
T is a tree.

9.8 Graph Coloring

Definition *1 A graph G is k-**colorable**, if each vertex can be assigned one of k colors so that **adjacent vertices** get the **different colors**.*

map coloring problem

*The smallest number of colors are called **chromatic number** of G, written as $\chi(G)$. $\chi(K_n) = n$.*

Theorem *1 The **chromatic** number of **planar** graph is no greater than 4.*

Theorem *2 A graph with maximum degree at most k is (k+1)-colorable.*

proof *Induction on number of **vertices**. P(n).*

basis *1-vertex graph, maximum degree 0 and 1-colorable. P(1) is true.*

induction *Let G be (n+1)-**vertex** graph with maximum degree at most k*

Remove *a vertex v and its incident edges.*

G' has n vertices and max. deg. at most k. $\therefore$ G' is (k+1)-colorable(IH).

v has at most k adjacent vertices. $\therefore$ G is (k+1)-colorable($(k+1) - k = 1$).

Bipartite graph

*Every **bipartite** graph is 2-colorable.*
*Every **path**, **tree**, and **even length cycles** are bipartite.*

Theorem *A graph is **bipartite** if and only if it contains **no odd lengh cycle**.*

9.7 Planar Graph

Definition *1 A graph is **planar**, if it can be drawn in the **plane** **without edge crossing**.*

K_4 is planar and Q_3 is planar, but $K_{3,3}$ is not planar.

***Region**: planar graph divides regions*

- ***Tree** has **one** region*
- ***Removing an edge** on the **cycle** merges **two** regions into **one**.*

Theorem *1 Let G be a* ***connected planar graph*** *with v* ***vertices*** *and e* ***edges****. Let r be the number of* ***regions*** *in planar reprsentation of G. Then*

$r = e - v + 2.$

proof *Induction on number of edges, P(e).*

basis *e=0, v=1, r=1.* $\therefore$ *1 = 0 – 1 + 2 = 1. O.K.*

induction *Consider a connected planar graph G with e+1 edges.*

1. If G is ***acyclic****. G is* ***tree****.* $\therefore$ *r = 1, e – v + 2 = –1 + 2 =1.* $\therefore$ *O.K.*

2. If G is not acyclic, G has at least one ***cycle****, C.*

Consider {u, v} ***in*** *C and a* ***spanning tree****, T. .϶. {u, v} is* ***not*** *in T.*

There exisits such an edge {u, v} because T is acyclic.

Remove *the* ***edge*** *{u, v} from G, it is called as G'.*

G' ***has one less regions****, since removing an edge {u, v} on the cycle C.*

G' is connected planar graph and has e edges.

$\therefore$ *r = e – v + 2 in G'* *induction hypothesis*

In G, r+1 regions, e+1 edges, and v vertices. $\therefore$ *(r +1) = (e+1) – v + 2.*

Corollary *1 If connected planar graph G with e edges and v vertices where $v \geq 3$. Then* $e \leq 3v - 6$.

proof *Consider degree of region number of boundary edges*

Sum of the degree of regions is 2e.

Minimum number of degree of each region r is 3r(triangle)

$\therefore 2e \geq 3r = 3(e - v + 2)$ $\therefore e \leq 3v - 6$.

Corollary *2 If G is a connected planar graph, then G has a vertex not exceeding five.*

proof *If G has one or two vertices, the result is true.*

If G has more that three vertices, $e \leq 3v - 6$. So $2e \leq 6v - 12$.

If every vertex has more than or equal 6, $2e \geq 6v$(Handshaking Theorem)

But it contradict with $2e \leq 6v - 12$.

Example *5 K_5 is not planar.*

$v = 5, e = 10. 3v - 6 = 9.$

9.5 Euler and Hamilton Path

Definition *1 Let G be a graph.*

*An **Euler path** is a path containing **every edge** of G.*

*An **Euler circuit** is a circuit containing **every edge** of G.*

Theorem *1 A connected graph has **Euler circuit** iff each vertex has **even degree**.*

proof *(→) The circuit contributes 2 to the degree of each node*

(←) Algorithm 1

Algorithm *1 Constructing Euler Circuit*

*Begins with **arbitrary** node.*

*Construct a **cycle** form the vertex to the vertex.*

*Repeat for each **remaining** subgraph,*

***inserting** the **new cycle** into the **original** one.*

Theorem *2 A connected graph has* ***Euler path*** *iff it has exaclty* ***two*** *vertices of* ***odd degree****.*
proof *One is the start, the other is the end.*

Definition *2 Let G be a graph.*
An ***Hamilton path*** *is a path traverse each vertex ib G exaclty* ***once****.*
An ***Hamilton circuit*** *s a circuit traverse each vertex ib G exaclty* ***once****.*

K_n *has a Hamilton circuit circuit*$(n \geq 3)$*.*

If a graph has a vertex of ***degree one****, there is* ***no Hamilon*** *circuit.*
Exactly ***two edges*** *incident to a vertex are in Hamaiton circuit.*

Trvaling Saleman
NP complete

9.6 Shortest-Path Problem

Let G = (V, E) be a graph, and f: E $\rightarrow \mathbf{R}^+$ *be a* ***cost*** *of the edges. Then (V, E, f) is an edge* ***weighted*** *graph or multigraph(**N**) in this text.*

Iterative *definition(extension) of* ***cost*** *of* ***path***.

Let $(v_0, v_1, \ldots, v_n) \in E^*$ *be a path of length* $n \geq 0$*. Then we define*

$$\text{if } n = 0,\ f^*(v_0, v_0) = 0.$$

$$\text{if } n \geq 1,\ f^*(v_0, v_n) = \sum_{j=0}^{n-1} f(v_j, v_{j+1})$$

Recursive *definition(extension) of* ***cost*** *of* ***path***.

$$f^*(u, u) = 0.$$

$$f^*(u, v) = f(u, x) + f^*(x, v) \quad \text{or} \quad f^*(u, x) + f(x, v).$$

We may use f instead of f^**, since* $f \subseteq f^*$.

$f: E^* \to \mathbf{R}^+$ **extend** *the domain of f from E to* E^*.

Shortest path problem

Let G = (E, V, f) be an edge weighted graph.

Definition $min_f(u, v) = (u, v_1, \ldots, v_n, v) \in E^*$ $n \geq 0$.϶.

$f(u, v_1, \ldots, v_n, v) \leq f(u, u_1, \ldots u_m, v)$.϶. $^\forall m \geq 0$ $^\forall(u, u_1, \ldots, u_m, v) \in E^*$.

Find u, v $\in$ *V, find a shortest path such that f(u, v) is minimum.*

Definition *Let* $W \subseteq V$, $u, v \in V$. $L_f^W(u, v) = (u, w_1, \ldots w_n, v)$.϶.

$f(u, w_1, \ldots w_n, v) \leq f(u, x_1, \ldots, x_m, v)$.϶. $\forall m, (u, x_1, \ldots, x_m, v) \in E^*$

$n, m \geq 0, 1 \leq \forall i \leq n, w_i \in W, 1 \leq \forall i \leq m, x_i \in W.$

Note that u, v may be in W or not.

Theorem *Let* $W \subseteq V$, $u \in V$, *and* $x \notin W$.϶. $L_f^{W}(u, x)$ *is the* **minimum**.

$$\forall y \notin W,\ L_f^{W \cup \{x\}}(u, y) = \min(L_f^{W}(u, y),\ L_f^{W}(u, x)+f(x, y)).$$

procedure *Shortest path* $G = (V, E \subseteq V \times V, f\colon E \to \mathbf{R}^{+})$

for $w \in V$ **do** $L(w) := \text{Infinite}$ **od**; $L(u), W := 0, \varnothing$;

Initialization $\forall y \in V, L(y) = L_f^{\{\}}(u, y)$.

do $v \notin W \to$

Loop invariance $\forall y \in V, L(y) = L_f^{W}(u, y)$.

$x := x \notin W \wedge \min(L(x))$; $W := W \cup \{x\}$;

Loop invariance, invalidated $\forall y \in V, L(y) = L_f^{W-\{x\}}(u, y)$.

do $y \notin W \wedge (x, y) \in E \to L(y) := \min(L(y), L(x) + f(x, y))$ **od**

Loop invariance, validated $\forall y \in V, L(y) = L_f^{W}(u, y)$.

od

After the loop termination $\forall y \in V, L(y) = L_f^W(u, y) \wedge v \in W$.

We should also prove that $L_f^W(u, v) = min_f(u, v)$, *if* $v \in W$.

But it is trivial, since we add the shortest path vertex x to W.