Computer Science and Information Engineering
National Chi Nan University

Combinatorial Optimization

Dr. Justie Su-Tzu Juan

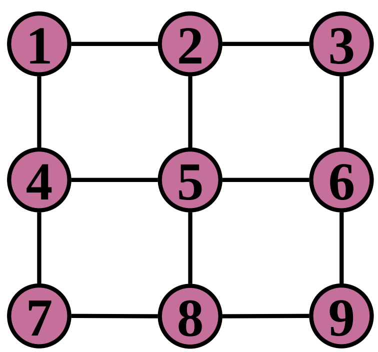
Lecture 2. Mathematical Preliminaries

§ 2.1 Graphs and digraphs

Slides for a Course Based on the Text
Combinatorial Optimization -
Networks and Matroids
by Eugene Lawler

2.1 Graphs and digraphs

- **Def:**
  - **A graph is an ordered pair $G = (V, E)$, where**
    $V(G) = V$ is a finite non-empty set of vertices (or nodes) and
    $E(G) = E$ is a set of un-ordered pairs of vertices, called edges (or links).
  - **If $\{x, y\}, \{y, z\} \in E$, then we say**
    1. x and y are adjacent;
    2. x is incident to $\{x, y\}$;
    3. $\{x, y\}$ and $\{y, z\}$ are adjacent;
    4. denote $\{x, y\}$ by (x, y) or xy.
- **Ex.**



⟺

$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$E = \{12, 23, 14, 25, 36, 45, 56, 47, 58, 69, 78, 89\}$

2.1 Graphs and digraphs

- **<u>Def</u>:**
  - **A digraph or directed graph is an ordered pair $D = (V, A)$, where**
    $V(D) = V$ is a finite non-empty set of vertices (or nodes) and
    $A(D) = A$ is a set of ordered pairs of vertices, called arcs (or edges).
  - **If $(x, y) \in A$, then we say**
    1. (x, y) are incident from x and incident to y.
    2. x is adjacent to y, y is adjacent from x.

2.1 Graphs and digraphs

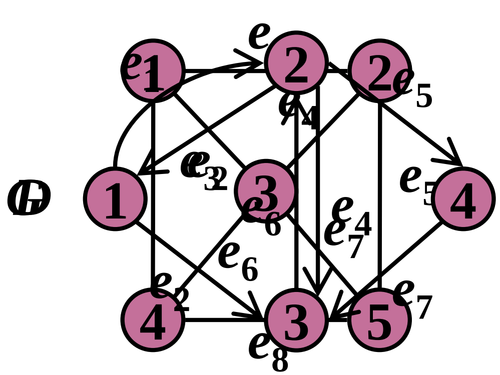
- <u>Def</u>:
  - The incidence matrix of a graph G, $M(G) = (b_{ik})$, is defined as follows:

$$b_{ik} = \begin{cases} 1, \text{ if node } i \text{ is incident to edge } k, \\ 0, \text{ otherwise.} \end{cases}$$

  - In the case of a digraph, the incidence matrix, $M(G) = (b_{ik})$ is defined as follows:

$$b_{ik} = \begin{cases} -1, \text{ if arc } k \text{ is incident to node } i, \\ 1, \text{ if arc } k \text{ is incident from node } i, \\ 0, \text{ otherwise.} \end{cases}$$

- Ex.



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{pmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 1 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix} \end{array}$$

2.1 Graphs and digraphs

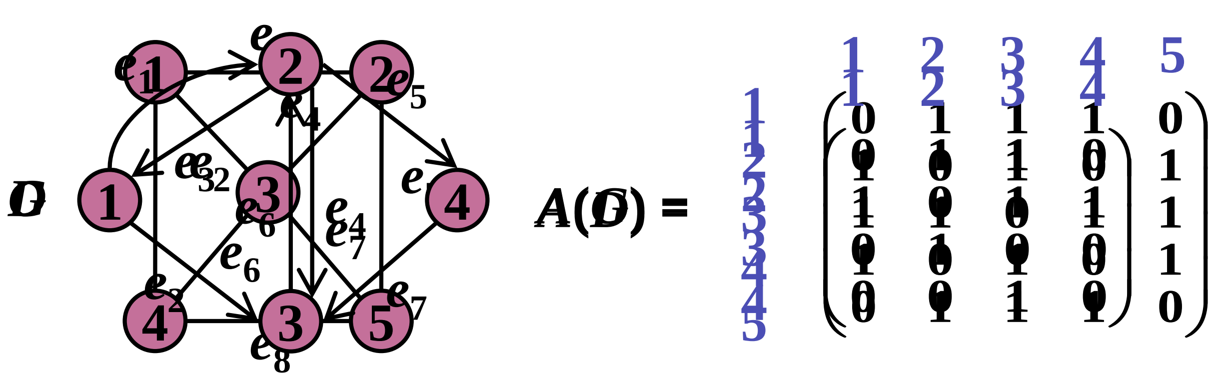
- **<u>Def</u>:**
  - **The adjacency matrix of a graph G, $A(G) = (a_{ij})$, is defined as :**

$$a_{ij} = \begin{cases} 1, \text{ if node } i \text{ is adjacent to node } j, \\ 0, \text{ otherwise.} \end{cases}$$

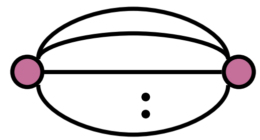
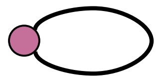
  - **The adjacency matrix of a digraph D, $A(G) = (a_{ij})$ is defined as :**

$$a_{ij} = \begin{cases} 1, \text{ if there is an arc } (i, j) \text{ from node } i \text{ to node } j, \\ 0, \text{ otherwise.} \end{cases}$$

- **Ex.**



2.1 Graphs and digraphs

- **Def:**
  - **multiple edges:** 
  - **loop:** 
  - **simple graph: a graph with no multiple edges and loop.**
    (本課程中若無特別說明，所稱 graph 皆指 simple graph)

- **Def:**
  - **In a graph $G = (V, E)$, $\forall x \in V$:**
    1. **neighbor of x: $N(x) = \{y \mid xy \in E\}$**
    2. **closed neighbor of x: $N[x] = \{x\} \cup N(x)$**
    3. **degree of x: $deg(x) = |N(x)|$**

2.1 Graphs and digraphs

- **Def:**
  - **In a digraph $D = (V, A)$, $\forall x \in V$:**

    1. the out-neighbor of $x \equiv N^+(x) = \{y \in V \mid \exists (x, y) \in A\}$

    2. the in-neighbor of $x \equiv N^-(x) = \{y \in V \mid \exists (y, x) \in A\}$

    3. the out-degree of $x \equiv deg^+(x) = |N^+(x)|$

    4. the in-degree of $x \equiv deg^-(x) = |N^-(x)|$

- **Note:**

  1. For incidence matrix of a graph G, $M(G) = (b_{ik})$:

  $$deg(i) = \sum_k b_{ik}.$$

  2. For incidence matrix of a digraph D, $M(D) = (b_{ik})$:

  $$deg^+(i) - deg^-(i) = \sum_k b_{ik}.$$

2.1 Graphs and digraphs

- **Def: $G = (V, E)$ is a graph, $x, y \in V$**
  - **An x-y walk with length r is a sequence $x_0, x_1, \ldots, x_r$ such that $x = x_0$, $y = x_r$, and $x_{i-1}x_i \in E$, $\forall\ 1 \le i \le r$.**
  - **An x-y trail is an x-y walk in which all edges are distinct.**
  - **An cycle is an x-y walk in which all vertices are distinct except $x = y$.**
  - **An x-y path is an x-y walk in which all vertices are distinct.**

- **Def: A graph $G = (V, E)$ is connected if $\forall\ x, y \in V$, $\exists\ x$-y path (or x-y walk) in G.**

- **Remark: For any two vertices x, y of G, $\exists\ x$-y walk $\Rightarrow \exists\ x$-y path.**

2.1 Graphs and digraphs

- **<u>Def</u>: Given graph $G = (V, E)$, define relation ~ on V by $x \sim y$ iff $\exists$ x-y walk.**

- **<u>Note</u>:**
  - **~ is an equivalent relation, i.e.**

    (1) $x \sim x, \forall x \in V$;

    (2) $x \sim y \Rightarrow y \sim x, \forall x, y \in V$;

    (3) $x \sim y$ and $y \sim z \Rightarrow x \sim z, \forall x, y, z \in V$
  - **Say the equivalence classes are $V_1, V_2, \ldots, V_r$, i.e.** $V = \dot{\bigcup}_{1 \le i \le r} V_i$

2.1 Graphs and digraphs

- **Def:**
  - **A graph $H = (U, F)$ is a subgraph of a graph $G = (V, E)$ iff $U \subseteq V$ and $F \subseteq E$.**
  - **For a graph $G = (V, E)$ and a set $S \subseteq V$:**

    A subgraph of G induced by S is the graph $G[S]$ or G_S whose vertex set is S and edge set is $\{xy \in E \mid x, y \in S\}$.

- **Def: A (connected) component of a graph G is the subgraph of G induced by an equivalent class of ~.**

- **Remark: If G has r components $G[V_i]$, $1 \leq i \leq r$, then $V = \dot{\bigcup}_{1\leq i\leq r} V_i$ and $E = \dot{\bigcup}_{1\leq i\leq r} E_i$, where E_i is the edge set of $G[V_i]$.**

2.1 Graphs and digraphs

- **Def:**
  - **The complete graph K_n is a graph with n nodes and**

    $$\forall\, x, y \in V(K_n),\ \{x, y\} \in E(K_n)$$
  - **A graph $G = (V, E)$ is called a bipartite graph $G = (S, T, E)$ ≡**

    (1) $V = S \cup T$ and $S \cap T = \phi$;

    (2) $\forall\ (x, y) \in E$, either $x \in S, y \in T$; or $x \in T, y \in S$.
  - **The complete bipartite graph $K_{p,q}$ is a bipartite graph $G = (S, T, E)$ with $|S| = p$, $|T| = q$, and $|E| = pq$.**

- **Def:**
  - **A complete subgraph ≡ a subgraph of G is a complete graph.**
  - **A maximal complete subgraph is called a clique.**

Computer Science and Information Engineering
National Chi Nan University

Combinatorial Optimization

Dr. Justie Su-Tzu Juan

Lecture 2. Mathematical Preliminaries

§ 2.2 Trees

Slides for a Course Based on the Text
Combinatorial Optimization -
Networks and Matroids
by Eugene Lawler

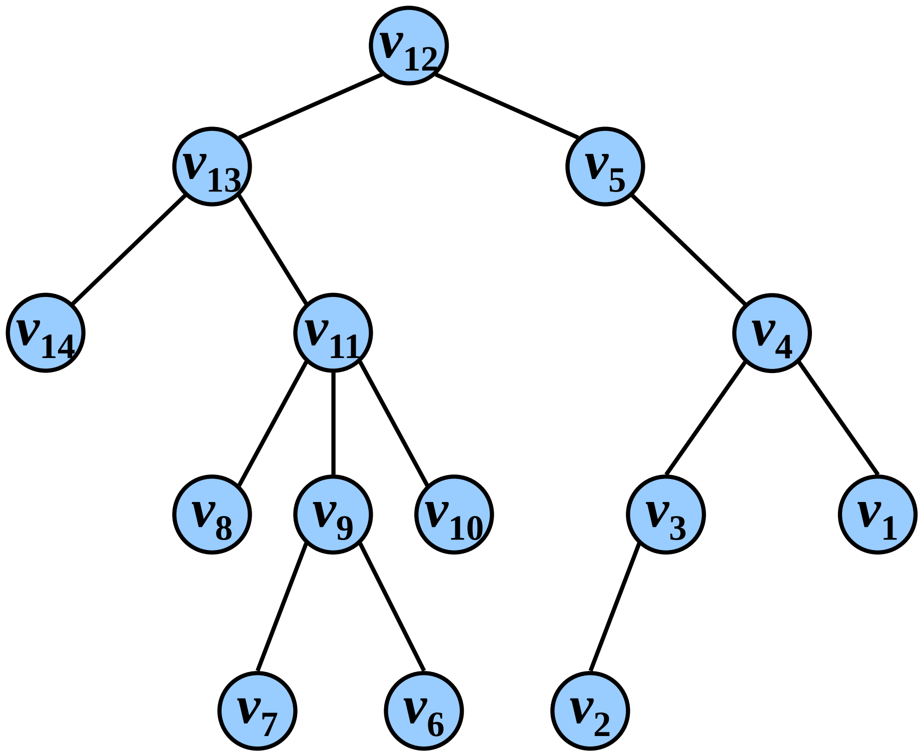
2.2 Trees

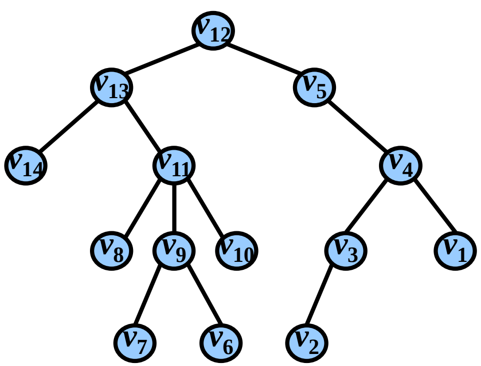
- **Def: A tree is a connected acyclic (i.e. without cycle) graph.**

- **Thm: The following statements are equivalent (TFSAE), for a graph $G = (V, E)$:**
  (1) G is a tree.
  (2) G is connected and $|E| = |V| - 1$.
  (3) G is acyclic and $|E| = |V| - 1$.
  (4) $\forall$ x, y in G, $\exists$ unique x-y path in G.
  (5) We can order V into $v_1, v_2, \ldots, v_n$ such that v_i is a leaf (a vertex with degree 1) of $G[\{v_i, v_{i+1}, \ldots, v_n\}]$ for $1 \le i \le n - 1$, called tree ordering.

2.2 Trees

- **Ex:**


v_{12}
v_{13}
v_5
v_{14}
v_{11}
v_4
v_8
v_9
v_{10}
v_3
v_1
v_7
v_6
v_2


v_{12}
v_{13}
v_5
v_{14}
v_{11}
v_4
v_8
v_9
v_{10}
v_3
v_1
v_7
v_6
v_2

2.2 Trees

<u>Lemma T1</u>: Every tree with at least 2 vertices has at least 2 leaves.

Proof.

Let $T = (V, E)$ is a tree with $|V| \geq 2$.

Choose a path P: $x_0, x_1, \ldots, x_r$ in the tree such that r is maximum.

$\because$ $|V| \geq 2$ and T is connected $\therefore$ $r \geq 1$.

$\because$ $x_1 \in N(x_0)$ $\therefore$ $deg(x_0) \geq 1$.

Suppose $deg(x_0) \geq 2$, then $\exists\, x \neq x_1$ such that $xx_0 \in E$

<u>Case 1</u>: x = some x_j, for some $j \geq 2$, $\exists$ cycle $x_0, x_1, \ldots, x_j, x_0$ $\rightarrow\leftarrow$

<u>Case 2</u>: x is not in the path P

Then $x, x_0, x_1, \ldots, x_r$ is a larger path in T contradicting to the choice of the path P $\rightarrow\leftarrow$

Therefore, $deg(x_0) = 1$, i.e. x_0 is a leaf.

Similarly, x_r is a leaf.

2.2 Trees

- **<u>Note</u>: Every acyclic graph with at least 2 vertices and one edge has at least 2 leaves.**

- **<u>Notation</u>: $G = (V, E)$, $S \subseteq V$, $x \in V$:**
  - $G - S = G[V - S]$
  - $G - x = G[V - \{x\}]$

2.2 Trees

<u>Lemma T2</u>: If x is a leaf of a tree $T = (V, E)$, then $T - x$ is a tree.

Proof.

A: $\because$ T is acyclic $\Rightarrow T - x$ is acyclic

B: $\forall$ two vertices y, z in $T - x$

$\Rightarrow y \neq x$ and $z \neq x$, and $y, z \in V(T)$

$\therefore \exists\ y - z$ path in T say P: $y = x_0, x_1, \ldots, x_r = z$

Case 1: $x \in P$, i.e. $x = x_i$ for some i $(0 < i < r)$

$\because |\{x, y, z\}| = 3 \quad \therefore r \geq 2$

Then $x_{i-1}, x_{i+1} \in N_T(x_i)$

$\Rightarrow deg(x_i) \geq 2 \quad \rightarrow\leftarrow$ ($\because x_i = x$ is a leaf)

Case 2: $x \notin P \quad \therefore P$ is a y-z path in $T - x$

Then, $T - x$ is connected.

By A, B, $T - x$ is a tree.

Computer Science and Information Engineering
National Chi Nan University

Combinatorial Optimization

Dr. Justie Su-Tzu Juan

Lecture 2. Mathematical Preliminaries

§ 2.3 Linear Programming

Slides for a Course Based on the Text
1. Combinatorial Optimization
by Cook, Cunningham, Pulleyblank and Schrijver
2. Combinatorial Optimization - Algorithms and Complexity
by Papadimitriou and Steiglitz

2.3 Linear Programming

- **Def:**

  1. Let A be an $m \times n$ matrix, and let $b \in \mathbb{R}^m$. A linear programming (LP) problem is

  $$\begin{cases} \text{Maximize } c^T x & \text{(A.1)} \\ \text{subject to } Ax \leq b \end{cases}$$

  that is, to determining: $\max\{c^T x : Ax \leq b\}$ (A.2)

  2. x is a feasible solution of (A.2) if x satisfies $Ax \leq b$.

  3. x is called an optimum(optimal) solution of (A.2) if x is a feasible solution and attains the maximum.

  4. $c^T = (c_1, c_2, \ldots, c_n)$ is the cost vector.

  $c^T x$ is the objective function.

  $A = (a_{ij})$ is an $m \times n$ coefficient matrix

  $b^T = (b_1, b_2, \ldots, b_m)$ is the constraint vector

  $a_i = (a_{i1}, a_{i2}, \ldots, a_{in})$, $A_j^T = (a_{1j}, a_{2j}, \ldots, a_{mj})$

  5. The dual LP problem of (A.2) is:

  $$\min\{y^T b : y \geq 0, y^T A = c^T\} \text{ where } y \in \mathbb{R}^m.$$

2.3 Linear Programming

- **Ex:**

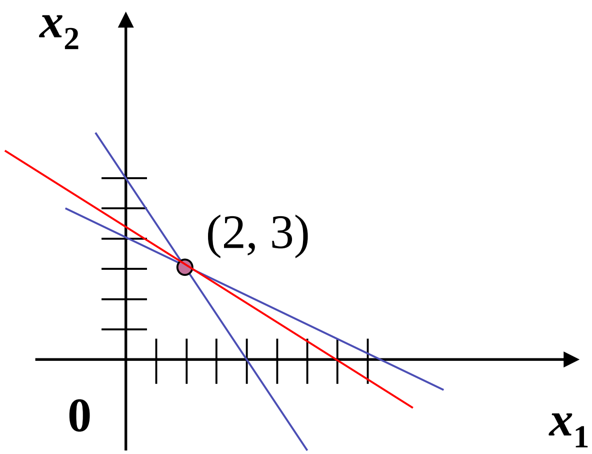
1. max $2x_1 + 3x_2$

subject to $\begin{cases} x_1 + 2x_2 \leq 8 \\ 3x_1 + 2x_2 \leq 12 \end{cases}$

$\Rightarrow$ when $x_1 = 2$, $x_2 = 3$

$2x_1 + 3x_2 = 4 + 9 = 13$ is max.

i.e. (2, 3) is an optimum solution.

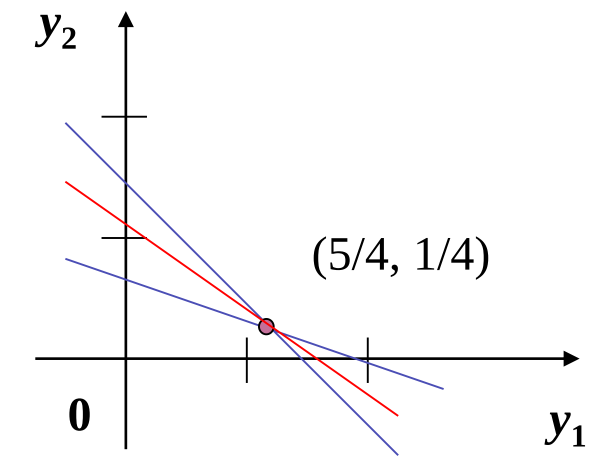


2. min $8y_1 + 12y_2$

subject to $\begin{cases} y_1 + 3y_2 = 2 \\ 2y_1 + 2y_2 = 3 \\ y_1 \geq 0 \\ y_2 \geq 0 \end{cases}$

$\Rightarrow$ when $y_1 = 5/4$, $y_2 = ¼$.

$8y_1 + 12y_2 = 10 + 3 = 13$ is min.



2.3 Linear Programming

- **Def:**
- **Theorem A.4: (Weak Duality Theorem)**

  Let A be an $m \times n$ matrix, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^m$. Suppose $\tilde{x}$ is a feasible solution to $Ax \leq b$ and $\tilde{y}$ is a feasible solution to $y \geq 0$, $y^TA = c^T$. Then $c^T\tilde{x} \leq \tilde{y}^Tb$.

Proof.

$$c^T\tilde{x} = (\tilde{y}^TA)\,\tilde{x} = \tilde{y}^T(A\tilde{x}) \leq \tilde{y}^Tb.$$

2.3 Linear Programming

- **Theorem A.5: (Duality Theorem)**
  Let A be an $m \times n$ matrix, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^m$. Then
  $$\max\{c^T x: Ax \le b\} = \min\{y^T b: y \ge 0, y^T A = c^T\}$$
  provided that both sets are nonempty.

- **Def: Given an LP in general form, called the primal, the dual is defined as follows:**

Primal		Dual
max $c^T x$		min $y^T b$
s.t. $a_i^T x = b_i$	$i \in M$	s.t. y_i unrestricted
$a_i^T x \le b_i$	$i \in \overline{M}$	$y_i \ge 0$
$x_j \ge 0$	$j \in N$	$y^T A_j \ge c_j$
x_j unrestricted	$j \in \overline{N}$	$y^T A_j = c_j$

2.3 Linear Programming

- **Theorem 3.1: If an LP has an optimal solution, so does its dual, and at optimality their costs are equal.**

- **Theorem 3.2: The dual of the dual is the primal.**

Computer Science and Information Engineering
National Chi Nan University

Combinatorial Optimization

Dr. Justie Su-Tzu Juan

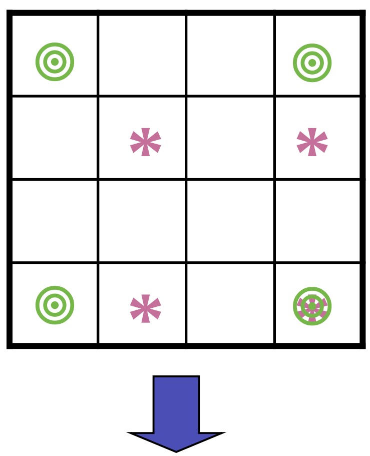
Lecture 2. Mathematical Preliminaries

§ 2.4 Domination Problem

Slides for a Course Based on the Text
Combinatorial Optimization
by Cook, Cunningham, Pulleyblank and Schrijver

2.4 Domination Problem

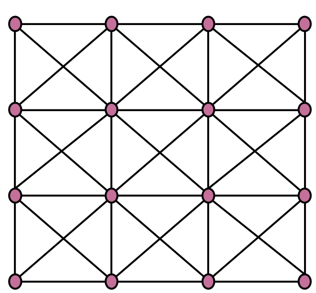
- **Source: <u>chessboard problem</u>, firehouse problem, location problem.**
- **Ex: In $m \times n$ chessboard, need $\lceil m/3 \rceil \times \lceil n/3 \rceil$ kings to "dominate" all.**



(1) $x \le \lceil m/3 \rceil \lceil n/3 \rceil$: 找一個方法 (easy)

(2) $x \ge \lceil m/3 \rceil \lceil n/3 \rceil$: 找最多格子使得沒有任兩格可能被同一個kings控制到

Primal-dual



格子 → vertex

可控制 → edge

2.4 Domination Problem

- **Def:**
  - **Given a graph $G = (V, E)$, a dominating set of G is a subset $D \subseteq V$, such that** $V = \bigcup_{x \in D} N[x]$
  - **The domination number of G: $\gamma(G) = \min\{|D|: D$ is a dominating set of $G\}$.**