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

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

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

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

2.3 Linear Programming

- **Ex:**

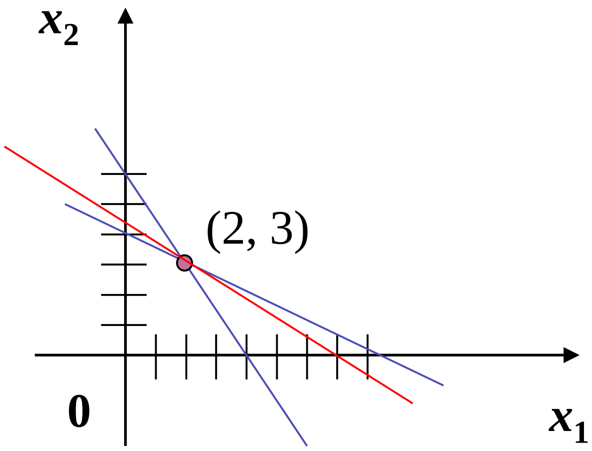
  1. max $2x_1 + 3x_2$

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

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

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

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

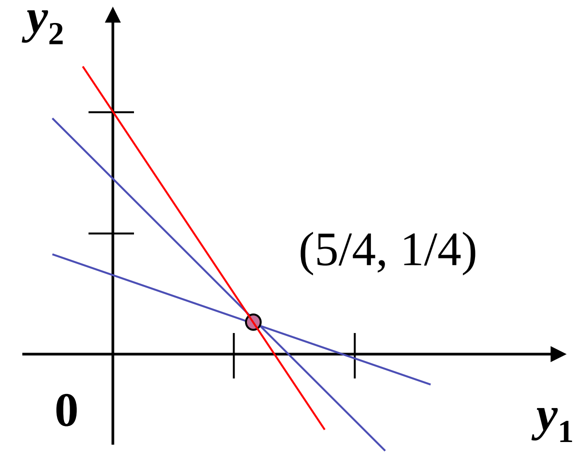


  2. min $8y_1 + 12y_2$

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

  $\Rightarrow$ 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 $y \geq 0$, $y^T A = c^T$. Then $c^T \tilde{x} \leq \tilde{y}^T b$.

Proof.

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

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

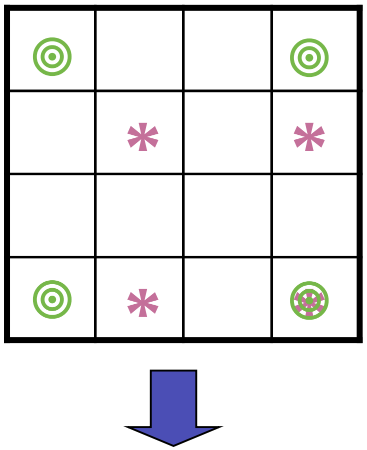
Lecture 2. Mathematical Preliminaries

§ 2.4 Domination Problem

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

2.4 Domination Problem

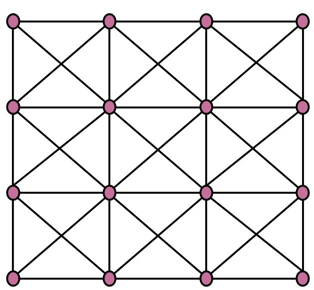
- Source: chessboard problem, firehouse problem, location problem.
- Ex: In $m \times n$ chessboard, need $\lceil m/3 \rceil \times \lceil n/3 \rceil$ kings to "dominate" all.



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

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

Primal-dual



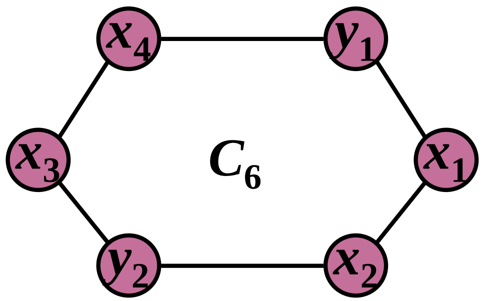
格子 → vertex

可控制 → edge

2.4 Domination Problem

- **Def: A dominating set of a graph $G = (V, E)$ is a subset D of V such that $\forall\, x \in V - D$, $\exists\, y \in D$ with $xy \in E$.**

- **Ex.**



Let $D_1 = \{y_1, y_2\}$, $D_2 = \{x_1, y_2, x_4\}$, then D_1, D_2 are dominating sets of C_6.

- **Notation: domination number of G $\gamma(G) = \min\{|D| : D$ is a dominating set of $G\}$.**

- **The Minimum Dominating Set Problem in general graph G is NP-complete.**

Computer Science and Information Engineering
National Chi Nan University

Combinatorial Optimization

Dr. Justie Su-Tzu Juan

Lecture 3. Domination Problem in Tree

§3.1 Method 1 : *L*-dominating

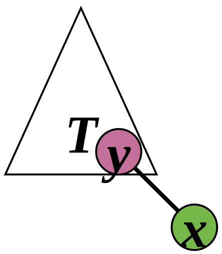
Slides for a Course Based on the Paper
E. J. Cockayne, S. E. Goodman and S. T. Hedetniemi, *“A linear algorithm for the domination number of a tree”*, Inform. Process. Lett. 4 (1975), 41-44.

3.1 Method 1: *L*-dominating

- **<u>Note</u> :**

  Suppose D is an optimal dominating set of tree T.

  ∀ x: a leaf, adjacent to vertex y :



  ① $y \in D$ ⇨ $x \notin D$

  (o.w. D–{x} is also a dominating set with |D–{x}| < |D| →←)

  ② $y \notin D$ ⇨ $x \in D$

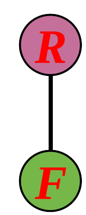
  ⇨ let D' = (D–{x})∪{y} is also a dominating set as $N[x] \subseteq N[y]$

  So, <u>always ∃ an optimal dominating set D such that $y \in D$ and $x \notin D$, for any leaf x adjacent to vertex y</u>

  ⇒ <u>Step 1</u>: 配合 "bound" vertex $N[y]$ *B*

  <u>Step 2</u>: 產生 "free" vertex x *F*

  <u>Step 3</u>: 再產生 "required" vertex y *R*



3.1 Method 1: L-dominating

- **<u>Def</u>: $G = (V, E)$ is a graph, $L : V \to \{B, R, F\}$, (i.e., each vertex x has a label $L(x) \in \{B, R, F\}$.)**

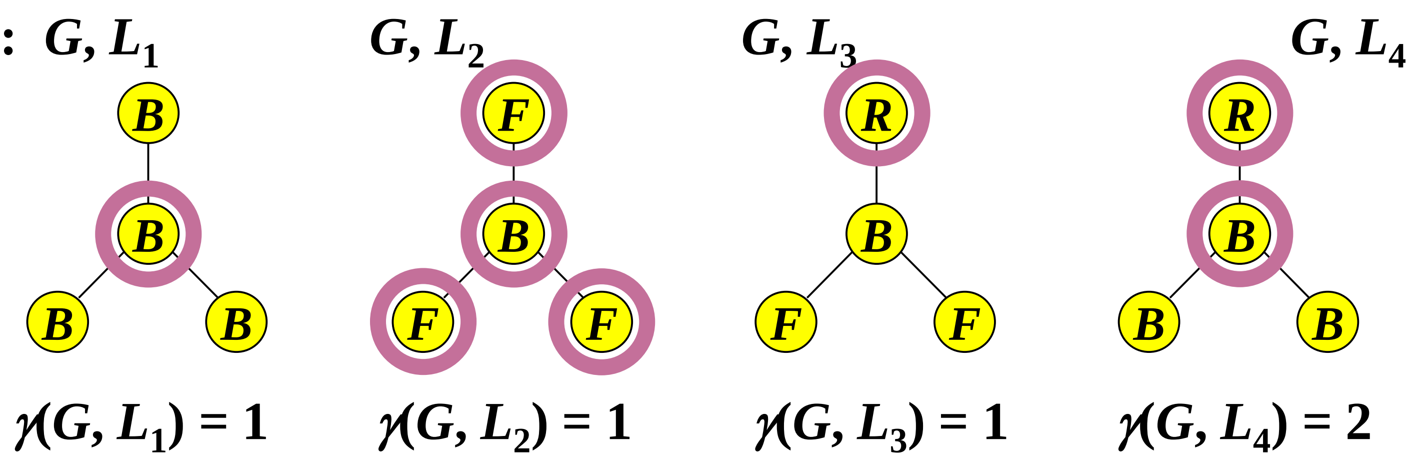
  An L-dominating (mixed dominating) set of G is a subset $D \subseteq V(G)$

  s.t. ① $L(x) = R \Rightarrow x \in D$

  ② $L(x) = B \Rightarrow N[x] \cap D \neq \phi$

- **<u>Notation</u>: $\gamma(G, L) = \min\{|D| : D$ is a L-dominating set of $G\}$.**

- **<u>Ex</u>:**



3.1 Method 1: *L*-dominating

- **Note**: If $L(v) = B$, $\forall\, v \in V(G)$, an L-dominating set = dominating set and $\gamma(G) = \gamma(G, L)$.

- **Thm**: A tree T has a leaf x adjacent to vertex y, let $T' = T - x$ then
  (1) $L(x) = F \Rightarrow \gamma(T, L) = \gamma(T', L)$
  (2) $L(x) = B \Rightarrow \gamma(T, L) = \gamma(T', L')$ where $L'(v) = \begin{cases} L(v), & \forall\, v \neq y \\ R, & v = y \end{cases}$
  (3) $L(x) = L(y) = R \Rightarrow \gamma(T, L) = \gamma(T', L)+1$
  (4) $L(x) = R$, but $L(y) \neq R \Rightarrow \gamma(T, L) = \gamma(T', L')+1$ where
  $$L'(v) = \begin{cases} L(v), & \forall\, v \neq y \\ F, & v = y \end{cases}$$

3.1 Method 1: L-dominating

- **Thm: A tree T has a leaf x adjacent to vertex y, let $T' = T - x$ then**
  (1) $L(x) = F \Rightarrow \gamma(T, L) = \gamma(T', L)$

Proof.

Choose a L-dominating set D of $T - x$ such that $|D| = \gamma(T', L)$,
then D is also a L-dominating set of T.
Hence $\gamma(T, L) \le |D| = \gamma(T', L)$.

Choose a L-dominating set D of T such that $|D| = \gamma(T, L)$,

Case 1: $x \notin D \Rightarrow D$ is also a L-dominating set of $T - x$
$\Rightarrow \gamma(T, L) = |D| \ge \gamma(T', L)$.

Case 2: $x \in D$, consider $D' = (D - \{x\}) \cup \{y\}$
D' is a L-dominating set of $T - x$
($\forall v \in V(T - x)$: 1. $v \ne y, L(v) = B : \exists u \in D - \{x\} \subseteq D'$, s.t. $uv \in E(T)$.
2. $v \ne y, L(v) = R : \because v \in D \Rightarrow v \in D' = (D - \{x\}) \cup \{y\}$.
3. $v = y : \because y \in D', \therefore N[y] \cap D' \ne \phi$.)
Also, $|D'| \le |D|$, $\gamma(T, L) = |D| \ge |D'| \ge \gamma(T', L)$.

So, $\gamma(T, L) = \gamma(T', L)$.

3.1 Method 1: L-dominating

- **Thm**: A tree T has a leaf x adjacent to vertex y, let $T' = T - x$ then (4) $L(x) = R$, but $L(y) \neq R \Rightarrow \gamma(T, L) = \gamma(T', L')+1$ where

$$L'(v) = \begin{cases} L(v), & \forall\, v \neq y \\ F, & v = y \end{cases}$$

Proof.

Let D be a minimum L'-dominating set of $T - x$.

$D \cup \{x\}$ is a L-dominating set of T,

$\therefore \gamma(T, L) \leq |D| + 1 = \gamma(T', L') + 1.$

Let D is a minimum L-dominating set of T. By definition, $x \in D$.

$\because L'(y) = F \Rightarrow \therefore$ no matter $y \in D$ or $y \notin D$,

$D' = D - \{x\}$ is a L'-dominating set of $T - x$.

$\gamma(T', L') \leq |D'| = |D| - 1 = \gamma(T, L) - 1.$

Hence $\gamma(T, L) = \gamma(T', L') + 1.$

3.1 Method 1: *L*-dominating

- **Algorithm 3.1:**

```
Given tree ordering [x_1, x_2, …, x_n] of T
D ← φ;
for i = 1 to n do L(x_i) = B.
for i = 1 to n – 1 do
    choose j > i such that x_i x_j ∈ E;
    if (L(x_i) = B) then L(x_j) ← R;
    if (L(x_i) = R) then
        D ← D ∪ {x_i};
        if (L(x_j) = B) then L(x_j) ← F;
if (L(x_n) ≠ F) then D ← D ∪ {x_n};
```

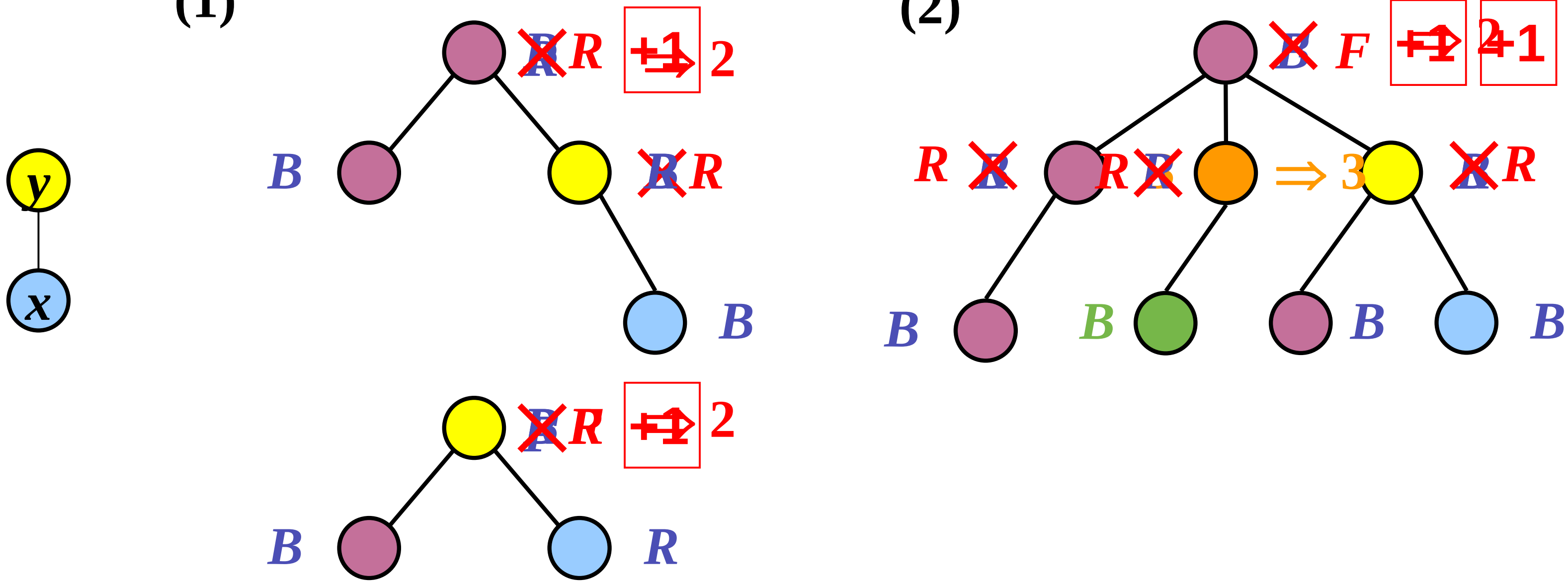
- **Time complexity =** $\mathcal{O}(n)$**.**

3.1 Method 1: *L*-dominating

- **Ex:**

(1)

(2)



3.1 Method 1: *L*-dominating

- **Exercise 1 (3/15): Develop a labeling algorithm for the vertex covering program on a tree.**

- **Def: A vertex cover of a graph $G = (V, E)$ is a subset $C \subseteq V$ such that $\forall$ edge $xy \in E$, $\{x, y\} \cap C \neq \phi$.**

- **Notation: $\beta(G) = \min\{|C|: C$ is a vertex cover of $G\}$**

Computer Science and Information Engineering
National Chi Nan University

Combinatorial Optimization

Dr. Justie Su-tzu Juan

Lecture 3. Domination Problem in Tree

§3.2 Method 2 : Primal-dual method

3.2 Method 2 : Primal-dual method

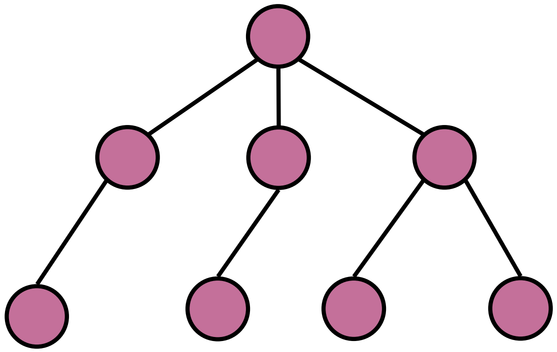
- **Def:** I is a **2-stable set** in a graph $G(V, E)$ if

$$\forall\, x \neq y \text{ in } I,\ d(x, y) > 2$$

- **Def:** $\gamma(G) = \min\{|D| : \bigcup_{x \in D} N[x] = V\}$

$$\alpha_2(G) = \max\{|I| : I \text{ is a 2-stable set}\}$$

- **Ex:**



3.2 Method 2 : Primal-dual method

- **Weakly duality inequality: $\forall$ 2-stable set I, $\forall$ dominating set D,**

$$|I| \le |D|.$$

Proof.

Define $f: I \to D$ by

$f(x)$ = some y with $x \in N[y]$ for some $y \in D$

Suppose $x, x' \in I$, $f(x) = y = y' = f(x')$

$d(x, y) \le 1$, $\because$ $x \in N[y]$

$d(x', y') \le 1$, $\because$ $x' \in N[y']$

$d(x, x') \le d(x, y) + d(x', y) = d(x, y) + d(x', y') \le 2.$

By the fact that I is a 2-stable set, $x = x'$

Hence f is 1-1 function

So $|I| \le |D|$.

- **Corollary:** $\alpha_2(G) \le \gamma(G)$

3.2 Method 2 : Primal-dual method

- **Note: If we find a 2-stable set I^* and a dominating set D^* such that $|D^*| = |I^*|$,**

  then $|I^*| \leq \alpha_2(T) \leq \gamma(T) \leq |D^*| = |I^*|$.

  That imply all "≤" are "=". i.e.

  1. I^* is a maximum 2-stable set, $|I^*| = \alpha_2(G)$.

  2. D^* is a minimum dominating set, $|D^*| = \gamma(G)$.

  3. $\gamma(G) = \alpha_2(G) = |D^*| = |I^*|$.

3.2 Method 2 : Primal-dual method

- **Algorithm 3.2:**

Given tree ordering $[v_1, v_2, \ldots, v_n]$ of T

$D^* \leftarrow \phi;$

$I^* \leftarrow \phi;$

do $i = 1$ **to** n

  choose $j > i$ **which** $v_i v_j \in E$; (**for** $i = n$ **choose** $j = n$)

  if $N_T[v_i] \cap D^* = \phi$ **then** $\begin{cases} D^* \leftarrow D^* \cup \{v_j\}; \\ I^* \leftarrow I^* \cup \{v_i\} \end{cases}$

end

- **Time complexity =** $\mathcal{O}(n)$.

3.2 Method 2 : Primal-dual method

- **Thm:**

① D^*: dominating set:

∵ 若$N_T[v_i] \cap D^* = \phi$，則把v_j放入D^*中　∴ D^*為domination set

② $|D^*| \le |I^*|$:

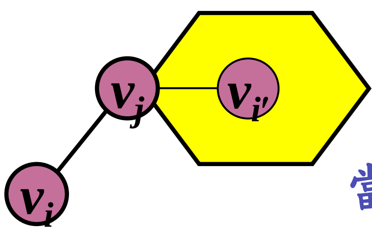
∵ 每次加v_j入D^*就加新的v_i入I^*　∴ $|I^*| \ge |D^*|$

③ I^* is 2-stable set:

If $\exists\, v_i, v_{i'} \in I^*$, such that $d(v_i, v_{i'}) \le 2$. W.L.O.G., say $i < i'$.

當 algorithm做到v_i時: 由 algorithm知，

當時將 v_j 放入 D^*, v_i 放入 I^*.



Since v_i is a leaf of $T[v_i, v_{i+1}, \ldots, v_n]$ and $v_iv_j \in E$,

$d(v_i, v_{i'}) \le 2 \Rightarrow d(v_j, v_{i'}) \le 1$, i.e. $v_j \in N[v_{i'}]$

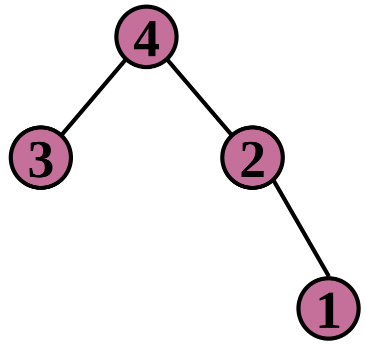
當 algorithm 做到 i'時: ∵$v_j \in D^*$，此時$N_T[v_{i'}] \cap D^* \ne \phi$

因此，不能將 $v_{i'}$放入 I^* 中 →←

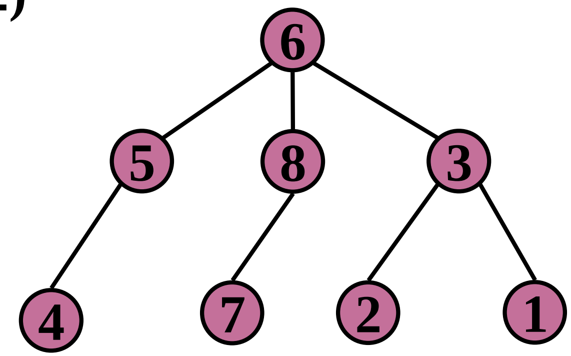
∴ I^* 為 2-stable set.

3.2 Method 2 : Primal-dual method

- **Ex:**
- **(1)**



(2)



$D^* = \{ \quad \}$

$I^* = \{ \quad \}$

$D^* = \{ \quad \}$

$I^* = \{ \quad \}$

choose $j > i$ which $v_i v_j \in E$; (for $i = n$ choose $j = n$)

if $N_T[v_i] \cap D^* = \phi$ then $D^* \leftarrow D^* \cup \{v_j\}$;

$I^* \leftarrow I^* \cup \{v_i\}$