- 若 $\boldsymbol{A}$ 為對稱矩陣，則 $\boldsymbol{A}^n$ 也是對稱矩陣。
- 若 $\boldsymbol{A}$ 為可逆上三角矩陣，則 $\operatorname{adj}(\boldsymbol{A})$ 和 $\boldsymbol{A}^{-1}$ 也是上三角矩陣。
- 若 $\boldsymbol{A}, \boldsymbol{B} \in F^{n\times n}$ 為可逆方陣，則 $\operatorname{adj}(\boldsymbol{AB}) = \operatorname{adj}(\boldsymbol{B})\operatorname{adj}(\boldsymbol{A})$。
- 若 $T \in \mathcal{L}(V, V')$，則
  - T 必保相依。
  - T 保獨立，即若 S 為線性獨立，則 $T(S)$ 亦是線性獨立 $\iff$ T 為一對一
  - T 保生成，即若 S 為 V 生成集，則 $T(S)$ 亦是 V 生成集 $\iff$ T 為映成
- $$\begin{aligned} &\boldsymbol{A} \in F^{m\times n} \text{ s.t. } \operatorname{rank}(\boldsymbol{A}) = 1 \\ \iff &\exists\ \boldsymbol{u} \neq \boldsymbol{0} \in F^{m\times 1},\ \boldsymbol{v} \neq 0 \in F^{1\times n},\ \text{s.t. } \boldsymbol{A} = \boldsymbol{uv} \end{aligned} \tag{1}$$
- 若 $\boldsymbol{A} \in F^{m\times n}, \boldsymbol{B} \in F^{n\times p}$，則

$$\operatorname{rank}(\boldsymbol{A}) + \operatorname{rank}(\boldsymbol{B}) - n \leq \operatorname{rank}(\boldsymbol{AB}) \tag{2}$$

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ -\boldsymbol{A} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{B} \\ \boldsymbol{A} & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{B} \\ \boldsymbol{O} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{AB} \end{bmatrix} = \boldsymbol{D} = \boldsymbol{ABC} \tag{3}$$

因為 $\boldsymbol{A}, \boldsymbol{B}$ 可逆，則

$$\operatorname{rank}\left(\begin{bmatrix} \boldsymbol{I} & \boldsymbol{B} \\ \boldsymbol{A} & \boldsymbol{O} \end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{AB} \end{bmatrix}\right) = n + \operatorname{rank}(-\boldsymbol{AB}) = n + \operatorname{rank}(\boldsymbol{AB}) \tag{4}$$

有 $\operatorname{rank}(\boldsymbol{A}) + \operatorname{rank}(\boldsymbol{B}) \leq n + \operatorname{rank}(\boldsymbol{AB})$ 得證。

- 若 $\boldsymbol{A}_1, \cdots, \boldsymbol{A}_k \in \mathbb{R}^{n\times n}$ 為方陣，且 $\boldsymbol{A}_1 \cdots \boldsymbol{A}_k = \boldsymbol{O}$，則

$$\operatorname{rank}(\boldsymbol{A}_1) + \operatorname{rank}(\boldsymbol{A}_2) + \cdots + \operatorname{rank}(\boldsymbol{A}_k) \leq (k-1)n \tag{5}$$

- 若 $T(x) = \boldsymbol{Ax}, T \in \mathcal{L}(V, V'), \boldsymbol{A} \in F^{m\times n}$，則
  - $\boldsymbol{A}$ 為一對一 $\iff$ $\boldsymbol{A}$ 有左反矩陣 $\iff$ $\operatorname{rank}(\boldsymbol{A}) = \dim(V) = n \leq \dim(V') = m$, $\mathrm{N}(\boldsymbol{A}) = \{\boldsymbol{0}\} \iff \boldsymbol{Ax} = \boldsymbol{0}$ 只有 $\boldsymbol{0}$ 唯一解 $\iff$ $\boldsymbol{A}$ 行獨立，列生成 $F^{1\times n} \iff \boldsymbol{Ax} = \boldsymbol{b} \leq 1$ 解 $\iff$ $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$ 可逆 $\iff$ $\boldsymbol{A}^{+} = (\boldsymbol{A}^{\intercal}\boldsymbol{A})^{-1}\boldsymbol{A}^{\intercal}$

  - $\boldsymbol{A}$ 為映成 $\iff$ $\boldsymbol{A}$ 有右反矩陣 $\iff$ $\operatorname{rank}(\boldsymbol{A}) = \dim(V') = m \leq \dim(V) = n \iff$ $\boldsymbol{A}$ 列獨立，行生成 $F^{m\times 1} \iff$ $\boldsymbol{Ax} = \boldsymbol{b} \geq 1$ 解 $\iff$ $\boldsymbol{AA}^{\mathsf{H}}$ 可逆 $\iff$ $\boldsymbol{A}^{+} = \boldsymbol{A}^{\intercal}(\boldsymbol{AA}^{\intercal})^{-1}$

- 若 $\boldsymbol{A}, \boldsymbol{B}$ 為方陣且 $\boldsymbol{A} \sim \boldsymbol{B}$，則 $\boldsymbol{A}$ 與 $\boldsymbol{B}$ 的
  - tr
  - det
  - rank
  - nullity
  - 特徵多項式
  - 特徵根
  - 喬丹型

  皆**相等**，反之不然。但**特徵向量**不保證相同，且僅**喬丹型**為充要條件。

- 若 $\boldsymbol{A}, \boldsymbol{B}$ 為方陣，則 $\boldsymbol{AB}$ 與 $\boldsymbol{BA}$ 有相同的
  - 特徵根
  - 特徵多項式

  若 $\boldsymbol{A}, \boldsymbol{B}$ **不**為方陣，只能保證 $\boldsymbol{AB}$ 與 $\boldsymbol{BA}$ 有相同的**非零**特徵根。

  Proof.

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{B} \\ \boldsymbol{O} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{A} & \boldsymbol{AB} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{B} \\ \boldsymbol{O} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{BA} & \boldsymbol{O} \\ \boldsymbol{A} & \boldsymbol{O} \end{bmatrix} = \boldsymbol{T} = \boldsymbol{PSP}^{-1} \tag{6}$$

  ，所以 $\boldsymbol{S} \sim \boldsymbol{T}$，有

$$\begin{aligned} &\det\left(\begin{bmatrix} -x\boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{A} & \boldsymbol{AB} - x\boldsymbol{I} \end{bmatrix}\right) = \det\left(\begin{bmatrix} \boldsymbol{BA} - x\boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{A} & -x\boldsymbol{I} \end{bmatrix}\right) \\ \Rightarrow &\det(-x\boldsymbol{I}) \det(\boldsymbol{AB} - x\boldsymbol{I}) = \det(\boldsymbol{BA} - x\boldsymbol{I}) \det(-x\boldsymbol{I}) \\ \Rightarrow &\det(\boldsymbol{AB} - x\boldsymbol{I}) = \det(\boldsymbol{BA} - x\boldsymbol{I}) \end{aligned} \tag{7}$$

  則 $\boldsymbol{AB}$ 與 $\boldsymbol{BA}$ 有相同特徵多項式。

- 若 $\boldsymbol{A}$ 為方陣，則
  - $\boldsymbol{A}^{-1}$，若 $\boldsymbol{A}$ 可逆

- $\boldsymbol{A}^m,\ \forall m \in \mathbb{Z}^+$
- $\alpha \boldsymbol{A}$
- $\boldsymbol{A} + \alpha \boldsymbol{I}$
- $f(\boldsymbol{A}),\ f(x) \in P$
- $\boldsymbol{A}^{\mathsf{H}}$，若 $\boldsymbol{A}$ 為正規矩陣，即 $\boldsymbol{A}\boldsymbol{A}^{\mathsf{H}} = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$。

特徵向量不改變。

- 若 $T, U \in \mathcal{L}(V, V)$ 皆可對角化，則

$$T, U \text{ 可同步對角化} \iff TU = UT \tag{8}$$

V, U 有相同特徵向量。

- 若 $T \in \mathcal{L}(V, V)$，且 $T^2 = T$，稱 T 為 V 上的冪等（idempotent）算子，則
  - $V = \mathrm{N}(T) \oplus \mathrm{Im}(T)$
  - $V(0) = \mathrm{N}(T), V(1) = \mathrm{Im}(T)$
- 若 $T \in \mathcal{L}(V, V)$，以下等價
  - $\mathrm{Im}(T) = \mathrm{Im}(T^2)$
  - $\mathrm{rank}(T) = \mathrm{rank}(T^2)$
  - $\mathrm{nullity}(T) = \mathrm{nullity}(T^2)$
  - $\mathrm{N}(T) = \mathrm{N}(T^2) \iff V = \ker(T) \oplus \mathrm{Im}(T)$
- 若 $T \in \mathcal{L}(V, V)$ 為冪零算子，且最小正整數 k 為 T 的指標，則 $\exists \boldsymbol{v} \in V \vee \boldsymbol{v} \in \mathrm{N}(T^k) - \mathrm{N}(T^{k-1})$ 且 $\boldsymbol{v} \neq \boldsymbol{0}$，$\{\boldsymbol{v}, T(\boldsymbol{v}), \cdots, T^{k-1}(\boldsymbol{v})\}$ 線性獨立。
- 若 $T \in \mathcal{L}(V, V)$，則
  - $\{\boldsymbol{0}\} \subseteq \mathrm{N}(T) \subseteq \mathrm{N}(T^2) \subseteq \cdots \subseteq V$
  - $W = \bigcup_{i=1}^{\infty} \mathrm{N}(T^i) = \mathrm{N}(T^k)$ 為最大冪零區。
  - $V \supseteq \mathrm{Im}(T) \supseteq \mathrm{Im}(T^2) \supseteq \cdots \supseteq \{\boldsymbol{0}\}$
  - $W = \bigcap_{i=1}^{\infty} \mathrm{Im}(T^i) = \mathrm{Im}(T^k)$ 為最大可逆區。
- 若 $T \in \mathcal{L}(V, V)$，則 $\exists\, k \in N$ 使得 $V = \mathrm{N}(T^k) \oplus \mathrm{Im}(T^k)$。

- 冪零矩陣特徵根全都是 0。
- 若 $T \in \mathcal{L}(V, V), \boldsymbol{v} \in V$，則
  - $\dim(C_{\boldsymbol{v}}(T)) = k$ **不能**保證 $\boldsymbol{v} \in \mathrm{N}(T^k) - \mathrm{N}(T^{k-1})$。
  - $\dim(C_{\boldsymbol{v}}(T)) = k$ 保證 $\beta = \{\boldsymbol{v}, T(\boldsymbol{v}), \cdots, T^{k-1}(\boldsymbol{v})\}$ 為 $C_{\boldsymbol{v}}(T)$ 的基底。
  - $\dim(C_{\boldsymbol{v}}(T)) = k$ **不能**保證 $T^k(\boldsymbol{v}) = 0$。
- 若 $T \in \mathcal{L}(V, V)$，W 為 T-不變子空間，則
  - T_W 的特徵多項式整除 T 的特徵多項式。
  - T_W 的極小多項式整除 T 的極小多項式。
  - 若 T 可對角化，則 T_W 也可對角化。
- 若 $T \in \mathcal{L}(V, V)$，且 $\lambda_1, \cdots, \lambda_r$ 為相異特徵根，則

$$T \text{ 可對角化} \iff m_T(x) = (x - \lambda_1) \cdots (x - \lambda_r) \tag{9}$$

- 若 $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times n}$ 為實方陣，當 $\boldsymbol{AB} = \boldsymbol{BA}$ 時，

$$e^A e^B = e^{A+B} \tag{10}$$

  可通過泰勒展開式證明。

-
  - $\mathrm{N}(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}) = \mathrm{N}(\boldsymbol{A})$
  - $\mathrm{rank}(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{A})$
  - $\mathrm{Lker}(\boldsymbol{A}\boldsymbol{A}^{\mathsf{H}}) = \mathrm{Lker}(\boldsymbol{A})$
  - 若 $\boldsymbol{A} \in \mathbb{R}^{m \times n}$，$\mathrm{rank}(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}) = \mathrm{rank}(\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}})$
  - $\mathrm{CS}(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}) = \mathrm{CS}(\boldsymbol{A}^{\mathsf{T}})$
  - $\boldsymbol{A}$ 行獨立 $\iff \boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$ 可逆
  - $\boldsymbol{A}$ 列獨立 $\iff \boldsymbol{A}\boldsymbol{A}^{\mathsf{H}}$ 可逆
- 若 $\boldsymbol{A} \in F^{m \times n}$，$W = \mathrm{CS}(\boldsymbol{A}), \boldsymbol{b} \in F^{m \times 1}$，則
  - 若 $\boldsymbol{A}$ 行獨立，且 $\boldsymbol{A} = \boldsymbol{QR}$ 為 $\boldsymbol{A}$ 的 QR 分解，則 $\boldsymbol{x} \in F^{n \times 1}$ 使得 $||\boldsymbol{Ax} - \boldsymbol{b}||$ 最小 $\iff \boldsymbol{Rx} = \boldsymbol{Q}^{\mathsf{H}}\boldsymbol{b}$。

– 若 $\boldsymbol{Q}$ 的行向量為**單範正交集**，$W = \mathrm{CS}(\boldsymbol{Q})$，則

$$\mathrm{proj}_W \boldsymbol{b} = \boldsymbol{Q}\boldsymbol{Q}^{\mathsf{H}}\boldsymbol{b} \tag{11}$$

- 若 $\boldsymbol{A} \in F^{n\times n}$ 為方陣，且 $\boldsymbol{A}$ **行獨立**，則
  - $P^2 = P$ 為冪等方陣且 $P^{\mathsf{H}} = P \iff P$ 為正交投影矩陣
  - $\mathrm{CS}(P) = \mathrm{CS}(\boldsymbol{A})$
  - $\mathrm{rank}(P) = \mathrm{rank}(\boldsymbol{A}) = n$
- 若 $W \subseteq V$，P 為 V 在 W 上的投影算子，則 $\mathrm{N}(P) = W^{\perp}$。
- 若 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 有解，則
  - **唯一** $\exists \boldsymbol{s} \in \mathrm{CS}(\boldsymbol{A}^{\mathsf{H}})$ 為 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 之極小解，即 $||\boldsymbol{x}||_2$ 為所有解中最小。
  - 若 $\boldsymbol{u}$ 滿足 $(\boldsymbol{A}\boldsymbol{A}^{\mathsf{H}})\boldsymbol{u} = \boldsymbol{b}$，則 $\boldsymbol{s} = \boldsymbol{A}^{\mathsf{H}}\boldsymbol{u}$。
- 證明 Cauchy-Schwarz inequality：

$$|<\boldsymbol{u},\boldsymbol{v}>| \le ||\boldsymbol{u}|| \times ||\boldsymbol{v}|| \tag{12}$$

Proof. 用數學歸納法證明：
若 $\boldsymbol{v} = \boldsymbol{0}$，成立。
若 $\boldsymbol{v} \neq \boldsymbol{0}$，取

$$\alpha = \frac{<\boldsymbol{u},\boldsymbol{v}>}{<\boldsymbol{v},\boldsymbol{v}>} \tag{13}$$

則

$$\begin{aligned}
0 &\le ||\boldsymbol{u} - \alpha\boldsymbol{v}||^2 \\
&= <\boldsymbol{u},\boldsymbol{u}> - \overline{\alpha} <\boldsymbol{u},\boldsymbol{v}> - \alpha <\boldsymbol{v},\boldsymbol{u}> + \alpha\overline{\alpha} <\boldsymbol{v},\boldsymbol{v}> \\
&= <\boldsymbol{u},\boldsymbol{u}> - \frac{\overline{<\boldsymbol{u},\boldsymbol{v}>}}{<\boldsymbol{v},\boldsymbol{v}>} <\boldsymbol{u},\boldsymbol{v}> - \frac{<\boldsymbol{u},\boldsymbol{v}>}{<\boldsymbol{v},\boldsymbol{v}>} <\boldsymbol{v},\boldsymbol{u}> \\
&\quad + \frac{<\boldsymbol{u},\boldsymbol{v}>}{<\boldsymbol{v},\boldsymbol{v}>} \frac{\overline{<\boldsymbol{u},\boldsymbol{v}>}}{<\boldsymbol{v},\boldsymbol{v}>} <\boldsymbol{v},\boldsymbol{v}> \\
&= ||\boldsymbol{u}||^2 - \frac{|<\boldsymbol{u},\boldsymbol{v}>|^2}{||\boldsymbol{v}||^2} \\
\Rightarrow & \frac{|<\boldsymbol{u},\boldsymbol{v}>|^2}{||\boldsymbol{v}||^2} \le ||\boldsymbol{u}||^2 \\
\Rightarrow & |<\boldsymbol{u},\boldsymbol{v}>|^2 \le ||\boldsymbol{u}||^2 \times ||\boldsymbol{v}||^2
\end{aligned} \tag{14}$$

- 若 $T \in \mathcal{L}(V, V)$，則

定義	λ	a_{ii}	det
self-adjoint $T^* = T$ Hermitian (over $\mathbb{C}$) $\boldsymbol{A}^{\mathsf{H}} = \boldsymbol{A}$	$\in \mathbb{R}$	$\in \mathbb{R}$	$\in \mathbb{R}$
symmetric (over $\mathbb{R}$) $\boldsymbol{A}^{\intercal} = \boldsymbol{A}$			
skew self-adjoint $T^* = -T$ skew Hermitian (over $\mathbb{C}$) $\boldsymbol{A}^{\mathsf{H}} = -\boldsymbol{A}$	0 或純虛數	0 或純虛數	$\begin{cases} \in \mathbb{R}, & \text{if } n \in 2k \\ 0\text{或純虛數}, & \text{if } n \in 2k+1 \end{cases}$
skew symmetric (over $\mathbb{R}$) $\boldsymbol{A}^{\intercal} = -\boldsymbol{A}$	0	0	$\begin{cases} \in \mathbb{R}, & \text{if } n \in 2k \\ 0, & \text{if } n \in 2k+1 \end{cases}$
positive definite $< T(\boldsymbol{x}), \boldsymbol{x} >> 0,\ \forall \boldsymbol{x} \neq \boldsymbol{0}$ $< \boldsymbol{A}\boldsymbol{x}, \boldsymbol{x} >= \boldsymbol{x}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{x} > 0,\ \forall \boldsymbol{x} \neq \boldsymbol{0}$	> 0	> 0	> 0
positive semi-definite $< T(\boldsymbol{x}), \boldsymbol{x} >\geq 0,\ \forall \boldsymbol{x}$ $< \boldsymbol{A}\boldsymbol{x}, \boldsymbol{x} >= \boldsymbol{x}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{x} \geq 0,\ \forall \boldsymbol{x}$	≥ 0	≥ 0	≥ 0
unitary (over $\mathbb{C}$) $T^*T = \boldsymbol{I}$ $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A} = \boldsymbol{I}$	$\lvert\lambda\rvert = 1$	$\times$	$\lvert\det(\boldsymbol{A})\rvert = 1$
orthogonal (over $\mathbb{R}$) $T^*T = \boldsymbol{I}$ $\boldsymbol{A}^{\intercal}\boldsymbol{A} = \boldsymbol{I}$	± 1	$\times$	± 1

- 若 $\boldsymbol{A} \in \mathbb{C}^{n\times n} (\vee\ \mathbb{R}^{n\times n})$，則以下等價
  - $\boldsymbol{A}$ 為么正（正交）。
  - $< \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{y} >=< \boldsymbol{x}, \boldsymbol{y} >$，保內積。
  - $||\boldsymbol{A}\boldsymbol{x}|| = ||\boldsymbol{x}||$，保長度。
- 若 $\boldsymbol{A} \in F^{n\times n}$ 為么正或是正交方陣，則
  - CS($\boldsymbol{A}$) 和 RS($\boldsymbol{A}$) 皆為單範正交集。

  - 若 $\boldsymbol{A}$ **不為方陣**，則 $\mathrm{RS}(\boldsymbol{A})$ **未必**為單範正交集。

- 若 $\boldsymbol{A}, \boldsymbol{B} \in F^{n\times n}$ 為方陣，且 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 么正相等，則 $\mathrm{tr}(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}) = \mathrm{tr}(\boldsymbol{B}^{\mathsf{H}}\boldsymbol{B})$。

-
  - 若 $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{C}^{n\times n}$ 為複數方陣，且 $\boldsymbol{A}$ 與 $\boldsymbol{B}$ 么正相似，則以下 $\boldsymbol{A}$ 與 $\boldsymbol{B}$ 的性質等價
    - 正規
    - Hermitian
    - 斜 Hermitian
    - 正定
    - 半正定
    - 么正
  - 若 $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$ 為實方陣，且 $\boldsymbol{A}$ 與 $\boldsymbol{B}$ 正交相似，則以下 $\boldsymbol{A}$ 與 $\boldsymbol{B}$ 的性質等價
    - 對稱
    - 斜對稱
    - 正交
  - 若 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 為複數方陣，則 $\boldsymbol{A}$ 為正規且上三角方陣 $\iff$ $\boldsymbol{A}$ 為對角方陣。
  - 若 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 為複數方陣，則 $\boldsymbol{A}$ 為正規方陣 $\iff$ $\boldsymbol{A}$ 可么正對角化。
  - 若 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 為實方陣，則 $\boldsymbol{A}$ 為對稱方陣 $\iff$ $\boldsymbol{A}$ 可正交對角化。

- Cholesky 分解必須**對稱**且**正定**。

- 若 $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{H}}$ 為 $\boldsymbol{A}$ 的奇異值分解，則
  - $$\sum_{i=1}^{r} \sigma^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2 \tag{15}$$
  - $\mathrm{CS}(\boldsymbol{V})$ 為 $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$ 的特徵向量且為單範正交集。
  - $\mathrm{CS}(\boldsymbol{U})$ 為 $\boldsymbol{A}\boldsymbol{A}^{\mathsf{H}}$ 的特徵向量且為單範正交集。
  - 若 $\mathrm{rank}(\boldsymbol{A}) = r$ 為非零奇異值個數，則
    - $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_r$ 為 $\mathrm{CS}(\boldsymbol{A}^{\mathsf{H}})$ 單範正交基底。
    - $\boldsymbol{v}_{r+1}, \cdots, \boldsymbol{v}_n$ 為 $\mathrm{N}(\boldsymbol{A})$ 單範正交基底。
    - $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r$ 為 $\mathrm{CS}(\boldsymbol{A})$ 單範正交基底。

    * $\boldsymbol{u}_{r+1},\ \cdots, \boldsymbol{u}_m$ 為 $\mathrm{N}(\boldsymbol{A}^{\mathsf{H}})$ 單範正交基底。

- 若 $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\intercal} \in \mathbb{R}^{m\times n}$，則 $\boldsymbol{X} = \boldsymbol{A}^{+}$ 有
  - $\boldsymbol{A}\boldsymbol{X}\boldsymbol{A} = \boldsymbol{A}$
  - $\boldsymbol{X}\boldsymbol{A}\boldsymbol{X} = \boldsymbol{X}$
  - $(\boldsymbol{A}\boldsymbol{X})^{\intercal} = \boldsymbol{A}\boldsymbol{X}$
  - $(\boldsymbol{X}\boldsymbol{A})^{\intercal} = \boldsymbol{X}\boldsymbol{A}$

  $\boldsymbol{X}^{+}$ 為**唯一**滿足這四個條件的矩陣。
- $2^{mn} \pmod{2^m - 1} = 1$。
- 若 p 為質數，$a \in \mathbb{Z}$，則 $a^{-1} \equiv a \pmod{p}$ 即 $a^2 \equiv 1 \pmod{p} \iff a \equiv \pm 1 \pmod{p}$。
- - Wilson's theorem：若 p 為質數，則

$$(p-1)! \equiv -1 \pmod{p} \tag{16}$$

  - Fermat's little theorem：若 p 為質數，$m \in Z$，且 $\gcd(m,p) = 1$，則

$$m^{p-1} \equiv 1 \pmod{p} \tag{17}$$

- - 若 $m \in \mathbb{Z}, n \in \mathbb{Z}^{+}$，且 $\gcd(m,n) = 1$，則 $m^{\phi(n)} \equiv 1 \pmod{n}$。
  - 若 p 為質數，$m \in Z$，且 $\gcd(m,p) = 1$，則 $m^{-1} \equiv m^{p-2} \pmod{p}$
- If $2^n - 1$ is prime, then n is prime.
- 證明 $\mathbb{Z}^{+}$ 中質數個數為 ∞。

  Proof. 若質數個數為有限個，令

$$P_1,\ P_2,\ \cdots,\ P_k \tag{18}$$

  為所有質數。取

$$E = P_1 P_2 \cdots P_k + 1 \tag{19}$$

  所以 E 為 composite，則

$$\exists\ P_j \quad \text{s.t. } P_j | E \tag{20}$$

又

$$\begin{aligned} & P_j | P_1 P_2 \cdots P_k \\ \Rightarrow & P_j | (E - P_1 P_2 \cdots P_k) \\ \Rightarrow & P_j | 1 \end{aligned} \tag{21}$$

但質數 P_j 不可能整除 1，矛盾，因此 $P_j = 1$，E 為質數。得證，$\mathbb{Z}^+$ 中質數個數為 ∞。

- 證明 $(0,1)$ 為不可數集。

  Proof. $f : \mathbb{Z}^+ \to (0,1)$ is bijective，令 $f(i) = r_i,\ \forall i = 1, 2, 3, \cdots$ 其中

$$\begin{cases} r_1 = 0.r_{11} r_{12} \cdots \\ r_2 = 0.r_{21} r_{22} \cdots \\ \quad \vdots \\ r_i = 0.r_{i1} r_{i2} \cdots \end{cases} \tag{22}$$

  取

$$s = 0.s_1 s_2 \cdots,\ s_i = \begin{cases} 4 & , r_{ii} \neq 4 \\ 5 & , r_{ii} = 4 \end{cases} \tag{23}$$

  $s_i \in (0,1)$ 但 $\nexists\, i \in \mathbb{Z}^+$ s.t. $f(i) = s$，因此 $(0,1)$ 為不可數集。

-

$$A = \{1,\ 2,\ \cdots,\ 2n\} \tag{24}$$

  在 A 取 $N+1$ 個數，

$$\exists\ a, b \quad \text{s.t. } a | b \vee b | a \tag{25}$$

  Proof.

$$\forall\ x \in A,\ x = 2^k \times y,\ k \in \mathbb{Z},\ y = 2l + 1,\ l \in \mathbb{Z} \tag{26}$$

  又 A 中只有 n 個奇數，則取 $n+1$ 個數時，

$$\begin{aligned} & \exists\ a, b \quad \text{s.t. } a = 2^{k_1} \times y,\ b = 2^{k_2} \times y \\ & a | b \vee b | a \end{aligned} \tag{27}$$

-

– 若 A 為一集合，且 $|A| = m$，A 上等價關係的個數，即相異分割數。

$$\sum_{i=1}^{m} \mathrm{S}(m, i) \tag{28}$$

– m 相異物放入 n 相同箱**可**空箱的方法。

$$\sum_{i=1}^{n} \mathrm{S}(m, i) \tag{29}$$

- Ordered sum of positive integers, where each summand is ≥ 2:

$$\begin{cases} a_n = a_{n-1} + a_{n-2} & , n \geq 2 \\ a_1 = 0, a_2 = 1 \end{cases} \tag{30}$$

- 若 G 與 $\overline{G}$ 同構，且 $|V| = n$，則 $n = 4k \vee n = 4k + 1$。
-
  – 一簡單無向圖，若所有點的度數 $\geq k$，則圖上必含一個長度至少為 $k+1$ 的環路（cycle）。
  – 若 A 為一鄰接矩陣，則
    * $\frac{1}{6}\operatorname{tr}(A^3)$ 為圖上三角形個數。
    *
$$\sum_{i=1}^{n}\sum_{j=1}^{n} A^2[i, j] = \sum_{i=1}^{n} \deg(v_i)^2 \tag{31}$$

-
  – Maximum length of a **trail** of K_n is $\binom{2n}{2} - (n-1)$.
  – Maximum length of a **circuit** of K_n is $\binom{2n}{2} - n$.

-
  – 圖中有尤拉迴路 $\iff$ 為連接圖且所有點的度數為偶數。
  – K_n 有尤拉迴路 $\iff n$ 為奇數。
  – $K_{m,n}$ 有尤拉迴路 $\iff m, n$ 為偶數。
  – 圖中有尤拉路線 $\iff$ 為連通圖且圖中恰含 0 個或 2 個點度數為奇數。

- 圖中有尤拉迴路 $\iff$ 為強連通圖且所有點的出度數與入度數相同。
- 若圖中有尤拉迴路，則有尤拉路線。

•

- K_n^* 必定有有向漢米爾頓路徑。
- 若 $G = (V, E),\ |V| = n \geq 3$ 為一無迴圈無向圖，
  - 若
$$\deg(x) + \deg(y) \geq n - 1,\ \forall x, y \in V, x \neq y \vee \deg(v) \geq \frac{n-1}{2},\ \forall v \in V \tag{32}$$
  ，則 G 有漢米爾頓**路徑**。
  - 若
$$\deg(x) + \deg(y) \geq n,\ \forall x, y \in V, x, y \text{ 不相鄰} \vee \deg(v) \geq \frac{n}{2},\ \forall v \in V \tag{33}$$
  ，則 G 有漢米爾頓**環路**。
- $K_n, n \geq 3$ 必有漢米爾頓**環路**。
- 若一圖有漢米爾頓環路，則該圖中任兩點至少有兩條路徑相連。
- 一連通雙分圖，若圖中有漢米爾頓**環路**，則兩邊的頂點數相同。
- 一連通雙分圖，若圖中有漢米爾頓**路徑**，則兩邊的頂點數相差 ≤ 1。
- K_n 有 $\frac{(n-1)!}{2}$ 個相異漢米爾頓環路。
- K_n，n 為奇數，有 $\leq \frac{n-1}{2}$ 個**不共邊**的漢米爾頓環路。
- $K_{n,n}$ 有 $\frac{1}{2}n!(n-1)!$ 個相異漢米爾頓環路。
- 若 $G = (V, E),\ |V| = n$，則
$$|E| \geq \binom{n-1}{2} + 2 \tag{34}$$
  時，G 有漢米爾頓**環路**。

•

- Euler formula：若 $G = (V, E), |V| = v, |E| = e, r$為區域個數, M為分量圖數，且 G 為平面圖，則 $v - e + r = 1 + M$。

- 若 $G=(V,E), |V|=v, |E|=e\geq 2, r$為區域個數, M為分量圖數，且 G 為無迴圈簡單**連通**平面圖，則
  - $$\frac{3}{2}r \leq e \leq 3v-6 \tag{35}$$
  - 若 G **不含任何三角形**，則
  $$e \leq 2v-4 \tag{36}$$
  - 若每個環路 $\geq k \geq 3$ 邊組成，則
  $$e \leq \frac{k}{k-2}(v-2M) \tag{37}$$
- 一無迴圈簡單平面圖必含一個度數 ≤ 5 的頂點。

-
  - 若 $P(G,\lambda)$ 為著色多項式，則
    - $P(G,\lambda)$ 常數項為 0。
    - $P(G,\lambda)$ 係數和為 0。
    - $P(G,\lambda)$ 最高次項係數為 1。
- Edge-coloring:
$$\chi'(K_n) = \begin{cases} n-1 & , n=2k \\ n & , n=2k+1 \end{cases}$$
$$\chi'(C_n) = \begin{cases} 2 & , n=2k \\ 3 & , n=2k+1 \end{cases} \tag{38}$$
$$\chi'(K_{n,n}) = n$$
- 若 $G=(V,E)$ is connected，則
$$|E| \geq |V|-1 \tag{39}$$

Proof. 用數學歸納法證明：
當 $|V|=1$ 時，成立。
設 $|V|<n$ 時成立。考慮 $|V|=n$ 時，$\forall\, v,\ \deg(v)=m$，則 $G-v$ 形成 k 個 components，有
$$G_1=(V_1,E_1),\ G_2=(V_2,E_2),\ \cdots,\ G_k=(V_k,E_k) \tag{40}$$

又 $G_i,\ 1 \leq k \leq m$ is connected，且 $|V_i| < n$。根據數學歸納法，

$$|E_i| \geq |V_i| - 1,\ \forall\ i = 1,\ \cdots, k \tag{41}$$

則

$$\begin{aligned} |E| &= |E_1| + \cdots + |E_k| + m \\ &\geq (|V_1| - 1) + \cdots + (|V_k| - 1) + m \\ &= (|V_1| + \cdots + |V_k|) + (m - k) \\ &= |V| - 1 + (m - k) \\ &\geq |V| - 1 \end{aligned} \tag{42}$$

-
  - 若 $T = (V, E), |V| = n$ 為 m-元樹，其中 i, l 分別表示內部節點與樹葉個數，則
    - $$n \leq mi + 1 \tag{43}$$
    - $$l \leq (m - 1)i + 1 \tag{44}$$

      當 T 為**滿** m-元樹時，等號成立。
  - 一滿 m-元樹，i 為內部節點個數，I, E 分別表示內部及外部路徑長，則

$$E = (m - 1)I + mi \tag{45}$$

-
  - K_n 相異生成樹個數為 n^{n-2}。
  - $K_{m,n}$ 相異生成樹個數為 $m^{n-1}n^{m-1}$。
  - 若 $G = (V, E)$ 為無向圖，且 $e = \{a, b\} \in E$，$N(G)$ 為 G 的相異生成樹個數，則

$$N(G) = N(G - e) + N(G \cdot e) \tag{46}$$

  - 一無向連通圖，其任意切集與環路必含**偶數**個共同邊。

- $\exists x\ (P(x) \wedge Q(x)) \neq \exists x\ P(x) \wedge \exists\ xP(x)$

- ($|A| = |B|$) A: The set of all programs that terminate. B: The set of all programs that do NOT terminate.
-
  - $\mathrm{CS}(\boldsymbol{A}^+) = \mathrm{CS}(\boldsymbol{A}^\intercal) = \mathrm{RS}(\boldsymbol{A})$
  - $\boldsymbol{x}_0 = \boldsymbol{A}^+\boldsymbol{b}$ 為 $||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2$ 的最小平方解。
-
- (**FALSE**) For any non-zero real **symmetric** matrix, its SVD can be the same as its eigenvalue decomposition.
- For any non-zero real matrix $\boldsymbol{A}$, $\boldsymbol{A}^\intercal\boldsymbol{A}$'s SVD can be the same as its eigenvalue decomposition. 因為 $\boldsymbol{A}^\intercal\boldsymbol{A}$ 正半定，$\boldsymbol{A}$ 可么正對角化，$\exists\ \boldsymbol{P}$ 為么正矩陣，使得

$$\begin{aligned} &\boldsymbol{P}^\intercal(\boldsymbol{A}^\intercal\boldsymbol{A})\boldsymbol{P} = \boldsymbol{D} \\ \Rightarrow &\boldsymbol{A}^\intercal\boldsymbol{A} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^\intercal \end{aligned} \tag{47}$$

  為 $\boldsymbol{A}^\intercal\boldsymbol{A}$ 的 SVD。
- 可對角化不保證 non-singular。
- If $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{A}+\boldsymbol{B}$ are non-singular square matrices, and $\boldsymbol{A}^{-1}+\boldsymbol{B}^{-1}$ is also non-singular. Since $\boldsymbol{A}(\boldsymbol{A}+\boldsymbol{B})^{-1}\boldsymbol{B}$ is invertible,

$$(\boldsymbol{A}(\boldsymbol{A}+\boldsymbol{B})^{-1}\boldsymbol{B})^{-1} = \boldsymbol{B}^{-1}(\boldsymbol{A}+\boldsymbol{B})\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}+\boldsymbol{B}^{-1} \tag{48}$$

- The transition matrix from one basis to another must be **non-singular**, but a linear transformation matrix can be singular.
- (**FALSE**) Let S be a subset of an inner product space, then $S = (S^\perp)^\perp$.
- (**FALSE**) Let S_1, S_2 be subsets of an inner product space, and $S_1^\perp = S_2^\perp$, then $S_1 = S_2$.
- (**FALSE**) If V is orthogonal to W, then $V^\perp$ is orthogonal to $W^\perp$.
- SVD 中 singular value 遞減排序。
- $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ ($\boldsymbol{b} \neq \boldsymbol{0}$) is consistent, then solution set is **NOT** a subspace, since $\boldsymbol{0}$ is NOT included.

- If W is a subset of $\mathbb{R}^n$, but $W \cup W^{\perp} \neq \mathbb{R}^n$, since $W^{\perp}$ is NOT a subset.
- If $V \in \mathbb{R}^{m \times n}$, $< \boldsymbol{A}, \boldsymbol{B} >= \text{tr}(\boldsymbol{B}^{\intercal}\boldsymbol{A})$ does **NOT** define an inner space in V, since if $m \neq n$, $\boldsymbol{B}^{\intercal}\boldsymbol{A}$ may **NOT** exist.
- (101NTU-10) If

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{49}$$

, then

$$\lambda_{\boldsymbol{A}} = 2 \times \cos(\frac{k\pi}{n+1}),\ k = 1,\ 2,\ \cdots,\ n \tag{50}$$

1-71 N toys are to be distributed randomly among N children. There is an interesting way for the children to choose the toys so that no two of them will choose the same toy. A graph such as the show in Figure A is drawn where there are N vertical lines and an arbitrary number of random horizontal segments between adjacent vertical lines with the stipulation that no two horizontal segments meet at the same point. The N toys are assigned to the bottoms of the vertical lines, and each child chooses as a starting point the top of a vertical line. From this starting point, the child will trace a path downward. However, whenever the child runs into a horizontal segment, he or she mush turn horizontal, and then turn downward again when the adjacent vertical line is reached. For example, Figure B shows the path that John follows. Show that (a) No matter how many horizontal segments are drawn, in whatever possible way, no two children will reach the same toy, and (b) all the horizontal and vertical segments will be passed. (92 師大資工)

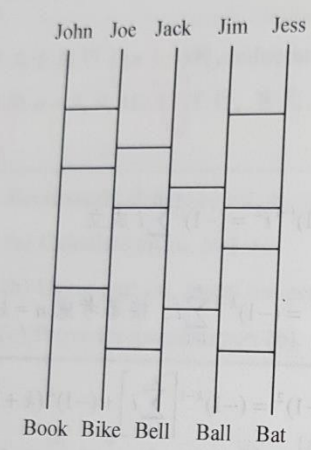


Figure A

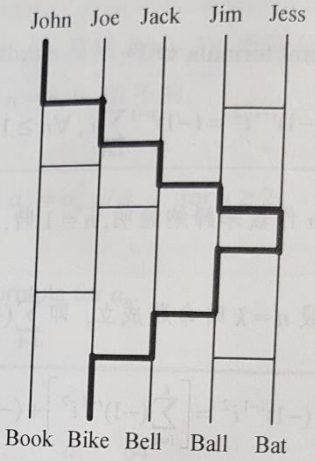


Figure B

解

假設 n 表示 horizontal line 的個數, a_i 表示第 i 個 children 所選到的 toy

欲證 $a_1, a_2, \ldots, a_N$ 皆相異且所有 horizontal lines 及 vertical lines 皆被使用到

對 n 作數學歸納證明, $n=0$ 時, $a_i = i, \forall i = 1, 2, \ldots, N$, 另外, N 條 vertical lines 皆被使用到, 假設 $n=k$ 時命題成立, 接著考慮 $n=k+1$, 假設 G 為它的圖

令 h 表示最下面的某一條 horizontal line 且 h 連接第 i 及第 j 條 vertical lines

假設 G 中去掉 h 形成 G', 則 G' 含有 k 條 horizontal lines

根據數學歸納假設, 在 G' 中 $a_1, a_2, \ldots, a_N$ 皆相異且 G' 中所有 horizontal lines 及 vertical lines 皆被使用到, 因為 G 為 G' 中加入 h, 所以 G 中相當於 a_i 與 a_j 交換, 因此 G 中 $a_1, a_2, \ldots, a_N$ 皆相異, 另外, 多加的一條 horizontal line h 也被使用到, 所以 $n=k+1$ 亦成立, 得證

1-85 Find all positive integer(s) p such that $1 + 2^p$ is a square number. You should also prove that there is no other integer (except what you just found) such that $1 + 2^p$ is a square number. (86 暨大資管)

當 $p = 1, 2$ 時顯然地 $1 + 2^p$ 不為完全平方數

若 $p \in \mathbf{Z}^+, p \geq 3$ 使得 $1 + 2^p$ 為一完全平方數

$\Rightarrow \exists x \in \mathbf{Z}^+$ 使得 $1 + 2^p = x^2$

$\Rightarrow 2^p = x^2 - 1 = (x + 1)(x - 1)$

因為 2^p 為偶數且 $x + 1$ 與 $x - 1$ 必同為奇數或同為偶數

$\Rightarrow x + 1$ 與 $x - 1$ 皆為偶數，即 x 為奇數

令 $x = 2k + 1$, for some $k \in \mathbf{Z}^+$

$\Rightarrow 2^p = (2k + 2)(2k) = 4k(k + 1)$

$\Rightarrow 2^{p-2} = k(k + 1)$

當 $k = 1$ 時 $p = 3$ 為一個解

當 $k \geq 2$ 時，令 $k = p_1^{e_1} p_2^{e_2} \ldots p_r^{e_r}$，$k + 1 = q_1^{f_1} q_2^{f_2} \ldots q_s^{f_s}$ 分別為 k 及 $k + 1$ 的質因數分解，

其中 $e_i, f_j \geq 1, \forall 1 \leq i \leq r, 1 \leq j \leq s, r, s \geq 1$

因為 k 與 $k + 1$ 必互質，所以 $p_i \neq q_j, \forall 1 \leq i \leq r, 1 \leq j \leq s$

$2^{p-2} = k(k + 1) = p_1^{e_1} p_2^{e_2} \ldots p_r^{e_r} q_1^{f_1} q_2^{f_2} \ldots q_s^{f_s}$ 為 2^{p-2} 的質因數分解

$\Rightarrow r = s = 1, e_1 + f_1 = p - 2, p_1 = q_1 = 2$，產生矛盾

根據以上討論，只有在 $p = 3$ 時 $1 + 2^p$ 才為完全平方數

2-97 In a twelve-day period Ms. Rosatone typed 81 letters to different clients. Show typed nine of these letters on the first day, six on the second day, and four on the eleventh day, and she finished the last two on the twelfth day. Show that for a period of three consecutive days Ms. Rosatone typed at least 22 letters. (96 暨大資工)

解：

第 1, 2, 11, 12 天的 letter 數分別為 9, 6, 4, 2，所以第 3 到 10 天的 letter 數為 $81 - (9 + 6 + 4 + 2) = 60$，將這 12 天分成 10 段：{1, 2, 3}, {2, 3, 4}, {3, 4, 5}, {4, 5, 6}, {5, 6, 7}, {6, 7, 8}, {7, 8, 9}, {8, 9, 10}, {9, 10, 11}, {10, 11, 12}，每段包含連續 3 天，欲證明這 10 段中至少有一段的 letter 數至少為 22，利用矛盾證法，若每段的 letter 數至多 21，則 10 段總合含重複的 letter 數至多 $21 \times 10 = 210$，但這 10 段重複算第 1 及第 12 天各 1 次，第 2 及第 11 天各 2 次，第 3 到第 10 天各 3 次，它的 letter 數為 $(9 + 2) + 2(6 + 4) + 3 \times 60 = 211$，產生矛盾，因此存在一段連續 3 天的 letter 數至少 22

1-125 If $A, B \in \boldsymbol{R}^{n\times n}$ and $tr(ABC) = tr(CBA)$ for all $C \in \boldsymbol{R}^{n\times n}$, prove that $AB = BA$. (96 花教大數學)

解。

因為 $tr(ABC) = tr((AB)C) = tr(C(AB)) = tr(CAB)$

所以 $tr(CAB) = tr(CBA)$

$\Rightarrow 0 = tr(CAB) - tr(CBA) = tr(CAB - CBA) = tr(C(AB - BA)), \forall C \in \boldsymbol{R}^{n\times n}$

取 $C = (AB - BA)^T$, 則 $tr((AB - BA)^T(AB - BA)) = 0$

因此 $AB - BA = O$, 所以 $AB = BA$

2-117 Andrea has 46 rectangular pieces of paper. If l, w (measured in centimeters) denote the length and width, respectively, of each rectangular piece, then for this situation we find that each of l, w is a positive integer, where $1 \le w \le l \le 90$. From among these 46 rectangles, prove that Andrea can select two, say R_1 and R_2, so that R_2 completely covers R_1 when R_2 is placed on top of R_1. (88 中央資工)

解。

令 R_i 表第 i 張 rectangular paper, 其中 R_i 具有 length l_i, width w_i, $i = 1, 2, \ldots, 46$

不失一般性令 $l_1 \le l_2 \le \cdots \le l_{46}$

(1) 若$\exists i \ne j$ 使得 $l_i = l_j$

當 $w_i \le w_j$ 時, R_j 完全 cover R_i, 得證

當 $w_i > w_j$ 時, R_i 完全 cover R_j, 得證

(2) 若$\exists i \ne j$ 使得 $w_i = w_j$

當 $l_i \le l_j$ 時, R_j 完全 cover R_i, 得證

當 $l_i > l_j$ 時, R_i 完全 cover R_j, 得證

(3)若 $l_i \ne l_j$ 且 $w_i \ne w_j$, $\forall i, j = 1, 2, \ldots, 46$ 且 $i \ne j$

根據鴿籠原理, $l_1 \le 45$, 則 $w_1 \le l_1 \le 45$

因為 $w_2, w_3, \ldots, w_{46} \ge 1$, 根據鴿籠原理, 存在 $w_k \ge 45$, for some $k \in \{2, 3, \ldots, 46\}$

$\Rightarrow l_k \ge l_1$ 且 $w_k \ge w_1$

$\Rightarrow R_k$ 完全 cover R_1, 得證

4-44 Determine the number of ways to distribute $2t+1$ indistinguishable coins to t boys so that any two boys together will have more coins than the other one. (92 清大資

解

假設 x_1, x_2, x_3 分別表示 3 個 boys 分到的 coins 個數，則 $x_1+x_2+x_3=2t+1$

因為 $x_1+x_2>x_3$，所以 $x_3<x_1+x_2=2t+1-x_3$

$\Rightarrow 2x_3<2t+1$

$\Rightarrow x_3<t+\frac{1}{2}$

$\Rightarrow x_3\le t$，同理可證 $x_1, x_2\le t$

另外，$x_3=2t+1-(x_1+x_2)\ge 2t+1-(t+t)=1$，同理可證 $x_1, x_2\ge 1$

所以欲使任二個 boys 的 coins 個數和大於第三個 boy 需滿足 $1\le x_i\le t, \forall i=1,2$

即求 $x_1+x_2+x_3=2t+1,\ 1\le x_i\le t, \forall i=1,2,3$ 的整數解個數

對應的生成函數為 $A(x)=(x+x^2+\cdots+x^t)^3$，欲求 x^{2t+1} 的係數

$$A(x)=x^3(1+x+\cdots+x^{t-1})^3=x^3\left(\frac{1-x^t}{1-x}\right)^3=x^3(1-3x^t+3x^{2t}-x^{3t})\sum_{r=0}^{\infty}\binom{3+r-1}{r}x^r$$

x^{2t+1} 的係數為 $\binom{3+(2t-2)-1}{2t-2}-3\binom{3+(t-2)-1}{t-2}=\binom{2t}{2}-3\binom{t}{2}=\frac{1}{2}t(t+1)$

範例 5

Let $K=\{x \mid x \text{ is a real number and } 0\le x\le 1\}$. Show that $K\sim K^2$. (89 政大資科)

解

定義 $f: K^2\to K$ 為 $f(0.x_1x_2\cdots, 0.y_1y_2\cdots)=0.x_1y_1x_2y_2\cdots$，欲證 f 為 one-to-one 且 onto

假設 $f(0.x_1x_2\cdots, 0.y_1y_2\cdots)=f(0.z_1z_2\cdots, 0.w_1w_2\cdots)$

$\Rightarrow 0.x_1y_1x_2y_2\cdots=0.z_1w_1z_2w_2\cdots$

$\Rightarrow x_i=z_i$ 且 $y_i=w_i, \forall i=1,2,\ldots$

$\Rightarrow 0.x_1x_2\cdots=0.z_1z_2\cdots$ 且 $0.y_1y_2\cdots=0.w_1w_2\cdots$

$\Rightarrow (0.x_1x_2\cdots, 0.y_1y_2\cdots)=(0.z_1z_2\cdots, 0.w_1w_2\cdots)$，所以 f 為 one-to-one

$\forall y=0.y_1y_2\cdots\in K$，取 $(0.y_1y_3y_5\cdots, 0.y_2y_4y_6\cdots)\in K^2$ 使得

$f(0.y_1y_3y_5\cdots, 0.y_2y_4y_6\cdots)=0.y_1y_2y_3y_4\cdots=y$，所以 f 為 onto，因此 $K^2\sim K$

範例 6

If A is any set, prove that $|A| < |\mathcal{P}(A)|$ (91 中正資工)(93 交大資訊)

解。

若 $A = \varnothing$, 則 $|A| = 0, |\mathcal{P}(A)| = 2^0 = 1$ 滿足 $|A| < |\mathcal{P}(A)|$

若 $A \neq \varnothing$, 定義 $f: A \to \mathcal{P}(A)$ 為 $f(a) = \{a\}, \forall a \in A$

則 f 顯然為一對一函數, 因此 $|A| = |f(A)| \leq |\mathcal{P}(A)|$

欲證明 $|A| \neq |\mathcal{P}(A)|$, 只需證明不存在 $g: A \to \mathcal{P}(A)$ 為映成函數即可

若存在 $g: A \to \mathcal{P}(A)$ 為映成函數, 令 $B = \{a \in A \mid a \notin g(a)\}$, 則 $B \in \mathcal{P}(A)$

因此存在 $b \in A$ 使得 $g(b) = B$, 這是因為 g 為映成函數

我們知道 $b \in g(b)$ 與 $b \notin g(b)$ 恰有一者成立

若 $b \in g(b) = B$, 根據 B 的定義知 $b \notin g(b)$, 產生矛盾

若 $b \notin g(b) = B$, 根據 B 的定義知 $b \in g(b)$, 產生矛盾

所以不論 $b \in g(b)$ 或 $b \notin g(b)$ 皆產生矛盾, 因此此種映成函數必不存在

14. (6%) Given positive integers $C > 1$ and n, the n-tuple optimization problem is to determine whether there exist positive integers $c_1, c_2, \ldots, c_n$ such that $\prod_{i=1}^{n} c_i = C$ and $\sum_{i=1}^{n} c_i$ is minimized. The prime number problem is to determine whether a given positive integer is a prime number or not. Please prove that if the n-tuple optimization problem can be solved in polynomial time, then the prime number problem can be solved in polynomial time.

(solved by max matching)

Hopcroft-Karp { Adj matrix: $O(V^{2.5})$ Adj List: $O(EV^{\frac{1}{2}})$

14. If $\sum_{i=1}^{n} c_i = C+1$, 則 C 為 prime.

If not, suppose $C = \prod_{i=1}^{n} c_i$, $1 < c_1 \leq c_2 \leq \cdots \leq c_n < C$

$\therefore \forall c_i \geq 2 \quad \therefore C = \prod_{i=1}^{n} c_i \geq 2^{n-1} c_n$,

$\nexists \sum_{i=1}^{n} c_i \leq n c_n \leq 2^{n-1} c_n \leq C < C+1$

6-41 Suppose $d_1, d_2, \ldots, d_n$ are the degree of all vertices of a graph G. Let $1 \le k \le n$. Show that $\sum_{i=1}^{k} d_i \le k(k-1) + \sum_{j=k+1}^{n} min\{k, d_j\}$. (89 中央數學)

解

假設 G 中 n 個點為 $v_1, v_2, \ldots, v_n$, 其 degree 分別為 $d_1, d_2, \ldots, d_n$, 考慮 G 中前面 k 個點 $v_1, v_2, \ldots, v_k$, 其 degree 分別為 $d_1, d_2, \ldots, d_k$, 這 k 個點所分散出去的邊數為 $\sum_{i=1}^{k} d_i$, 其中有些邊可能重複, 而這些邊可以分配到 $v_1, v_2, \ldots, v_k$ 或者分配到其餘的 $n-k$ 個點 $v_{k+1}, v_{k+2}, \ldots, v_n$, 分別討論之

(a) 分配到 $v_1, v_2, \ldots, v_k$:

最壞情形在 $v_1, v_2, \ldots, v_k$ 中任二點皆有邊相連, 則分配到 $v_1, v_2, \ldots, v_k$ 的邊數為 $k(k-1)$, 所以分配到 $v_1, v_2, \ldots, v_k$ 的邊不會超過 $k(k-1)$ 個邊

(b) 分配到 $v_{k+1}, v_{k+2}, \ldots, v_n$:

$\forall j = k+1, k+2, \ldots, n$, v_j 的 degree 為 d_j, 所以 v_j 最多分配到的邊數為 d_j, 另外, 最壞情形 v_j 與 $v_1, v_2, \ldots, v_k$ 皆有邊相連時, 此時 v_j 分配到的邊數為 k, 因此 v_j 分配到的邊數不會超過 $min\{k, d_j\}$

由(1), (2)的討論知 $\sum_{i=1}^{k} d_i \le k(k-1) + \sum_{j=k+1}^{n} min\{k, d_j\}$

範例 3

已知集合 $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, 求 S 具有下列性質的子集個數:

此子集至少含有兩個元素, 且任意兩個元素的差的絕對值大於 1 (98 竹教大資科)

解

假設 a_n 表示 $\{1, 2, \ldots, n\}$ 中具有該性質的子集個數, 將這些子集分成二類

(a) 第一類包含 n, 這類的子集分成 $\{1, 2, \ldots, n-2\}$ 中具有這種性質的子集與 $\{n\}$ 聯集, 以及 $\{1, n\}, \{2, n\}, \ldots, \{n-2, n\}$, 總共的子集個數為 $a_{n-2} + n - 2$

(b) 第二類不含 n, 這相當於 $\{1, 2, \ldots, n-1\}$ 中具有這種性質的子集, 總共的子集個數為 a_{n-1}

所以 $a_n = a_{n-1} + a_{n-2} + n - 2$

當 $n = 3$ 時, $S = \{1, 2, 3\}$, 具這種性質的子集為 $\{1, 3\}$, 所以 $a_3 = 1$

6-105 If $G = (V, E)$ is an undirected graph, a subset D of V is called a *dominating set* if for all $v \in V$, either $v \in D$ or v is adjacent to a vertex in D. If D is a dominating set and no proper subset of D has this property, then D is called *minimal*. The size of any smallest dominating set in G is denoted by $\gamma(G)$ and is called the *domination number* of G.

(a) If G has no isolated vertices, prove that if D is a minimal dominating set, then $V - D$ is a dominating set.

(b) If $I \subseteq V$ is independent, prove that I is a dominating set if and only if I is maximal independent.

(c) Show that $\gamma(G) \le \beta(G)$, and that $|V| \le \beta(G)\chi(G)$.

解。

(a) 若 $V-D$ 不為 dominating set, 則$\exists v \in D$ 使得 v 不與 $V-D$ 中任何點有邊相連

因為 G 無 isolated vertices, 所以 v 在 $D-\{v\}$必與某點有邊相連

因為 D 為 dominating set, 所以 $V-D$ 中任何點皆與 D 中的某點有邊相連

因為 $V-D$ 的點皆不與 v 有邊相連, 所以 $V-D$ 中任何點皆與 $D-\{v\}$中的某點有邊相連

因為 v 與 $D-\{v\}$中某點有邊相連, 所以 $D-\{v\}$為 dominating set 此與 D 為 minimal dominating set 矛盾, 因此 $D-V$ 為 dominating set

(b)

($\Rightarrow$):

若 I 為 independent set 但不為 maximal independent set

$\Rightarrow \exists v \in V-I$ 且 v 與 I 中任何點皆無邊相連

$\Rightarrow I$ 不為 dominating set, 產生矛盾, 所以 I 為 maximal independent set

($\Leftarrow$):

$\forall v \in V-I$

若 v 不與 I 中某點有邊相連

$\Rightarrow I \cup \{v\}$為 independent set, 此與 I 為 maximal independent set 矛盾

所以 v 必與 I 中某點有邊相連

$\Rightarrow I$ 為 dominating set

(c) 考慮 I 為 independent set 且為 dominating set

根據(b)的結果, I 為 maximal independent set

$\Rightarrow |I| \le \beta(G)$

因為$\gamma(G) \le |I|$, 所以$\gamma(G) \le \beta(G)$

令$\chi(G) = m$

$\Rightarrow G$ 可用 m 種顏色 $c_1, \ldots, c_m$ 作正當著色

令 $V_i = \{v \in V \mid v$ 著顏色 $c_i\}, i = 1, 2, \ldots, m$

則 V_i 為 independent set, $\forall i = 1, 2, \ldots, m$

$\Rightarrow |V_i| \le \beta(G), \forall i = 1, 2, \ldots, m$

因為$|V| = |V_1| + \cdots + |V_m|$

$$\Rightarrow |V| = \sum_{i=1}^{m} |V_i| \le \sum_{i=1}^{m} \beta(G) = m\beta(G) = \chi(G)\beta(G)$$

例 51

Show that an n-cube Q_n has a Hamiltonian cycle for any $n \geq 2$.

(86 中興應數)(89, 93 逢甲資工)(89, 93 交大資訊)(93 成大工科)
(95 台科大資管)(95 高雄第一科大資管)(96 成大電機)(96 成大工科)

解

遞迴定義 Gray 碼(Gray code) G_n 如下 :

$G_1 = 0, 1$ 為初始條件, 定義 G_n 根據下列規則

(a) 假設 G_{n-1}^R 表示將 G_{n-1} 中的序列按照反次序排列

(b) 假設 $0G_{n-1}$ 表示將 G_{n-1} 中所有序列前面加上 0

(c) 假設 $1G_{n-1}^R$ 表示將 G_{n-1} 中所有序列前面加上 1

(d) G_n 定義為包含 $0G_{n-1}$ 的所有序列在前面, 接著包含 $1G_{n-1}^R$ 的所有序列在後面

例如:

$0G_1 = 00, 01,\ G_1^R = 1, 0,\ 1G_1^R = 11, 10$

$G_2 = 00, 01, 11, 10$

$0G_2 = 000, 001, 011, 010,\ G_2^R = 10, 11, 01, 00,\ 1G_2^R = 110, 111, 101, 100$

$G_3 = 000, 001, 011, 010, 110, 111, 101, 100$

我們利用對 n 作數學歸納證明 G_n 表示 Q_n 的一個 Hamiltonian cycle

$n = 1$ 時, G_1 顯然表示 Q_1 的一個 Hamiltonian cycle

假設 $n = k$ 時命題成立, 即 G_k 表示 Q_k 的一個 Hamiltonian cycle, 接著考慮 $n = k$

因為 G_k 表示 Q_k 的一個 Hamiltonian cycle, 所以在 G_k 中相鄰二個序列(包含頭尾)恰有一個 bit 不同, 所以 $0G_k$ 仍然維持相鄰二個序列(包含頭尾)恰有一個 bit 不同, 因為 G_k^R 為 G_k 的反次序排列, 所以 G_k^R 中相鄰二個序列(包含頭尾)恰有一個 bit 不同, 因此 $1G_k^R$ 仍然維持相鄰二個序列(包含頭尾)恰有一個 bit 不同, 因為 G_k 的最後一個序列與 G_k^R 的第一個序列相同, 所以 $0G_k$ 的最後一個序列與 $1G_k^R$ 的第一個序列只有最左邊的 bit 不同, 同理 $0G_k$ 的第一個序列與 $1G_k^R$ 的最後一個序列只有最左邊的 bit 不同, 因此, $G_{k+1} = 0G_k, 1G_k^R$ 中相鄰二個序列(包含頭尾)恰有一個 bit 不同, 即 G_{k+1} 表示 Q_{k+1} 的一個 Hamiltonian cycle, 這說明 $n = k+1$ 亦成立, 完成證明 ◆

例 61

假設 S 為 6 個正整數的集合，其最大值為 14，證明 S 的所有非空子集的元素和不可能皆不同

(88 逢甲資工)

解

(1) 我們欲證明有二個非空子集的元素和相同，所以直覺上把非空子集當鴿子，把元素和當鴿籠，總共有 $2^6-1=63$ 種非空子集，元素和最小為 1，最大為 $(9+10+\cdots+14)=69$，元素和的可能方法數最大為 69，發現鴿子數比鴿籠數少，不適用鴿籠原理

(2) 接下來我們把範圍縮小，考慮 S 的非空子集 A 滿足 $|A|\le 5$，A 的元素和記作 s_A

則 $1\le s_A\le 10+11+\cdots+14=60$，另外，$S$ 的非空子集個數為

$$\binom{6}{1}+\binom{6}{2}+\binom{6}{3}+\binom{6}{4}+\binom{6}{5}=62,$$

每一個非空子集皆唯一對應到一個元素和，因為 $62>60$，根據鴿籠原理，存在二個非空子集(事實上它的元素個數最多 5)具有相同的和，完成證明

範例 5

Let $m\in \mathbf{Z}^+$ with m odd. Prove that there exists a positive integer n such that m divides 2^n-1.

(85 成大電機)(88 逢甲資工)(89 朝陽資管)(97 彰師資工)(98 中山資工)

解

考慮 $m+1$ 個正整數 $2^1-1, 2^2-1, \cdots, 2^m-1, 2^{m+1}-1$

根據鴿籠原理，存在二數除以 m 具有相同的餘數

即 $\exists s,t\in \mathbf{Z}^+, 1\le s<t\le m+1$ 使得 (2^s-1)與(2^t-1)除以 m 具相同的餘數

$\Rightarrow m\mid[(2^t-1)-(2^s-1)]$

$\Rightarrow m\mid(2^t-2^s)$

因為 $2^t-2^s=2^s(2^{t-s}-1)$

$\Rightarrow m\mid 2^s(2^{t-s}-1)$

因為 m 為奇數，所以 $m\nmid 2^s$

$\Rightarrow m\mid(2^{t-s}-1)$

取 $n=t-s\in \mathbf{Z}^+$ 使得 $m\mid(2^n-1)$

範例 9

Using the pigeonhole principle, show that in a list of $n^2 + 1$ distinct numbers, there are either $n + 1$ numbers (not necessarily consecutive) in increasing order or $n + 1$ numbers in decreasing order. (For example, in the list 1, 5, 3, 4, 2, we have both the increasing list 1, 3, 4 and the decreasing lists 5, 4, 2 and 5, 3, 2.)

(86 元智資訊)(92 清大資應)(94 中華資工)(95 師大資工)
(95 交大應數)(96 清大資工)(97 北大資工)

解

假設這 n^2+1 個整數序列為 $a_1, a_2, \ldots, a_{n^2+1}$,

令 x_k 及 y_k 分別表示由 a_k 開始最長的遞增及遞減子序列的長度, $k = 1, 2, \ldots, n^2+1$

利用矛盾證法, 假設該整數序列 $a_1, a_2, \ldots, a_{n^2+1}$ 中不存在長度為 $n+1$ 的遞增子序列且不存在長度為 $n+1$ 的遞減子序列

則 $1 \le x_k, y_k, \le n, \forall k = 1, 2, \ldots, n^2+1$

所以集合 $\{(x_k, y_k) \mid k = 1, 2, \ldots, n^2+1\}$ 的元素個數最多為 n^2

由於每個 a_k 唯一對應一組 $(x_k, y_k), \forall k = 1, 2, \ldots, n^2+1$

根據鴿籠原理, $\exists i, j, i < j$ 使得 $(x_i, y_i) = (x_j, y_j)$

因為 $a_i \ne a_j$

若 $a_i < a_j$, 則 $x_i > x_j$, 因此 $(x_i, y_i) \ne (x_j, y_j)$, 產生矛盾

若 $a_i > a_j$, 則 $y_i > y_j$, 因此 $(x_i, y_i) \ne (x_j, y_j)$, 產生矛盾

所以存在長度為 $n+1$ 的遞增子序列或長度為 $n+1$ 的遞減子序列

範例 4

上星期六是政大附小的學校日，三年忠班有35位小朋友，但是只有20位小朋友的家長來參加，在教室裡，家長隨意入座，結果竟然沒有家長坐對位子，請問這種情形共有幾種呢？ (92 政大資管)

解

考慮20位家長坐到35個座位的可能性中，令 a_i 表示第 i 個家長坐對位子的性質，$i=1, 2, \ldots, 20$，欲求 $N(\overline{a}_1\overline{a}_2\cdots\overline{a}_{20}) = S_0 - S_1 + S_2 - S_3 + \cdots + S_{20}$

20個家長坐到35個座位的方法數為 P_{20}^{35}，所以 $S_0 = P_{20}^{35}$

若第 i 個家長坐對位子，相當於19個家長坐到34個座位，方法數為 P_{19}^{34}，即 $N(a_i) = P_{19}^{34}$，，所以 $S_1 = \binom{20}{1} P_{19}^{34}$

若第 i 個家長及第 j 個家長都坐對位子，相當於18個家長坐到33個座位，方法數為 P_{18}^{33}，即 $N(a_i a_j) = P_{18}^{33}$，所以 $S_2 = \binom{20}{2} P_{18}^{33}$，同理 $S_k = \binom{20}{k} P_{20-k}^{35-k}$，$\forall k = 1, 2, \ldots, 20$

因此 $N(\overline{a}_1\overline{a}_2\cdots\overline{a}_{20}) = \sum_{k=0}^{20} (-1)^k \binom{20}{k} P_{20-k}^{35-k}$

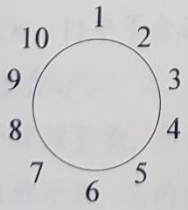
5. (A)

```
karatsuba (u, v, n)
    if (u<10 or v<10)
        return u*v;
    a = higher half of u, b = lower half of u;
    c = higher half of v, d = lower half of v;
    x = karatsuba (b, d);
    y = karatsuba ((b+a), (d+c));
    z = karatsuba (a, c);
    return (z*10^n) + (y-z+x) * 10^⌈n/2⌉ + x;
```

例 41

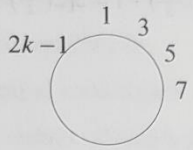
一個被稱為約瑟夫問題(Josephus problem)的描述如下:

總共有 n 個人編號分別為 $1, 2, \ldots, n$, 這 n 個人圍成一個圓圈, 每次越過一個人殺掉第二個人, 直到剩下一個人為止, 假設決定最後存活人的編號為 $J(n)$, 例如 $n = 10$ 時, 如下所示:



被殺掉的人編號依序為 $2, 4, 6, 8, 10, 3, 7, 1, 9$, 最後存活的人為 5 號, 因此 $J(10) = 5$,

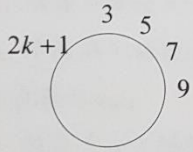
(1) 考慮 $n = 2k$ 為偶數時, 第一輪被殺掉人編號為 $2, 4, 6, \ldots, 2k$, 保留的人編號為 $1, 3, 5, \ldots, 2k-1$



此時相當於 k 個人圍成一圓, 但編號為原先編號的 2 倍再減 1, 所以

$$J(2k) = 2J(k) - 1$$

(2) 考慮 $n = 2k + 1$ 為奇數時, 第一輪被殺掉人編號為 $2, 4, 6, \ldots, 2k, 1$, 保留的人編號為 $3, 5, 7, \ldots, 2k + 1$



此時相當於 k 個人圍成一圓, 但編號為原先編號的 2 倍再加 1, 所以

$$J(2k+1) = 2J(k) + 1$$

根據(1), (2), 加上初始條件 $J(1) = 1$ 得下列遞迴關係

$$\begin{cases} J(2k) = 2J(k) - 1, & k \geq 1 \\ J(2k+1) = 2J(k) + 1, & k \geq 1 \\ J(1) = 1 \end{cases}$$

將 $J(n)$ 列表如下:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$J(n)$	1	1	3	1	3	5	7	1	3	5	7	9	11	13	15	1

因此我們可以猜出 $J(2^m + t) = 2t + 1$, 其中 $m \geq 0, 0 \leq t < 2^m$,

範例 4

n cities are connected by a network of k highway. (A highway is defined to be a road between two cities that does not go through any intermediate cities.) Show that if $k > \frac{1}{2}(n-1)(n-2)$, then one can always travel any two cities through connecting highways.

(85 中山資工)(92 交大應數)(95 暨大資工)(95 輔大電子)(96 政大資科)(98 清大資應)

解

假設這 n 個 cities 形成一個 graph $G=(V, E)$, 其中 $|V|=n$, $|E|=k$

欲證若 $k > \frac{1}{2}(n-1)(n-2)$, 則 G 為 connected graph

利用矛盾證法, 若 G 為 disconnected graph

則 G 中含 r 個 components $G_i=(V_i, E_i)$, 其中 $i=1, 2, \ldots, r$, $r \geq 2$

令 $|V_i|=n_i$, $i=1, 2, \ldots, r$

當每個 component 皆為 complete graph 時具有最多邊數, 此時的邊數為

$$k=\sum_{i=1}^{r}\binom{n_i}{2}, \text{ 其中 } n_1+n_2+\cdots+n_r=n$$

很顯然地當 $r=2$ 時具最多邊數 $k=\binom{n_1}{2}+\binom{n_2}{2}$, 其中 $n_1+n_2=n$

$$\Rightarrow \binom{n_1}{2}+\binom{n_2}{2}=\binom{n_1}{2}+\binom{n-n_1}{2}=\frac{1}{2}n_1(n_1-1)+\frac{1}{2}(n-n_1)(n-n_1-1)$$

$$=n_1^2-nn_1+\frac{1}{2}(n^2-n)$$

假設 $f(n_1)=n_1^2-nn_1+\frac{1}{2}(n^2-n)$, 將 f 視為一個 $f: \boldsymbol{R} \to \boldsymbol{R}$ 的函數

則 $f'(n_1)=2n_1-n$, 假設 $f'(n_1)=0$ 得 $n_1=\frac{n}{2}$, 另外, $f''(n_1)=2>0$

所以 f 在 $\frac{n}{2}$ 具有小值

這說明式子 $n_1^2-nn_1+\frac{1}{2}(n^2-n)$ 在 $n_1=1$ 或 $n-1$ 時具有最大值, 它的值為

$$\frac{1}{2}(n-1)(n-2)$$

所以 $k \leq \frac{1}{2}(n-1)(n-2)$, 這產生矛盾, 得 G 為 connected graph

例 52

Prove that if $G = (V, E)$ is a loop-free undirected graph with $|V| = n \geq 3$, and if $|E| \geq \binom{n-1}{2} + 2$, then G has a Hamiltonian cycle. (95 彰師資工)

解。

$\forall x, y \in V$, x, y 不相鄰，欲證 $\deg(x) + \deg(y) \geq n$

假設 $G' = (V - \{x, y\}, E')$ 為 G 中去掉 x, y 後的圖

因為 $\{x, y\} \notin E$

$\Rightarrow |E| = |E'| + \deg(x) + \deg(y)$

$\Rightarrow |E'| = |E| - (\deg(x) + \deg(y))$

因為 G' 具有 $n-2$ 個點

$\Rightarrow |E'| \leq \binom{n-2}{2}$

$\Rightarrow |E| - (\deg(x) + \deg(y)) \leq \binom{n-2}{2}$

$\Rightarrow \deg(x) + \deg(y) \geq |E| - \binom{n-2}{2} \geq \binom{n-1}{2} + 2 - \binom{n-2}{2}$

$= \frac{(n-1)(n-2)}{2} + 2 - \frac{(n-2)(n-3)}{2} = n$

根據定理 6–10, G 具 Hamiltonian cycle

(c) (4%) Give an $O(|V| + |E|)$ time algorithm to solve the following problem: Given a graph G and a positive integer k, determine whether the value of the bottleneck spanning tree is at most k.

11(C) 利用 DFS 且 weight 大者優先走，則 DFS ST 中最大者再與 k 比較

10. (5%) Given the matrix A and the vector $\boldsymbol{b}$ in the above problem, which of the following vectors $\boldsymbol{c}$ will make $A\boldsymbol{x} = \boldsymbol{c}$ have the same shortest (minimal 2-norm) least-squares solution as the system $A\boldsymbol{x} = \boldsymbol{b}$?

(a) $\boldsymbol{c} = (2, 1, 1)^T$.

(b) $\boldsymbol{c} = (2-\sqrt{3}, 1+\sqrt{3}, \sqrt{3})^T$.

(c) $\boldsymbol{c} = (2.5, 0.5, 0.5)^T$.

(d) $\boldsymbol{c} = (1, 2, 2)^T$.

(e) $\boldsymbol{c} = (2/\sqrt{2}, 1/\sqrt{2}, 1/\sqrt{2})^T$.

解 (a), (c), (d)

假設 $A \in \boldsymbol{R}^{m\times n}$, $A = U\Sigma V^T$ 為 A 的 singular value decomposition

$A\boldsymbol{x} = \boldsymbol{b}$ 與 $A\boldsymbol{x} = \boldsymbol{c}$ 具有相同的 shortest least-squares solution $\hat{\boldsymbol{x}} = A^+\boldsymbol{b} = A^+\boldsymbol{c}$，其中 $A^+ = V\Sigma^+U^T$ 為 A 的 pseudoinverse

$\Leftrightarrow A^+(\boldsymbol{b} - \boldsymbol{c}) = \boldsymbol{0}$

$\Leftrightarrow \boldsymbol{b} - \boldsymbol{c} \in N(A^+)$

1-63 Use mathematical induction to show that

$$\sum \frac{1}{n_1 \cdot n_2 \cdots n_k} = n$$

where the sum is taken over all nonempty subsets $\{n_1, n_2, \ldots, n_k\}$ of $\{1, 2, \ldots, n\}$.

(88 台科大資工)

假設 $A_n = \{1, 2, \ldots, n\}$，欲證 $\sum_{\{n_1, n_2, \ldots, n_k\} \subseteq A_n} \frac{1}{n_1 \cdot n_2 \cdots n_k} = n$

對 n 作數學歸納證明，$n = 1$ 時，$\sum_{\{n_1, n_2, \ldots, n_k\} \subseteq A_1} \frac{1}{n_1 \cdot n_2 \cdots n_k} = \frac{1}{1} = 1$ 成立

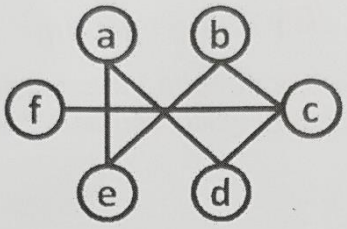
假設 $n = p$ 時命題成立，即 $\sum_{\{n_1, n_2, \ldots, n_k\} \subseteq A_p} \frac{1}{n_1 \cdot n_2 \cdots n_k} = p$，接著考慮 $n = p + 1$

$$\sum_{\{n_1, n_2, \ldots, n_k\} \subseteq A_{p+1}} \frac{1}{n_1 \cdot n_2 \cdots n_k}$$

$$= \left(\sum_{\{n_1, n_2, \ldots, n_k\} \subseteq A_p} \frac{1}{n_1 \cdot n_2 \cdots n_k} \right) + \frac{1}{p+1} + \frac{1}{p+1} \left(\sum_{\{n_1, n_2, \ldots, n_k\} \subseteq A_p} \frac{1}{n_1 \cdot n_2 \cdots n_k} \right)$$

$= p + \frac{1}{p+1} + \frac{1}{p+1} \cdot p = p + 1$，所以 $n = p + 1$ 亦成立，得證

8. (8%) Given a graph $G = (V, E)$, a subset of vertices $S \subseteq V$ is said to be a ***k-plex*** if the minimum degree of the subgraph induced by S on G is at least $|S| - k$. For example, $S = \{a, b, c, d, e\}$ is a ***3-plex*** with $|S| = 5$. In contrast, $S^* = \{a, b, c, d, e, f\}$ is not a ***3-plex***.



8-1 (4%) The ***k-plex problem*** is defined as follows: Given a graph $G = (V, E)$, positive integers k and p, does G contain a subset of vertices S such that $|S| = p$ and S is a ***k-plex***? Either provide a polynomial time algorithm to solve the k-plex problem or prove that this problem is NP-hard.

k-plex 即找 a subset of vertices S, $\forall v \in S$, $\deg(v) \geq p-k$

可從 clique problem reduce 至 k-plex problem.

所以 k-plex 為 NP-hard.

8-2 (4%) Given a ***k-plex*** S and any nonempty subset $S' \subseteq S$, is S' also a ***k-plex***? If yes, please provide the proof. If no, please provide a counterexample.

假設 S 為 k-plex, 若存在 $S' \subseteq S$ 且 S' 不為 k-plex,

令 H 為 G 之 subgraph induced by S, 又 S' 不為 k-plex,

H' 之 min deg $\geq |S'| - k$, $\forall v \in S'$ 為 H' 中 min deg 之 vertices,

則除 S' 中 vertices 之外, v 在 H 中最多和 $|S| - |S'|$ 個 vertices 相連, 所以 H 之 min deg $\geq |S'| - k + |S| - |S'| = |S| - k$, 矛盾

2. (9%) Prove that every tree with a vertex of degree k has at least k leaves.

2. 設 $T=(V,E)$ 為 tree 且 leaves 數為 l, $|V|=n$

當 $k=1$ 時，成立。

考慮 $k\geq 2$，假設 T 中存在一 vertex degree $=k$ 且 $l\leq k-1$

設 $m=\sum_{v\in V}\deg(v)=2|E|=2(|V|-1)=2(n-1)=2n-2$

∵ 有 l 個 vertices degree $=1$，一 vertex degree $=k$，

所以另外有 $n-l-1$ 個 vertices degree ≥ 2，則

$$m=\sum_{v\in V}\deg(v)=2n-2\geq l+k+2(n-l-1)$$
$$=2n-l+k-2$$
$$=2n+(k-1)-(l-1)$$
$$\geq 2n+l-l-1=2n-1 \text{ 矛盾}$$

2 Consider ternary strings with symbols 0, 1, 2 used. For $n\geq 1$, let a_n count the number of ternary strings of length n, where there are no consecutive 1's and no consecutive 2's. Show that a_n can be expressed recursively as $2a_{n-1}+a_{n-2}$. (10%)

2. b_n = 第 n 位為 0

c_n = 第 n 位為 1，第 $n-1$ 位為 2 or 0 (不含連續 1 or 2)

d_n = 第 n 位為 2，第 $n-1$ 位為 1 or 0

→ $a_n=b_n+c_n+d_n$ (第 $n-1$ 位不為 1) (第 $n-1$ 位不為 2)

$$=a_{n-1}+(a_{n-1}-c_{n-1})+(a_{n-1}-d_{n-1})$$
$$=3a_{n-1}-c_{n-1}-d_{n-1} \quad (\because a_{n-1}=b_{n-1}+c_{n-1}+d_{n-1})$$
$$=2a_{n-1}+b_{n-1}$$
$$=2a_{n-1}+b_{n-2}$$

12. (6%) In a type-2 grammar, every production is of the form $A \to \alpha$, where A is a single non-terminal, and α is a string consisting of terminals and non-terminals. Please find a type-2 grammar for $L = \{a^n b^m c^k : k = |n - m|\}$.

CFG

12. $L = \{a^n b^m c^{k=|n-m|}\}$

$= \{a^{n=m+k} b^m c^k \mid k, m \geq 0\} \cup \{a^n b^{m=n+k} c^k \mid n \geq 0, k \geq 0\}$

$= L_1 \cup L_2$

L_1: $A \to aAc \mid C$, $C \to aCb \mid \lambda$

L_2: $B \to DE$, $D \to aDb \mid \lambda$, $E \to bEc \mid bc$

$\to L$:

$S \to A \mid B$

$A \to aAc \mid C$

$C \to aCb \mid \lambda$

$B \to DE$

$D \to aDb \mid \lambda$

$E \to bEc \mid bc$

10. (6%) Let n be an integer, and S be a set of integers, with range from 1 to n^2. It is known that S has at least $\sqrt{n}$ items. Explain in details how to sort S in O($|S|$) time.

10. 利用 counting sort (1 ~ 10000 共 10 個)

pass1: key i % $\sqrt{n}$

pass2: $\lceil$key i / $\sqrt{n}\rceil$ % $\sqrt{n}$

pass3: $\lceil$key i / n$\rceil$ % $\sqrt{n}$ > time complexity $5O(\sqrt{n})$

pass4: $\lceil$key i / n$\sqrt{n}\rceil$ % $\sqrt{n}$ $= O(\sqrt{n})$

pass5: $\lceil$key i / $n^2\rceil$ % $\sqrt{n}$ $= O(|S|)$

Suppose $A_{3\times3}$ has three distinct eigenvalues 0, 1, 3 with corresponding eigenvectors $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$. Which of the following statements are true?

(a) The rank of A must be 2.

(b) The matrix $A^2 - I$ is invertible.

(c) $\boldsymbol{v}$ and $\boldsymbol{w}$ must span the column space of A.

(d) The least-squares error of $A\boldsymbol{x} = 2\boldsymbol{v} + 3\boldsymbol{w} + \boldsymbol{u}$ must be $\|\boldsymbol{u}\|^2$.

(d) False:

當 A 為 symmetric 時, $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ 為 orthogonal, 假設 $\boldsymbol{b} = 2\boldsymbol{v} + 3\boldsymbol{w} + \boldsymbol{u}$

因為 $R(A) = span\{\boldsymbol{v}, \boldsymbol{w}\}$

所以 $proj_{R(A)}\boldsymbol{b} = \frac{<\boldsymbol{b},\boldsymbol{v}>}{<\boldsymbol{v},\boldsymbol{v}>}\boldsymbol{v} + \frac{<\boldsymbol{b},\boldsymbol{w}>}{<\boldsymbol{w},\boldsymbol{w}>}\boldsymbol{w}$

$$= \frac{<2\boldsymbol{v}+3\boldsymbol{w}+\boldsymbol{u},\boldsymbol{v}>}{<\boldsymbol{v},\boldsymbol{v}>}\boldsymbol{v} + \frac{<2\boldsymbol{v}+3\boldsymbol{w}+\boldsymbol{u},\boldsymbol{w}>}{<\boldsymbol{w},\boldsymbol{w}>}\boldsymbol{w} = 2\frac{<\boldsymbol{v},\boldsymbol{v}>}{<\boldsymbol{v},\boldsymbol{v}>}\boldsymbol{v} + 3\frac{<\boldsymbol{w},\boldsymbol{w}>}{<\boldsymbol{w},\boldsymbol{w}>}\boldsymbol{w}$$

$$= 2\boldsymbol{v} + 3\boldsymbol{w}$$

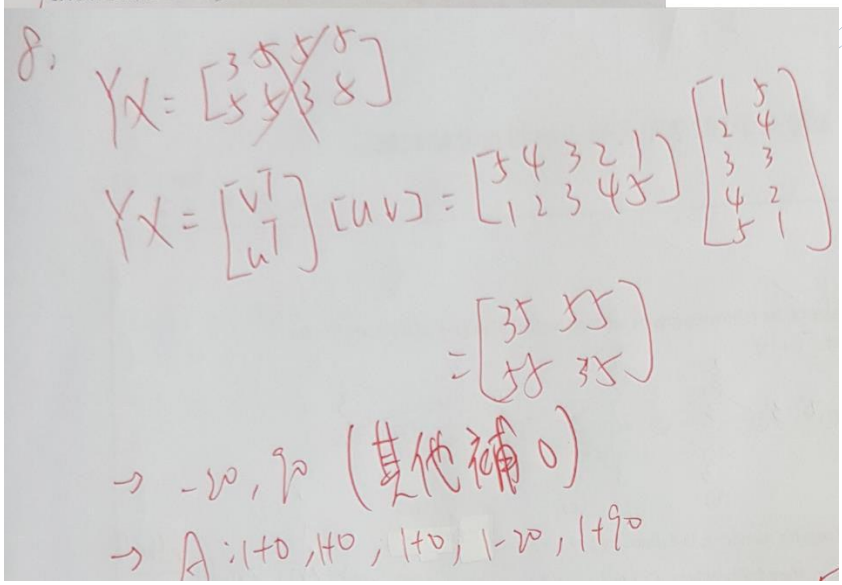
因此 $A\boldsymbol{x} = \boldsymbol{b}$ 的 least square error 為 $\|proj_{R(A)}\boldsymbol{b} - \boldsymbol{b}\| = \|\boldsymbol{u}\|$

[註]：有些書的 least square error 定義為 $\|proj_{R(A)}\boldsymbol{b} - \boldsymbol{b}\|^2 = \|\boldsymbol{u}\|^2$

本題因為未假設 A 為 symmetric, 也就是 $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ 未必為 orthogonal, 所以無法保證 $A\boldsymbol{x} = \boldsymbol{b}$ 的 least square error 為 $\|\boldsymbol{u}\|$, 反例可取三個不 orthogonal 的向量 $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$

8. If $u = \begin{bmatrix}1\\2\\3\\4\\5\end{bmatrix}$, $v = \begin{bmatrix}5\\4\\3\\2\\1\end{bmatrix}$, and $A = I_5 + [u \quad v]\begin{bmatrix}v^T\\u^T\end{bmatrix}$,

then all the eigenvalues of A are_____. (5%)



Suppose $A_{3\times3} = \boldsymbol{x}\boldsymbol{y}^T$ is a rank-1 matrix. Which of the following statements are true?

(a) The eigenvector matrix S (i.e., $AS = \Lambda S$) must be invertible.

(b) A must have one non-zero eigenvalue.

(c) When A is factorized by singular value decomposition (SVD) into $U\Sigma V^T$, the only non-zero singular value is $||\boldsymbol{x}||\,||\boldsymbol{y}||$.

(d) When A is factorized by SVD into $U\Sigma V^T$, U must include $\dfrac{\boldsymbol{x}}{||\boldsymbol{x}||}$ as one of its column vectors.

(c) True:

假設 $B = A^TA = (\boldsymbol{x}\boldsymbol{y}^T)^T(\boldsymbol{x}\boldsymbol{y}^T) = \boldsymbol{y}\boldsymbol{x}^T\boldsymbol{x}\boldsymbol{y}^T = (\boldsymbol{x}^T\boldsymbol{x})\boldsymbol{y}\boldsymbol{y}^T$

因為 $B\boldsymbol{y} = (\boldsymbol{x}^T\boldsymbol{x})\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{y} = (\boldsymbol{x}^T\boldsymbol{x})(\boldsymbol{y}^T\boldsymbol{y})\boldsymbol{y} = \lambda\boldsymbol{y}$, 其中 $\lambda = (\boldsymbol{x}^T\boldsymbol{x})(\boldsymbol{y}^T\boldsymbol{y}) = ||\boldsymbol{x}||^2||\boldsymbol{y}||^2$

所以 λ 為 B 的 eigenvalue

假設 $W = span\{\boldsymbol{y}\}$, 則 $\dim(W) = 1$, 所以 $\dim(W^{\perp}) = 2$, 取 $\{\boldsymbol{y}_1, \boldsymbol{y}_2\}$ 為 $W^{\perp}$ 的一組 basis

因為 $B\boldsymbol{y}_i = (\boldsymbol{x}^T\boldsymbol{x})\boldsymbol{y}\boldsymbol{y}_T\boldsymbol{y}_i = (\boldsymbol{x}^T\boldsymbol{x})(\boldsymbol{y}^T\boldsymbol{y}_i)\boldsymbol{y} = \boldsymbol{0} = 0\boldsymbol{y}_i$, $i = 1, 2$, 所以 0 為 B 的 eigenvalue

因為 $\boldsymbol{y}, \boldsymbol{y}_1, \boldsymbol{y}_2$ 為 B 的三個線性獨立 eigenvectors

所以 B 除了 λ 及 0 以外不具其他 eigenvalue

因為 A 的 singular value 為 $B = A^TA$ 的 eigenvalue 開根號

所以 A 的 singular value 為 $\sqrt{\lambda}$, 0, 0, 因此 A 的唯一的非零 singular value 為

$\sigma = \sqrt{\lambda} = ||\boldsymbol{x}||\,||\boldsymbol{y}||$

(d) True:

因為 $\lambda = ||\boldsymbol{x}||^2||\boldsymbol{y}||^2$ 為 A^TA 的 eigenvalue 且對應的 eigenvector 為 $\boldsymbol{y}$

所以 $\boldsymbol{v} = \dfrac{\boldsymbol{y}}{||\boldsymbol{y}||}$ 為 V 的某一個行向量, 因此 $\boldsymbol{u} = \dfrac{1}{\sigma}A\boldsymbol{v}$ 為 U 的某一個行向量

因為 $\boldsymbol{u} = \dfrac{1}{\sigma}A\boldsymbol{v} = \dfrac{1}{||\boldsymbol{x}||\,||\boldsymbol{y}||}\boldsymbol{x}\boldsymbol{y}^T\dfrac{\boldsymbol{y}}{||\boldsymbol{y}||} = \dfrac{\boldsymbol{y}^T\boldsymbol{y}}{||\boldsymbol{x}||\,||\boldsymbol{y}||^2}\boldsymbol{x} = \dfrac{||\boldsymbol{y}||^2}{||\boldsymbol{x}||\,||\boldsymbol{y}||^2}\boldsymbol{x} = \dfrac{\boldsymbol{x}}{||\boldsymbol{x}||}$

所以 $\dfrac{\boldsymbol{x}}{||\boldsymbol{x}||}$ 為 U 的某一個行向量

5. (4%) What is the largest n so that the following assertion is always true?

Assertion Let G be a graph with 10 vertices in which there is at least one edge among any three vertices. Then G must contain K_n, where K_n is the complete graph with n vertices.

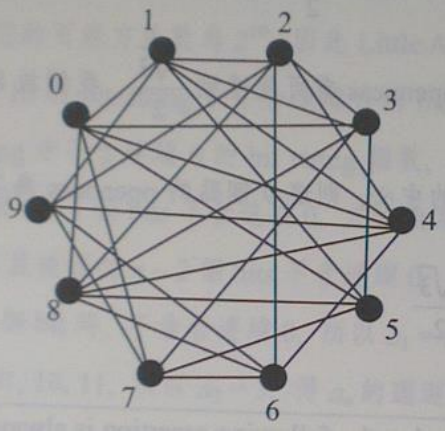
解

首先證明 G 中必含有一 induced subgraph 為 K_4，假設 G 中含有一點 v, $\deg(v) \leq 5$，因為 G 中的任三點間須至少有兩點相鄰，所以考慮所有 G 中與 v 不相鄰的點，任取兩點皆須有邊相鄰，因此形成 K_m, $n \geq 5$。否則，G 中所有點的 degree 皆至少為 6，則任取一點 x，考慮 6 個與 x 相鄰的點，其中先固定一點 a，並將其他 5 點分成兩堆，一堆與 a 相鄰，另一堆與 a 不相鄰，根據鴿籠原理，至少有一堆點的個數至少為 $\left\lceil \frac{5}{2} \right\rceil = 3$，在此分成下列二種情形討論：

(1) 若與 a 相鄰那堆有至少三個點 $\{b, c, d\}$，因為該三點之間必連有至少一邊，不失一般性令 $(b, c) \in E$，則 $\{x, a, b, c\}$ 的 induced subgraph 形成 K_4，得證

(2) 若不與 a 相鄰那堆有至少三個點 $\{b, c, d\}$，因為根據題目已知，考慮 $\{b, c, d\}$ 中任取兩點再加上 a，其中必含有一邊，所以 $\{b, c, d\}$ 的 induced subgraph 會形成 K_3，則 $\{x, b, c, d\}$ 的 induced subgraph 形成 K_4，得證

欲說明 G 中不一定含有具 4 個點以上的完全圖，下圖為一例子滿足任三點間必含一邊，且其中最大的 complete subgraph 為 K_4，因此我們證明了 $n = 4$ 確實為最大值



7. (5 points) Suppose $\mathbf{A_{4x4} = xy^T}$ is a rank-1 real matrix with $\mathbf{x^Ty = 0}$ and $\mathbf{\|x\|_2 = \|y\|_2 = 1}$. Which of the following statements are TRUE?
   (A) $\mathbf{A}$ must have a non-zero eigenvalue.
   (B) There must be an invertible eigenvector matrix $\mathbf{S}$ ($\mathbf{AS = S\Lambda}$).
   (C) The nullspace of $\mathbf{A}$ contains its column space, i.e., $\mathbf{N(A) \supset C(A)}$.
   (D) $\mathbf{v = x + y}$ can be the shortest (minimal 2-norm) least-squares solution to some problem of the form $\mathbf{Av = b}$.
   (E) When $\mathbf{A}$ is factorized by singular value decomposition (SVD) into $\mathbf{U\Sigma V^T}$, the only non-zero singular value is 1.

(c) True:

因為 $A = xy^T$, 所以 $C(A) \subseteq C(x) = \text{span}\{x\}$

$\forall v \in C(A)$, $v = cx$, for some $c \in R$

$\Rightarrow Av = (xy^T)v = xy^T(cx) = cx(y^Tx) = cx0 = 0$, 則 $v \in N(A)$

因此 $C(A) \subseteq N(A)$

(d) False:

假設 $\hat{x}$ 為 $Av = b$ 的 shortest least-squares solution,

則 $\hat{x} = A^+b \in R(A^+)$, 其中 A^+為 A 的 pseudoinverse

因為 $R(A^+) = R(A^T) = span\{y\} \Rightarrow \hat{x} = A^+b \in span\{y\}$

因為 x, y 為 linearly independent, 所以 $x + y \notin span\{y\}$

因此 $x + y$ 不為 $Av = b$ 的 shortest least-squares solution

(e) True:

假設 $B = A^TA = (xy^T)^T(xy^T) = yx^Txy^T = (x^Tx)yy^T$

因為 $By = (x^Tx)yy^Ty = (x^Tx)(y^Ty)y = \lambda y$, 其中 $\lambda = (x^Tx)(y^Ty) = \|x\|^2\|y\|^2$

所以 λ 為 B 的 eigenvalue

假設 $W = span\{y\}$, 則 $\dim(W) = 1$, 所以 $\dim(W^\perp) = 3$

取 $\{y_1, y_2, y_3\}$ 為 $W^\perp$ 的一組 basis

因為 $By_i = (x^Tx)yy^Ty_i = (x^Tx)(y^Ty_i)y = 0 = 0y_i$, $i = 1, 2, 3$, 所以 0 為 B 的 eigenvalue

因為 y_1, y_2, y_3 為 B 的三個線性獨立 eigenvectors

所以 B 除了 λ 及 0 以外不具其他 eigenvalue

因為 A 的 singular value 為 $B = A^TA$ 的 eigenvalue 開根號

所以 A 的 singular value 為 $\sqrt{\lambda}$, 0, 0, 0, 因此 A 的唯一的非零 singular value 為

$\sigma = \sqrt{\lambda} = \|x\| \|y\| = 1$

The line graph $L(G) = (V', E')$ of an undirected simple graph $G = (V, E)$ is defined by

(1) $V' = E$, and

(2) $E' = \{((u,v),(v,w)) | (u,v) \in E, (v,w) \in E, u \neq w\}$

Put another way, (1) the edges of G are the vertices of $L(G)$, and (2) there is an edge connecting vertices (u,v) and (v,w) in $L(G)$ iff the corresponding edges share the same endpoint v in G.

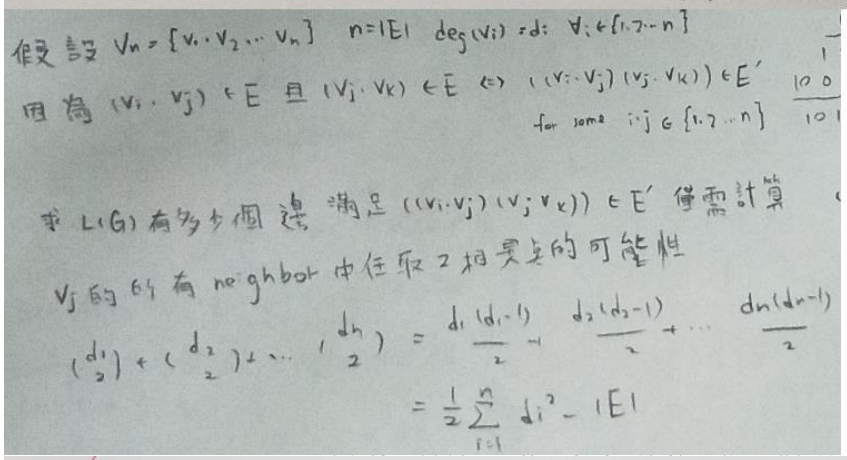
(a) (3 points) For each statement below, determine if it is true.

(1) If G has an Euler circuit, then $L(G)$ has a Hamiltonian circuit.

(2) If G is disconnected, then $L(G)$ is disconnected.

(3) If G is a cycle graph, then G and $L(G)$ are isomorphic.

(b) (4 points) How many edges are there in $L(G)$? Hint: Count the degrees



4-207 Let A and B be two $n \times n$ matrices such that $AB = O$. Show that the nullspace of BA has dimension at least $n/2$. (97 台大數學)

解:

因為 $AB = O$, 所以 $BAB = B \cdot O = O$

$\Rightarrow BABx = \mathbf{0}$, $\forall x \in F^n$

$\Rightarrow CS(B) \subseteq \ker(BA)$

$\Rightarrow \dim(CS(B)) \leq \dim(\ker(BA))$

$\Rightarrow rank(B) \leq nullity(BA) = n - rank(BA)$

因為 $rank(BA) \leq rank(B)$, 加上 $rank(B) \leq n - rank(BA)$

所以 $n - rank(BA) \geq rank(B) \geq rank(BA)$

$\Rightarrow 2\, rank(BA) \leq n$

$\Rightarrow rank(BA) \leq \frac{n}{2}$, 所以 $nullity(BA) = n - rank(BA) \geq \frac{n}{2}$

3. Find a positive integer p such that $\frac{(p+13)!}{p!13!} \equiv 7 \pmod{13}$ or show that such an integer does not exist. Prove the correctness of your answer. (15%)

參考解析 可以取 $p=78$。

根據上標和公式 $\binom{n}{13}=\binom{12}{12}+\binom{13}{12}+\binom{14}{12}+\cdots+\binom{n-2}{12}+\binom{n-1}{12}$，

且其中 $\binom{i}{12} \equiv \begin{cases} 1 & i \equiv 12 \bmod 13 \\ 0 & \text{else} \end{cases} \pmod{13}$，證明如後 (*)

故若取 $p=78$，則 $\frac{(p+13)!}{p!13!}=\binom{91}{13}$

$$=\left(\binom{12}{12}+\binom{13}{12}+\ldots+\binom{24}{12}\right)+\left(\binom{25}{12}+\binom{26}{12}+\cdots\binom{37}{12}\right)+\cdots$$

$$+\left(\binom{77}{12}+\binom{78}{12}+\cdots\binom{89}{12}\right)+\binom{90}{12} \quad \rightarrow 90+1-13=78$$

$$\equiv \underbrace{(1+\overbrace{0+\ldots+0}^{12\text{個}})+(1+\overbrace{0+\ldots+0}^{12\text{個}})+\cdots+(1+\overbrace{0+\ldots+0}^{12\text{個}})}_{6\text{組}}+1=7 \pmod{13}$$

證明：

若 $i \equiv 12 \pmod{13}$，則

令 $x=\binom{i}{12}=\frac{i\times(i-1)\times(i-1)\times\cdots\times(i-11)}{12!} \pmod{13}$，

則 $12!\times x \equiv i\times(i-1)\times(i-2)\times\cdots\times(i-11) \equiv 12\times11\times10\times\cdots\times1 \pmod{13}$，

而因為 12!與 13 互質，故 $x \equiv \binom{i}{12} \equiv 1 \pmod{13}$。

若 $i \equiv r \pmod{13}$，$r \neq 12$，則

$i\times(i-1)\times(i-2)\times\cdots\times(i-11)$ 中間恰有一數為 13 的倍數，

故 $\binom{i}{12}=\frac{i\times(i-1)\times(i-2)\times\cdots\times(i-11)}{12!}$ 為 13 的倍數，

$\therefore \binom{i}{12} \equiv 0 \pmod{13}$。

9. (a) (5 points). Define $F(x) = x/2$ if x is even. Otherwise, $F(x) = F(F(5x+3))$. Prove that we can compute $F(x)$ in a finite amount of time for any positive integer x.

(a) 若 x is even, $F(x) = x/2$, 結束。

若 x is odd, 則 $5x+3$ is even, 分二種情況:

① $(5x+3)/2$ is even, 則 $F(x) = (5x+3)/4$, 結束。

② $(5x+3)/2$ is odd, 則 $\exists x' = (5x+3)/2$, $F(x) = F(x')$.

若經有限次計算, 則必 $\exists x'$, s.t. $4 \mid 5x'+3$

$5x+3 = x + (4x+3)$, 令 $x = (b_{n-1} \cdots b_0)_2$, $b_i \in \{0,1\}$

$\Rightarrow 4x = (b_{n-1} \cdots b_0 00)_2$, $4x+3 = (b_{n-1} \cdots b_0 11)_2$

∵ x is odd, x 的右端必只有有限個 1, 每次計算 $(5x+3)/2$ 時, 皆會將右端減去一個 1 直到連續 2 個 0 出現, 即 $4 \mid (5x+3)$, 經過有限次

5. Suppose $A \in \mathbf{R}^{3\times 3}$ and $\det(xI_3 - A) = x^3 - x^2 + 3x - 2$, then $\det(xI_3 - A^2) =$ _____. (5%)

5. $P_A(x) = -x^3 + x^2 - 3x + 2$

⇒ 有特徵根 a, b, c

$$\begin{cases} a+b+c = 1 \\ ab+bc+ac = 3 \\ abc = 2 \end{cases} \Rightarrow \begin{cases} a^2+b^2+c^2 = 1^2 - 2\times 3 = -5 \\ a^2b^2+b^2c^2+a^2c^2 = 3^2 - 2\times 2\times 1 = 5 \\ a^2b^2c^2 = 2^2 = 4 \end{cases}$$

$\Rightarrow P_{A^2}(x) = \det(A^2 - Ix) = -x^3 - 5x^2 - 5x + 4$

$\Rightarrow \det(xI - A^2) = x^3 + 5x^2 + 5x - 4$

4. (10%) Define function $f : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ as

$$f(x, y) = x_1y_1 - x_1y_2 - x_2y_1 + 4x_2y_2,$$

for any vector x (respectively, y) with standard coordinate (x_1, x_2) (respectively, (y_1, y_2)). Let

$$u = (1, 0).$$

Find a vector v such that

$$f(x, y) = s_1t_1 + s_2t_2,$$

where (s_1, s_2) (respectively, (t_1, t_2)) is the coordinate of x (respectively, y) with respect to the ordered basis of $\mathbb{R}^2$ consisting of u and v.

5. (10%) Find the pseudo-inverse of

$$\frac{1}{6}\begin{pmatrix} 1 & 1 & 1 \\ 3 & 0 & -3 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

$\beta = \{(1,0), (0,1)\}$

$\gamma = \{u = (1,0), v = (a,b)\}$

$A^\dagger = (A^TA)^{-1}A^T$

$[x]_\beta = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $[y]_\beta = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$, $[x]_\gamma = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$, $[y]_\gamma = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$

$[I]_\gamma^\beta = \begin{bmatrix} 1 & a \\ 0 & b \end{bmatrix}$ 見背面

$\to [x]_\beta = \begin{bmatrix} s_1 + s_2a \\ s_2b \end{bmatrix}$, $[y]_\beta = \begin{bmatrix} t_1 + t_2a \\ t_2b \end{bmatrix}$

$\to f(x,y) = (s_1 + s_2a)(t_1 + t_2a - t_2b) + s_2b(-t_1 - t_2a + 4t_2b)$

$= s_1t_1 + s_2t_2 \to a = b = \pm\frac{1}{\sqrt{3}}$

比較係數

$x = u - 1$

$y = \frac{1}{2}v + \frac{\sqrt{3}}{2}w + \left(-1 - \frac{\sqrt{3}}{2}\right)$

$z = -\frac{\sqrt{3}}{2}v + \frac{1}{2}w + \left(\sqrt{3} - \frac{1}{2}\right)$

$1 = 1$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} & -1 - \frac{\sqrt{3}}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{1}{2} & \sqrt{3} - \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$$

$O_2 = O_1 - e_u - 2e_v - e_w$

$r_1 = xe_x + ye_y + ze_z$

$r_2 = ue_u + ve_v + we_w$

$r = r_1 + O_1 = r_2 + O_2$

$\to r = (u-1)e_u + (v-2)e_v + (w-1)e_w + O_1$

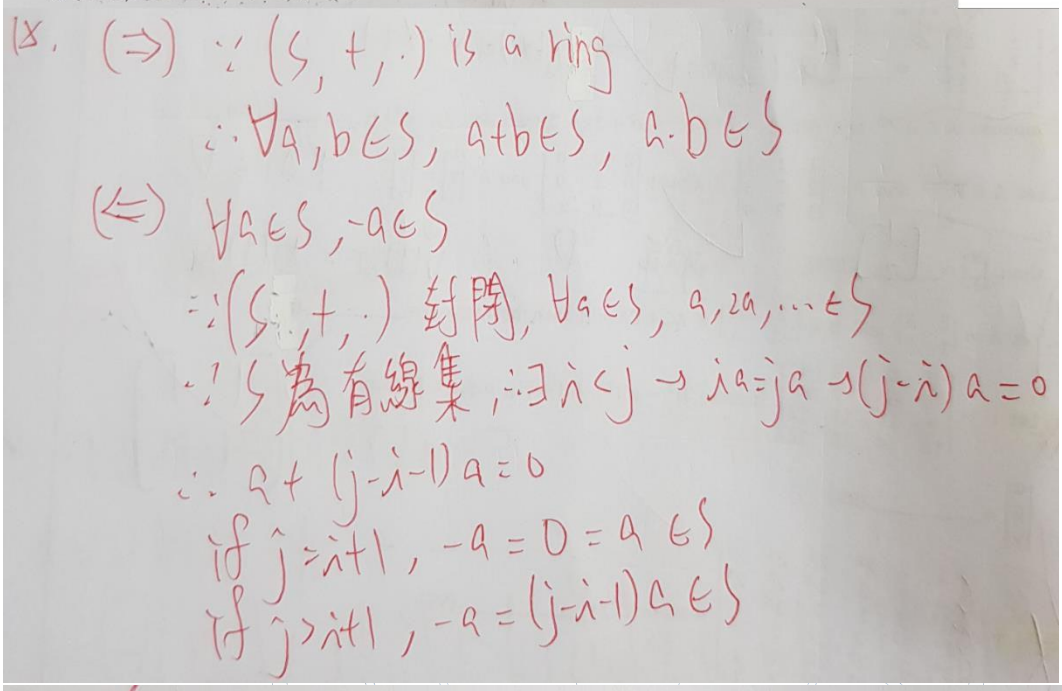
$= (u-1)e_x + (v-2)\left(\frac{1}{2}e_y - \frac{\sqrt{3}}{2}e_z\right)$

$+ (w-1)\left(\frac{\sqrt{3}}{2}e_y + \frac{1}{2}e_z\right) + O_1$

15 Suppose that $(R, +, \cdot)$ is a ring and S is a nonempty subset of R. Then, $(S, +, \cdot)$ is a ring if and only if

- for all $a, b \in S$, $a+b \in S$ and $a \cdot b \in S$;
- for all $a \in S$, $-a \in S$.

Please show that when S is finite, $(S, +, \cdot)$ is a ring if and only if for all $a, b \in S$, $a+b \in S$ and $a \cdot b \in S$. (10%)



2-102 Let $A, B \in F^{n\times n}$, $A^2 = I = B^2$, $\det(A) + \det(B) = 0$. Prove $\det(A + B) = 0$.

(98 逢甲應數)

解。

因為 $A(A + B) = A^2 + AB = I + AB$ 且 $(A + B)B = AB + B^2 = AB + I = I + AB$

所以 $A(A + B) = (A + B)B$, 因此 $\det(A(A + B)) = \det((A + B)B)$

$\Rightarrow \det(A)\det(A + B) = \det(A + B)\det(B)$

因為 $\det(A) + \det(B) = 0$, 所以 $\det(B) = -\det(A)$

因此 $\det(A)\det(A + B) = -\det(A + B)\det(A)$, 所以 $2\det(A)\det(A + B) = 0$

因為 $A^2 = I$, 所以 A 為可逆矩陣, 因此 $\det(A) \neq 0$, 所以 $\det(A + B) = 0$

9. If $P_n = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n} \end{bmatrix}$, then $\frac{\det(P_{n+1})}{\det(P_n)} =$ _____. (5%)

9.

$$= \begin{bmatrix} \frac{n-1}{2(n+1)} & \frac{n-1}{3(n+2)} & \cdots & \frac{n-1}{2n(n+1)} \\ \frac{n-2}{3(n+1)} & \frac{n-2}{4(n+2)} & \cdots & \frac{n-2}{2n(n+2)} \\ \vdots & \vdots & & \vdots \\ \frac{1}{n(n+1)} & \frac{1}{(n+1)(n+2)} & \cdots & \frac{1}{2n(2n-1)} \\ \frac{1}{n+1} & \frac{1}{n+2} & & \frac{1}{2n} \end{bmatrix} = \left(\prod_{i=1}^{n-1} i\right)\left(\prod_{j=1}^{n} \frac{1}{n+j}\right) \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+2} \\ \vdots & \vdots & & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{bmatrix}$$

$$= \frac{n!\cdot(n-1)!}{(2n)!} \begin{bmatrix} \frac{n-1}{2(n+1)} & \frac{n-2}{3(n+1)} & \cdots & \frac{1}{n(n+1)} & \frac{1}{n+1} \\ \frac{n-1}{3(n+2)} & \frac{n-2}{4(n+2)} & \cdots & \frac{1}{(n+1)(n+2)} & \frac{1}{n+2} \\ \vdots & \vdots & & \vdots & \vdots \\ \frac{n-1}{n(2n-1)} & \frac{n-2}{(n+1)(2n-1)} & \cdots & \frac{1}{(2n-2)(2n-1)} & \frac{1}{2n-1} \\ 0 & 0 & & 0 & \end{bmatrix}$$

$$= \frac{n!\cdot(n-1)!}{(2n)!}\left(\prod_{i=1}^{n-1} i\right)\left(\prod_{j=1}^{n-1} \frac{1}{n+j}\right) \begin{bmatrix} \frac{1}{1} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-2} \\ 0 & 0 & \cdots & 0 \end{bmatrix} = \frac{[n!]^2[(n+1)!]^2 \times 2n}{[2n)!]^2} \times P_{n-1}$$

15 Suppose that $f: G \to H$ is a group homomorphism and f is onto. Prove that if G is abelian, then H is abelian. (10%)

同態 (結構不變的 mapping)

if $x<50$ and $y<50$, then $x+y<100$

Let $*$ 為 G 的運算子, $\cdot$ 為 H 的運算子

$\because f$ is onto $\therefore \forall y_1, y_2 \in H, \exists x_1, x_2$,

s.t. $f(x_1)=y_1$, $f(x_2)=y_2$

$\to y_1 \cdot y_2 = f(x_1) \cdot f(x_2) = f(x_1 * x_2)$

$= f(x_2 * x_1) = f(x_2) \cdot f(x_1)$

$= y_2 \cdot y_1$

試題隨卷繳回

9. (10%) Let

$$A=\begin{bmatrix} 2 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix}.$$

Let

$$U=\left\{ a\begin{bmatrix}1\\0\\1\\0\end{bmatrix}+b\begin{bmatrix}0\\1\\0\\1\end{bmatrix}+c\begin{bmatrix}-1\\0\\1\\0\end{bmatrix} : a,b,c\in\mathbb{R} \right\}.$$

The numbers of elements $-2, -1, 0, 1, 2$ in a matrix B with

$$Bx=\begin{cases} Ax & \text{if } x\in U \\ 0 & \text{if } x\in U^{\perp}. \end{cases}$$

are D, D, 10, 4, 2, respectively.

$\dim U = 3 \to \dim U^{\perp} = 1$

$\to U^{\perp} = \left\{ \begin{bmatrix}0\\1\\0\\-1\end{bmatrix} \right\}$ (n)

$\alpha=1$, $\beta=0$, $\gamma=0$, $\delta=1$ $\leftarrow Bn=0$

$\because B-A=\begin{bmatrix} \alpha(0\ 1\ 0\ -1) \\ \beta(0\ 1\ 0\ -1) \\ \gamma(0\ 1\ 0\ -1) \\ \delta(0\ 1\ 0\ -1) \end{bmatrix} \to B=\begin{bmatrix} 2\alpha & 0 & 0 & 2\alpha \\ 0 & \beta & 0 & -\beta \\ 0 & \gamma & 0 & -\gamma \\ 2\delta & 0 & 0 & 2-\delta \end{bmatrix} \to n \in N(B)$

$\begin{cases} \forall x\in U \to (B-A)x=0 \to x\in N(B-A) \to N(B-A)=U \to \text{rank}(B-A)=1 \\ \forall x\in U^{\perp} \to Bx=0 \to x\in N(B) \to N(B)=U^{\perp} \to \text{rank}(B)=3 \end{cases}$

接次頁

$B[(B-A)=U^{\perp}$

3-122 Let V be a vector space, and A, B, and C be subspaces of V with $C \subset A$. Show that $A \cap (B+C) = (A \cap B) + C$. (98 交大電信)

解。

(a) $\forall v \in A \cap (B+C)$, 則 $v \in A$ 且 $v \in B+C$

$\Rightarrow v = x + y$, for some $x \in B, y \in C$

$\Rightarrow x = v - y$, 因為 $C \subset A$ 且 $y \in C$, 所以 $y \in A$

因為 $v \in A, y \in A$ 且 A 為 V 的 subspace, 所以 $x = v - y \in A$

因為 $x \in B$, 所以 $x \in A \cap B$, 因此 $v = x + y \in (A \cap B) + C$

以上證明了 $A \cap (B+C) \subseteq (A \cap B) + C$

(b) $\forall v \in (A \cap B) + C$, 則 $v = x + y$, for some $x \in A \cap B, y \in C$

因為 $x \in A \cap B$, 所以 $x \in A$

因為 $C \subset A$ 且 $y \in C$, 所以 $y \in A$

因為 A 為 V 的 subspace, 所以 $v = x + y \in A$

因為 $x \in A \cap B$, 所以 $x \in B$, 加上 $y \in C$, 所以 $v = x + y \in B + C$

因此 $v \in A \cap (B+C)$, 以上證明了 $(A \cap B) + C \subseteq A \cap (B+C)$

由(a), (b)得 $A \cap (B+C) = (A \cap B) + C$

3-146 A is an $n \times n$ matrix. Suppose $A^k x = 0$ has a vector solution α and $A^{k-1}\alpha \neq 0$, where k is an integer and x is a vector. Is $\alpha, A\alpha, \ldots, A^{k-1}\alpha$ linearly independent or not? (95 台科大電子)

解。

$\alpha, A\alpha, \ldots, A^{k-1}\alpha$為 linearly independent, 證明如下

假設 $a_0\alpha + a_1 A\alpha + \cdots + a_{k-1}A^{k-1}\alpha = 0$ —— (*)

二邊 apply A^{k-1} 得 $A^{k-1}(a_0\alpha + a_1 A\alpha + \cdots + a_{k-1}A^{k-1}\alpha) = A^{k-1} \cdot 0 = 0$

$\Rightarrow a_0 A^{k-1}\alpha + a_1 A^k \alpha + \cdots + a_{k-1}A^{2k-2}\alpha = 0$

$\Rightarrow a_0 A^{k-1}\alpha = 0$

$\Rightarrow a_0 = 0$, 代回(*)得 $a_1 A\alpha + \cdots + a_{k-1}A^{k-1}\alpha = 0$

利用同樣的方法, 二邊依序 apply $A^{k-2}, \ldots, A, I$, 可依序得 $a_1, \ldots, a_{k-2}, a_{k-1}$ 為 0,

因此$\alpha, A\alpha, \ldots, A^{k-1}\alpha$為 linearly independent

4-215 Let $A=\begin{bmatrix}1&0&-1&2&1\\-1&1&3&-1&0\\-2&1&4&-1&3\\3&-1&-5&1&-6\end{bmatrix}$.

(a) Suppose that B is a 5×5 matrix such that $AB=O$. Prove that $rank(B)\le 2$.

(b) Find a 5×5 matrix M with rank 2 such that $AM=O$.

(93 清大應數)(98 東華應數)

解。

(a) A 的 rref 為 $U=\begin{bmatrix}1&0&-1&0&-3\\0&1&2&0&-1\\0&0&0&1&2\\0&0&0&0&0\end{bmatrix}$，因為 U 具三個非零列，所以 $rank(A)=3$

$\Rightarrow nullity(A)=5-3=2$

因為 $AB=O$，所以 $AB\boldsymbol{x}=\boldsymbol{0}, \forall \boldsymbol{x}\in F^{5\times 1}$

$\Rightarrow CS(B)\subseteq \ker(A)$，其中 $CS(B)$表示 B 的 column space

$\Rightarrow \dim(CS(B))\le \dim(\ker(A))$

$\Rightarrow rank(B)\le nullity(A)=2$

(b) 先求 $\ker(A)$需要解 $A\boldsymbol{x}=\boldsymbol{0}$，它等價於解 $U\boldsymbol{x}=\boldsymbol{0}$，得 $\begin{cases}x_1-x_3-3x_5=0\\x_2+2x_3-x_5=0\\x_4+2x_5=0\end{cases}$

$$\Rightarrow \begin{cases}x_1=x_3+3x_5\\x_2=-2x_3+x_5\\x_4=-2x_5\end{cases}, \text{所以 } \ker(A)=span\left\{\boldsymbol{u}=\begin{bmatrix}1\\-2\\1\\0\\0\end{bmatrix}, \boldsymbol{v}=\begin{bmatrix}3\\1\\0\\-2\\1\end{bmatrix}\right\}$$

取 $M=[\boldsymbol{u}\quad \boldsymbol{v}\quad \boldsymbol{u}\quad \boldsymbol{u}\quad \boldsymbol{u}]=\begin{bmatrix}1&3&1&1&1\\-2&1&-2&-2&-2\\1&0&1&1&1\\0&-2&0&0&0\\0&1&0&0&0\end{bmatrix}$, $rank(M)=2$

因為 $A\boldsymbol{u}=A\boldsymbol{v}=\boldsymbol{0}$，所以 $AM=[A\boldsymbol{u}\quad A\boldsymbol{v}\quad A\boldsymbol{u}\quad A\boldsymbol{u}\quad A\boldsymbol{u}]=O$

4-238 Let A be a linear transformation on a finite-dimensional vector space V. Prove that $\dim(\ker(A+I)) + \dim(\ker(A-I)) = \dim(V)$ if and only if $A^2 = I$.

(91 交大應數)(98 成大應數)

$(\Rightarrow)$:

因為 $\dim(\ker(A+I)) + \dim(\ker(A-I)) = \dim(V)$

所以 $\dim(\ker(A+I)) = \dim(V) - \dim(\ker(A-I)) = rank(A-I) = \dim(CS(A-I))$

接著我們證明 $\ker(A+I) \subseteq CS(A-I)$

$\forall \boldsymbol{x} \in \ker(A+I)$, 則 $(A+I)\boldsymbol{x} = \boldsymbol{0}$, 所以 $A\boldsymbol{x} + \boldsymbol{x} = \boldsymbol{0}$, 因此 $A\boldsymbol{x} = -\boldsymbol{x}$

$\Rightarrow (A-I)\boldsymbol{x} = A\boldsymbol{x} - \boldsymbol{x} = -2\boldsymbol{x}$

$\Rightarrow \boldsymbol{x} = (A-I)(\frac{1}{2}\boldsymbol{x}) \in CS(A-I)$, 所以 $\ker(A+I) \subseteq CS(A-I)$

因為 $\dim(\ker(A+I)) = \dim(CS(A-I))$, 所以 $\ker(A+I) = CS(A-I)$

$\Rightarrow (A+I)(A-I) = O$, 所以 $A^2 - I = O$, 因此 $A^2 = I$

$(\Leftarrow)$:

因為 $A^2 = I$, 所以 $A^2 - I = O$, 因此 $(A+I)(A-I) = O$

$\Rightarrow CS(A-I) \subseteq \ker(A+I)$

根據$(\Rightarrow)$的證明知 $\ker(A+I) \subseteq CS(A-I)$, 所以 $\ker(A+I) = CS(A-I)$

$\Rightarrow \dim(\ker(A+I)) = \dim(CS(A-I)) = rank(A-I) = \dim(V) - \dim(\ker(A-I))$

$\Rightarrow \dim(\ker(A+I)) + \dim(\ker(A-I)) = \dim(V)$

範例 7

Suppose $A \in \boldsymbol{R}^{m\times n}$. If the columns of A all add up to a fixed constant c, Show that c is an eigenvalues of A.

(93 清大資應)

解

假設 $A = [\boldsymbol{a}_1 \quad \boldsymbol{a}_2 \quad \cdots \quad \boldsymbol{a}_n]$, 其中 $\boldsymbol{a}_j$ 表示 A 的第 j 行, $j = 1, 2, \ldots, n$

取 $\boldsymbol{u} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$, 因為 $\boldsymbol{a}_j$ 的元素和為 c, 所以 $\boldsymbol{a}_j^T \boldsymbol{u} = c$, $\forall j = 1, 2, \ldots, n$

$$\Rightarrow A^T \boldsymbol{u} = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_n^T \end{bmatrix} \boldsymbol{u} = \begin{bmatrix} \boldsymbol{a}_1^T \boldsymbol{u} \\ \boldsymbol{a}_2^T \boldsymbol{u} \\ \vdots \\ \boldsymbol{a}_n^T \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} = c\boldsymbol{u}$$, 所以 c 為 A^T 的 eigenvalue

因為 A 與 A^T 具相同的 eigenvalue, 所以 c 為 A 的 eigenvalue

8-128 (a) Find a 3×3 orthogonal matrix U that maps the x-y plane $z = 0$ to the plane $P : x + y + z = 0$.

(b) Use U as the matrix of change of coordinates to deduce the formula for the rotation around the axis $L = \{(t, t, t) \mid t \in \boldsymbol{R}\}$ with rotation angle 90° (counterclockwise).

(99 彰師大數學)

解：

(a) 令 W 表示 xy-平面，則 $(0, 0, 1)^T$ 為 W 的單位法向量，且 $\{(1, 0, 0)^T, (0, 1, 0)^T\}$ 為 W 的一組 orthonormal basis，另外，$\boldsymbol{u} = (\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})^T$ 為 P 的單位法向量且 $\{\boldsymbol{v}_1 = (-1, 1, 0)^T, \boldsymbol{v}_2 = (-1, 0, 1)^T\}$ 為 P 的一組 basis，利用 Gram-Schmidt process 對 $\boldsymbol{v}_1, \boldsymbol{v}_2$ 正交化得 $\boldsymbol{u}_1 = (-1, 1, 0)^T$，$\boldsymbol{u}_2 = (\frac{-1}{2}, \frac{-1}{2}, 1)^T$

令 $\boldsymbol{w}_1 = \dfrac{\boldsymbol{u}_1}{\| \boldsymbol{u}_1 \|} = (\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)^T, \boldsymbol{w}_2 = \dfrac{\boldsymbol{u}_2}{\| \boldsymbol{u}_2 \|} = (\frac{-1}{\sqrt{6}}, \frac{-1}{\sqrt{6}}, \frac{2}{\sqrt{6}})^T$，則 $\{\boldsymbol{w}_1, \boldsymbol{w}_2\}$ 為 W 的一組 orthonormal basis，因為 U 將 W 對應到 P，所以 $U(0, 0, 1)^T = \boldsymbol{u}$, $U(1, 0, 0)^T = \boldsymbol{w}_1$, $U(0, 1, 0)^T = \boldsymbol{w}_2$，所以 $U = \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{bmatrix}$

(b) 假設 T 為對 L 逆時針旋轉 90°的 linear transformation，則 $L(\boldsymbol{u}) = \boldsymbol{u}$

因為 $P = L^{\perp}$ 且 $\det(U) = 1$，所以 T 相當於在 P 平面上逆時針旋轉 90°

取 $\beta = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{u}\}$ 為 $\boldsymbol{R}^3$ 的 orthonormal basis，

則 $[T]_{\beta} = \begin{bmatrix} \cos 90° & -\sin 90° & 0 \\ \sin 90° & \cos 90° & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

取 $\gamma = \{(1, 0, 0)^T, (0, 1, 0)^T, (0, 0, 1)^T\}$ 為 $\boldsymbol{R}^3$ 的 standard basis

則 $[T]_{\gamma} = [I]_{\beta}^{\gamma} [T]_{\beta} [I]_{\gamma}^{\beta} = U[T]_{\beta} U^{-1} = U[T]_{\beta} U^T$

範例 5

If $\begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 1 \\ 2 & 3 & 2 & 2 & 2 & 2 \\ 3 & 3 & 4 & 3 & 3 & 3 \\ 4 & 4 & 4 & 5 & 4 & 4 \\ 5 & 5 & 5 & 5 & 6 & 5 \\ 6 & 6 & 6 & 6 & 6 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$, find $x_1 + x_2 + x_3 + x_4 + x_5 + x_6$. (99 台大資工)

解

令 $A = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 1 \\ 2 & 3 & 2 & 2 & 2 & 2 \\ 3 & 3 & 4 & 3 & 3 & 3 \\ 4 & 4 & 4 & 5 & 4 & 4 \\ 5 & 5 & 5 & 5 & 6 & 5 \\ 6 & 6 & 6 & 6 & 6 & 7 \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}, B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 & 6 \end{bmatrix} = A - I$

因為 $\det(B) = \det(B - 0I) = 0$, 所以 0 為 B 的 eigenvalue

因為 $nullity(B - 0I) = 6 - rank(B) = 6 - 1 = 5$, 所以 eigenvalue 0 的代數重數至少為 5

假設λ為 B 的另一個 eigenvalue, 因為 $tr(B)$為 B 的所有 eigenvalue 的和

所以 $5 \cdot 0 + \lambda = tr(B) = 1 + 2 + \cdots + 6 = 21$, 因此$\lambda = 21$

得 B 的所有 eigenvalue 為 0, 0, 0, 0, 0, 21

因為 $A = B + I$, 所以 A 的所有 eigenvalue 為 1, 1, 1, 1, 1, 22

因為 $\det(A)$為 A 的所有 eigenvalue 乘積, 所以 $\det(A) = 22$

因為 $\det(A)\boldsymbol{x} = \boldsymbol{b}$, 所以 $\boldsymbol{x} = \dfrac{1}{\det(A)}\boldsymbol{b} = \dfrac{1}{22}\boldsymbol{b}$

因此 $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = \dfrac{1}{22}(1+2+3+4+5+6) = \dfrac{21}{22}$

例 52

Show that if $T : V \to V$ is a linear transformation such that $rank(T) = rank(T^2)$, then $V = \ker(T) \oplus Im(T)$. (88 交大應數)(91 成大應數)

解

根據 5–28, 只需證明 $\ker(T) \cap Im(T) = \{\mathbf{0}\}$

因為 $rank(T) = rank(T^2)$, 所以 $\dim(V) - nullity(T) = \dim(V) - nullity(T^2)$

$\Rightarrow nullity(T) = nullity(T^2)$, 即 $\dim(\ker(T)) = \dim(\ker(T^2))$

因為 $\ker(T)$為 $\ker(T^2)$的 subspace, 所以 $\ker(T) = \ker(T^2)$

$\forall \boldsymbol{v} \in \ker(T) \cap Im(T)$, 則 $\boldsymbol{v} \in \ker(T)$且 $\boldsymbol{v} \in Im(T)$

$\Rightarrow T(\boldsymbol{v}) = \mathbf{0}$ 且存在 $\boldsymbol{u} \in V$ 使得 $T(\boldsymbol{u}) = \boldsymbol{v}$

$\Rightarrow \mathbf{0} = T(\boldsymbol{v}) = T(T(\boldsymbol{u})) = T^2(\boldsymbol{u})$, 所以 $\boldsymbol{u} \in \ker(T^2)$

因為 $\ker(T) = \ker(T^2)$, 所以 $\boldsymbol{u} \in \ker(T)$, 因此 $\boldsymbol{v} = T(\boldsymbol{u}) = \mathbf{0}$

這證明了 $\ker(T) \cap Im(T) = \{\mathbf{0}\}$

∵ $rank(T) = rank(T^2)$

∴ $\ker(T) = \ker(T^2)$

$\forall v = \ker(T) \cap Im(T)$

有 $T(v) = 0$ 且 $\exists u \in V$, $T(u) = v$

$\Rightarrow 0 = T(v) = T^2(u)$

∴ $u \in \ker(T^2)$

又 $\ker(T) = \ker(T^2)$,

$\Rightarrow u \in \ker(T)$ ∴ $v = T(u) = 0$

範例 1

Suppose that A and B are both symmetric positive definite matrices. Then all eigenvalues of AB are positive. (91 靜宜應數)(98 嘉大應數)

解

假設λ為 AB 的 eigenvalue, 則存在 $\boldsymbol{x} \neq \mathbf{0}$ 使得 $AB\boldsymbol{x} = \lambda\boldsymbol{x}$

因為 B 為 symmetric, 所以 $B^T = B$

$\Rightarrow B^TAB\boldsymbol{x} = B^T(\lambda\boldsymbol{x}) = \lambda B^T\boldsymbol{x} = \lambda B\boldsymbol{x}$

$\Rightarrow \boldsymbol{x}^TB^TAB\boldsymbol{x} = \boldsymbol{x}^T(\lambda B\boldsymbol{x}) = \lambda(\boldsymbol{x}^TB\boldsymbol{x})$

因為 B 為 positive definite, 所以 $\boldsymbol{x}^TB\boldsymbol{x} > 0$, 因此 $\lambda = \dfrac{\boldsymbol{x}^TB^TAB\boldsymbol{x}}{\boldsymbol{x}^TB\boldsymbol{x}} = \dfrac{(B\boldsymbol{x})^TA(B\boldsymbol{x})}{\boldsymbol{x}^TB\boldsymbol{x}}$

因為 B 為 positive definite, 所以 B 為 nonsingular, 因此 $B\boldsymbol{x} \neq \mathbf{0}$

因為 A 為 positive definite, 所以 $(B\boldsymbol{x})^TA(B\boldsymbol{x}) > 0$

$\Rightarrow \lambda = \dfrac{(B\boldsymbol{x})^TA(B\boldsymbol{x})}{\boldsymbol{x}^TB\boldsymbol{x}} > 0$, 因此 AB 的所有 eigenvalue 皆為正

範例 4

Let $C[-1, 1]$ be the vector space of continuous real-valued functions on $[-1, 1]$ with the following inner product:

$$< f, g >= \int_{-1}^{1} f(x)g(x)dx .$$

Let A be the span of subspace generated by $1, x, x^2$.

(a) Find an orthonormal basis for A.

(b) Let $T : A \to A$ be the linear transformation defined by

$$T(a_0 + a_1x + a_2x^2) = a_2 + a_1x + a_0x^2.$$

Determine the linear transformation $T^* : A \to A$ such that $<T(f), g> = <f, T^*(g)>$ for all $f, g \in A$. (98 交大應數)

解

令 $w_1(x) = \dfrac{h_1(x)}{\| h_1(x) \|} = \dfrac{1}{\sqrt{2}}$, $w_2(x) = \dfrac{h_2(x)}{\| h_2(x) \|} = \sqrt{\dfrac{3}{2}}x$, $w_3(x) = \dfrac{h_3(x)}{\| h_3(x) \|} = \dfrac{\sqrt{45}}{\sqrt{8}}(x^2 - \dfrac{1}{3})$

則 $\beta = \{w_1(x), w_2(x), w_3(x)\}$ 為 A 的一組 orthonormal basis

(b) 假設 $\gamma = \{1, x, x^2\}$, $T(1) = x^2$, $T(x) = x$, $T(x^2) = 1$, 所以 $D = [T]_\gamma = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

$[T]_\gamma \to [T]_\beta \to [T^*]_\beta \to [T^*]_\gamma$

γ 不是單範正交基底，不能直接轉換

$$\Rightarrow B = [T]_\beta = [I]_\gamma^\beta [T]_\gamma [I]_\beta^\gamma = ([I]_\beta^\gamma)^{-1}[T]_\gamma[I]_\beta^\gamma = P^{-1}DP, \text{ 其中 } P = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{-\sqrt{5}}{\sqrt{8}} \\ 0 & \sqrt{\frac{3}{2}} & 0 \\ 0 & 0 & \frac{\sqrt{45}}{\sqrt{8}} \end{bmatrix}$$

$$\Rightarrow [T^*]_\beta = B^H = (P^{-1}DP)^H = P^H D^H (P^{-1})^H = P^H D (P^H)^{-1}$$

$$\Rightarrow [T^*]_\gamma = [I]_\beta^\gamma [T^*]_\beta [I]_\gamma^\beta = P[T^*]_\beta P^{-1} = PP^H D(P^H)^{-1}P^{-1} = (PP^H)D(PP^H)^{-1}$$

因為 $PP^H = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{-\sqrt{5}}{\sqrt{8}} \\ 0 & \sqrt{\frac{3}{2}} & 0 \\ 0 & 0 & \frac{\sqrt{45}}{\sqrt{8}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \sqrt{\frac{3}{2}} & 0 \\ \frac{-\sqrt{5}}{\sqrt{8}} & 0 & \frac{\sqrt{45}}{\sqrt{8}} \end{bmatrix} = \begin{bmatrix} \frac{9}{8} & 0 & \frac{-15}{8} \\ 0 & \frac{3}{2} & 0 \\ \frac{-15}{8} & 0 & \frac{45}{8} \end{bmatrix}$

$$\Rightarrow [T^*]_\gamma = (PP^H)D(PP^H)^{-1} = \begin{bmatrix} \frac{9}{8} & 0 & \frac{-15}{8} \\ 0 & \frac{3}{2} & 0 \\ \frac{-15}{8} & 0 & \frac{45}{8} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 & \frac{2}{3} \\ 0 & \frac{2}{3} & 0 \\ \frac{2}{3} & 0 & \frac{2}{5} \end{bmatrix} = \begin{bmatrix} -3 & 0 & \frac{-4}{5} \\ 0 & 1 & 0 \\ 10 & 0 & 3 \end{bmatrix}$$

$$\Rightarrow [T^*(a_0 + a_1x + a_2x^2)]_\gamma = [T^*]_\gamma [a_0 + a_1x + a_2x^2]_\gamma = \begin{bmatrix} -3 & 0 & \frac{4}{5} \\ 0 & 1 & 0 \\ 10 & 0 & 3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -3a_0 + \frac{4}{5}a_2 \\ a_1 \\ 10a_0 + 3a_2 \end{bmatrix}$$

$$\Rightarrow T^*(a_0 + a_1x + a_2x^2) = (-3a_0 + \frac{4}{5}a_2) + a_1x + (10a_0 + 3a_2)x^2$$

例 24

假設 $A=\begin{bmatrix}0.8 & 0.2 & 0.1\\ 0.1 & 0.7 & 0.3\\ 0.1 & 0.1 & 0.6\end{bmatrix}$，求一正交矩陣 P 使得 $P^TAP=R$ 為上三角矩陣

解

$p_A(x)=-(x-0.6)(x-1)(x-0.5)$，得 A 的特徵根為 0.6, 1, 0.5

取 $\lambda_1=0.6$，$V(\lambda_1)=\ker(A-\lambda_1 I)=span\left\{\begin{bmatrix}1\\-1\\0\end{bmatrix}\right\}$，令 $\boldsymbol{w}_1=\begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{-1}{\sqrt{2}}\\ 0\end{bmatrix}$ 為 A 相對於 λ_1 的一個單位特徵向量，求 $\boldsymbol{w}_2, \boldsymbol{w}_3$ 使得 $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3\}$ 為 $\boldsymbol{R}^3$ 的單範正交基底，令 $W=span\{\boldsymbol{w}_1\}$，求 $W^{\perp}$ 的一組單範正交基底 $\{\boldsymbol{w}_2, \boldsymbol{w}_3\}$，其中 $\boldsymbol{w}_2=\begin{bmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\\ 0\end{bmatrix}$，$\boldsymbol{w}_3=\begin{bmatrix}0\\0\\1\end{bmatrix}$

令 $W=\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0\\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0\\ 0 & 0 & 1\end{bmatrix}$，則 $W^TAW=\left[\begin{array}{c|cc}0.6 & 0.1 & -0.1\sqrt{2}\\ \hline 0 & 0.9 & 0.2\sqrt{2}\\ 0 & 0.1\sqrt{2} & 0.6\end{array}\right]$ → 仍不為上三角矩陣

令 $M=\begin{bmatrix}0.9 & 0.2\sqrt{2}\\ 0.1\sqrt{2} & 0.6\end{bmatrix}$，則 M 的特徵根為 1, 0.5

取 $\lambda_2=1$，$\ker(M-\lambda_2 I)=span\left\{\begin{bmatrix}2\sqrt{2}\\1\end{bmatrix}\right\}$，令 $\boldsymbol{w}_1'=\begin{bmatrix}\frac{2\sqrt{2}}{3}\\ \frac{1}{3}\end{bmatrix}$ 為 M 相對於 λ_2 的一個單位特徵向量，求 $\boldsymbol{w}_2'$ 使得 $\{\boldsymbol{w}_1', \boldsymbol{w}_2'\}$ 為 $\boldsymbol{R}^2$ 的單範正交基底，取 $\boldsymbol{w}_2'=\begin{bmatrix}\frac{1}{3}\\ \frac{-2\sqrt{2}}{3}\end{bmatrix}$

令 $W'=\begin{bmatrix}\frac{2\sqrt{2}}{3} & \frac{1}{3}\\ \frac{1}{3} & \frac{-2\sqrt{2}}{3}\end{bmatrix}$ → 是 λ_2

取 $P=W\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & & & \\ \vdots & & W' & \\ 0 & & & \end{bmatrix}=\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0\\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & \frac{2\sqrt{2}}{3} & \frac{1}{3}\\ 0 & \frac{1}{3} & \frac{-2\sqrt{2}}{3}\end{bmatrix}=\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{2}{3} & \frac{\sqrt{2}}{6}\\ \frac{1}{\sqrt{2}} & \frac{2}{3} & \frac{\sqrt{2}}{6}\\ 0 & \frac{1}{3} & \frac{-2\sqrt{2}}{3}\end{bmatrix}$

則 $P^TAP=\begin{bmatrix}0.6 & \frac{\sqrt{2}}{30} & \frac{1}{6}\\ 0 & 1 & \frac{\sqrt{2}}{10}\\ 0 & 0 & 0.5\end{bmatrix}=R$ 為上三角矩陣

範例 11

Let U and V be two $m \times m$ positive definite matrices.

(a) Find a $m \times 1$ complex vector $\boldsymbol{b}$, such that

$$Q = \frac{\boldsymbol{b}^H U \boldsymbol{b}}{\boldsymbol{b}^H V \boldsymbol{b}}$$

is maximized.

(b) What is the maximum value of Q in (a)? (99 中山通訊)

解

(a) 因為 V 為 positive definite matrix, 所以存在 B 為可逆矩陣使得 $V = B^H B$

$$\Rightarrow Q = \frac{\boldsymbol{b}^H U \boldsymbol{b}}{\boldsymbol{b}^H B^H B \boldsymbol{b}} = \frac{\boldsymbol{b}^H U \boldsymbol{b}}{(B\boldsymbol{b})^H (B\boldsymbol{b})}, \text{ 令 } \boldsymbol{x} = B\boldsymbol{b} \text{ 或 } \boldsymbol{b} = B^{-1}\boldsymbol{x}$$

$$\text{則 } Q = \frac{(B^{-1}\boldsymbol{x})^H U (B^{-1}\boldsymbol{x})}{\boldsymbol{x}^H \boldsymbol{x}} = \frac{\boldsymbol{x}^H (B^{-1})^H U B^{-1} \boldsymbol{x}}{\boldsymbol{x}^H \boldsymbol{x}} = \frac{\boldsymbol{x}^H (B^H)^{-1} U B^{-1} \boldsymbol{x}}{\boldsymbol{x}^H \boldsymbol{x}} = \frac{\boldsymbol{x}^H A \boldsymbol{x}}{\boldsymbol{x}^H \boldsymbol{x}},$$

其中 $A = (B^H)^{-1} U B^{-1}$, 根據 Rayleigh principle, Q 的最大值為 A 的最大 eigenvalue, 假設 λ 為 A 的最大 eigenvalue, 當 $\boldsymbol{x}$ 為 A 相對於 λ 的 eigenvector 時, Q 為最大值

因為 $A\boldsymbol{x} = \lambda \boldsymbol{x}$, 所以 $(B^H)^{-1} U B^{-1} \boldsymbol{x} = \lambda \boldsymbol{x}$

$$\Rightarrow (B^H)^{-1} U B^{-1} B \boldsymbol{b} = \lambda B \boldsymbol{b}$$

$$\Rightarrow (B^H)^{-1} U \boldsymbol{b} = \lambda B \boldsymbol{b}$$

$$\Rightarrow B^{-1} (B^H)^{-1} U \boldsymbol{b} = \lambda \boldsymbol{b}$$

$\Rightarrow (B^H B)^{-1} U \boldsymbol{b} = \lambda \boldsymbol{b}$, 即 $V^{-1} U \boldsymbol{b} = \lambda \boldsymbol{b}$, 因此取 $\boldsymbol{b}$ 為 $V^{-1}U$ 相對於 λ 的 eigenvector 使得 Q 為最大

(b) 同(a)的說明, Q 的最大值為 $A = (B^H)^{-1} U B^{-1}$ 的最大 eigenvalue, 即 $V^{-1}U$ 的最大 eigenvalue

例 62

假設 $\boldsymbol{x}=\begin{bmatrix}3\\4\\4\\2\end{bmatrix}$，求一個 Householder 矩陣 H 使得 $H\boldsymbol{x}$ 中後面 2 項為 0

解

根據例 61，取 $H'=\frac{1}{3}\begin{bmatrix}2&2&1\\2&-1&-2\\1&-2&2\end{bmatrix}$，則 $H'\begin{bmatrix}4\\4\\2\end{bmatrix}=\begin{bmatrix}6\\0\\0\end{bmatrix}$

令 $H=\begin{bmatrix}I_1&O\\O&H'\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&\frac{2}{3}&\frac{2}{3}&\frac{1}{3}\\0&\frac{2}{3}&\frac{-1}{3}&\frac{-2}{3}\\0&\frac{1}{3}&\frac{-2}{3}&\frac{2}{3}\end{bmatrix}$，則 $H\begin{bmatrix}3\\4\\4\\2\end{bmatrix}=\begin{bmatrix}3\\6\\0\\0\end{bmatrix}$

$y=\begin{bmatrix}4\\4\\2\end{bmatrix}\rightarrow \|y\|=6 \rightarrow u=\frac{y-6e_1}{\|y-6e_1\|}=\frac{1}{\sqrt{24}}\begin{bmatrix}-2\\4\\2\end{bmatrix}$

$\rightarrow H=I-2uu^T=\frac{1}{3}\begin{bmatrix}2&2&1\\2&-1&-2\\1&-2&2\end{bmatrix}\rightarrow Hy=6e_1$

2-107 If $F(x)=\det(A)=\det\begin{bmatrix} x & \frac{x^2}{2} & \frac{x^3}{3} & \frac{x^4}{4} & \frac{x^5}{5} \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \end{bmatrix}$. Show that $F'(x)=c(x^2-3x+2)(x^2+ax+b)$ for some choice of constants a, b, c and give the values of the three constants. (90 師大資工)

$$F(x)=x\,cof(a_{11})-\frac{x^2}{2}cof(a_{12})+\frac{x^3}{3}cof(a_{13})-\frac{x^4}{4}cof(a_{14})+\frac{x^5}{5}cof(a_{15})$$

因為 $cof(a_{11}),\ldots,cof(a_{15})$皆為常數

所以 $F'(x)=cof(a_{11})-x\,cof(a_{12})+\ x^2cof(a_{13})-x^3cof(a_{14})+x^4cof(a_{15})$

$$=\det\begin{bmatrix} 1 & x & x^2 & x^3 & x^4 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \end{bmatrix}$$

這個矩陣為一個 Vandermonde matrix

因此 $F'(x)=(1-x)(2-x)(3-x)(4-x)(2-1)(3-1)(4-1)(3-2)(4-2)(4-3)$

$=12(1-x)(2-x)(3-x)(4-x)=12(x^2-3x+2)(x^2-7x+12)$

所以 $c=12,\ a=-7,\ b=12$

Trees				
Tree	Insert x	Delete x	Search x	Remark
BST	$O(\log n) \sim O(n)$			Create: $O(n \log n) \sim O(n^2)$
AVL tree	$O(\log_m n)$			$F_{h+2} - 1 \leq n \leq 2^h - 1$
B tree				$1 + 2\frac{\lceil \frac{m}{2} \rceil^{h-1} - 1}{\lceil \frac{m}{2} \rceil - 1} \leq n \leq \frac{m^h - 1}{m-1}$
RBT				$h \leq 2\log(n+1)$
Splay tree				Worst: $O(n)$, Amortized: $O(\log n)$

Priority queues					
Operations	Max (Min)	Min-max & Deap & SMMH	Leftist	Binomial	Fibonacci
Insert x	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n), O(1)^*$	$O(1)^*$
Delete max	$O(\log n)$	$O(\log n)$			
Delete min	$O(n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)^*$
Delete x				$O(\log n)$	$O(\log n)^*$
Merge	$O(n)$		$O(\log n)$	$O(\log n)$	$O(1)^*$
Decrease key				$O(\log n)$	$O(1)^*$
Search x	$O(n)$				
Find max	$O(1)$	$O(1)$			
Find min		$O(1)$		$O(\log n)$	$O(1)$
Remark			$shortest(\text{root}) \leq \log(n+1) - 1$		

Sorting algorithms					
Method	Time complexity			Space complexity	Stable
	Best	Worst	Average		
Insertion	$O(n)$	$O(n^2)$		$O(1)$	$\surd$
Selection	$O(n^2)$			$O(1)$	$\times$
Bubble	$O(n)$	$O(n^2)$		$O(1)$	$\surd$
Shell	$O(n^{1.5})$	$O(n^2)$		$O(1)$	$\times$
Quick	$O(n \log n)$	$O(n^2)$	$O(n \log n)$	$O(\log n) \sim O(n)$	$\times$
Merge	$O(n \log n)$			$O(n)$	$\surd$
Heap	$O(n \log n)$			$O(1)$	$\times$
LSD Radix	$O(n \times k)$			$O(n+k)$	$\surd$
Bucket/MSD Radix	$O(n)$	$O(n^2)$	$O(n+k)$	$O(n \times k)$	$\surd$
Counting	$O(n+k)$				$\surd$

Dynamic Programming algorithms		
Problem	Time complexity	Space complexity
Making change	$O(kn)$	$O(n)$
Fractional Knapsack problem	$\Theta(n \log n)$	$O(n)$
0/1 Knapsack problem (DP)	$O(n2^{\log W})$	$O(n2^{\log W})$
0/1 Knapsack problem (Branch-and-Bound)	$O(2^n)$	
Longest Common Subsequence (LCS)	$O(mn)$	$O(mn)$
Longest Increasing Subsequence (LIS)	$O(n^2)$	$O(n^2)$
Longest Common Substring	$O(mn)$	$O(mn)$
Minimum Edit Distance	$O(mn)$	$O(mn)$
Matrix-chain Multiplication	$O(n^3)$	$O(n^2)$
Traveling Salesperson problem	$\Theta(n^2 2^n)$	$O(n2^n)$
Optimal Binary Search Tree (OBST)	$\Theta(n^3)$	$\Theta(n^2)$

Graph algorithms		
Problem	Time complexity	Remark
Depth-First Search (DFS)	$O(\|V\| + \|E\|)$	
Kosaraju's	$O(\|V\| + \|E\|)$	
Kruskal's	$O(\|E\| \log \|V\|)$	
Prim's (Adjacency matrix)	$O(\|V\|^2)$	
Prim's (Adjacency list)	$O(\|V\|\|E\|)$	
Prim's (Min-Heap, Adjacency list)	$O(\|E\| \log \|V\|)$	
Prim's (Fibonacci heap, Adjacency list)	$O(\|E\| + \|V\| \log \|V\|)$	
Sollin's (Borůvka's)	$O(\|E\| \log \|V\|)$	
Dijkstra's (Min-heap)	$\Theta((\|E\| + \|V\|) \log \|V\|)$	Greedy, no negative edges or cycles
Dijkstra's (Fibonacci-heap)	$\Theta(\|E\| + \|V\| \log \|V\|)$	
Bellman-Ford	$O(\|V\|\|E\|)$	DP
Floyd-Warshall	$\Theta(\|V\|^3)$	DP, no negative cycles
Johnson's	$\Theta(\|V\|\|E\| + \|V\|^2 \log \|V\|)$	No negative cycles
Ford-Fulkerson	$O(\|E\|\|f^*\|)$	Greedy，f^* 為最大流
Edmond-Karp	$O(\|V\|\|E\|^2)$	
Push-relabel	$O(\|V\|^2\|E\|)$	

- Matrix-chain Multiplication：

$$m[i,j] = \begin{cases} 0 & ,i = j \\ \min_{i \le k \le j-1}\{m[i,k] + m[k+1,j] + p_{i-1}p_k p_j\} & ,i < j \end{cases} \tag{51}$$

- Optimal Binary Search Tree (OBST)：

$$e[i,j] = \begin{cases} q_{i-1} & ,j = i-1 \\ \min_{i \le r \le j}\{e[i,r-1] + e[r+1,j] + w[i,j]\} & ,i \le j \end{cases} \tag{52}$$
$$w[i,j] = w[i,j-1] + p_j + q_j$$

- Minimum vertex cover (tree)：

$$V(v) = \min\{1 + \text{sum}\{V(c),\ \forall c \in v.child\}, \\ \text{Length}\{v.child\} + \text{sum}\{V(g),\ \forall c \in v.child\ \forall g \in c.child\}\} \tag{53}$$

  First part: root is in the cover; second part: root is NOT in the cover.

- Max-cut：
  - NPC。
  - 若所有邊權重皆負，則可乘上 -1，變為 Min-cut。
  - 若為平面圖，可轉換為 Chinese Postman Problem（若為無向圖，即 Euler circuit，若為有向圖，則為 NPC）。

- 如果可以證明 **lower bound** of **worst case** of NPC problems is polynomial，則 $P = NP$。

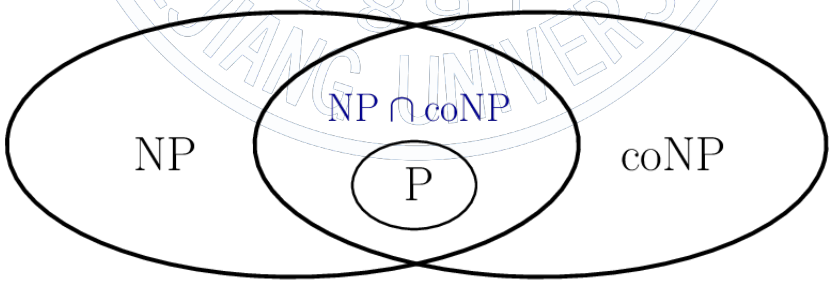


Figure 1: Relationship between NP and CO-NP.

- Permutation：

```
1: function PERM(list, i, n)
2:     if i == n then
3:         PRINT(list)
4:     else
5:         for j := i to n do
6:             SWAP(list, i, j)
7:             PERM(list, i + 1, n)
8:             SWAP(list, i, j)
9:         end for
10:    end if
11: end function
```

- 節點數：

$$n = (\sum_{i=1}^{\deg} i \times n_i) + 1$$
$$n_0 = n_2 + 1\text{（二叉樹）} \tag{54}$$

-

```
1: function CREATEMINHEAP(Tree s, size n)
2:     for i := n/2 to 1 do                    ▷ Start from parent of the last node.
3:         tmp := s[i]
4:         j := 2 × i                          ▷ Left child of i.
5:         while j ≤ n do                      ▷ There is a child.
6:             if j < n then                   ▷ Right child exists.
7:                 if s[j] > s[j + 1] then     ▷ Choose the smaller child.
8:                     j := j + 1
9:                 end if
10:            end if
11:            if tmp ≤ s[j] then
12:                Break.
13:            else                            ▷ Percolate one level.
14:                s[j/2] := s[j]
15:                j := j × 2
16:            end if
17:        end while
18:        s[j/2] := tmp
19:    end for
20: end function
```

- 尋找 articulation point：若 root 有 ≥ 2 子節點，則 root 為 articulation point；$\exists$ 非 root 節點 u，若 $\exists\, v$ 為 u 子節點，且 $low(v) \geq dfn(u)$，則 u 為 articulation point。

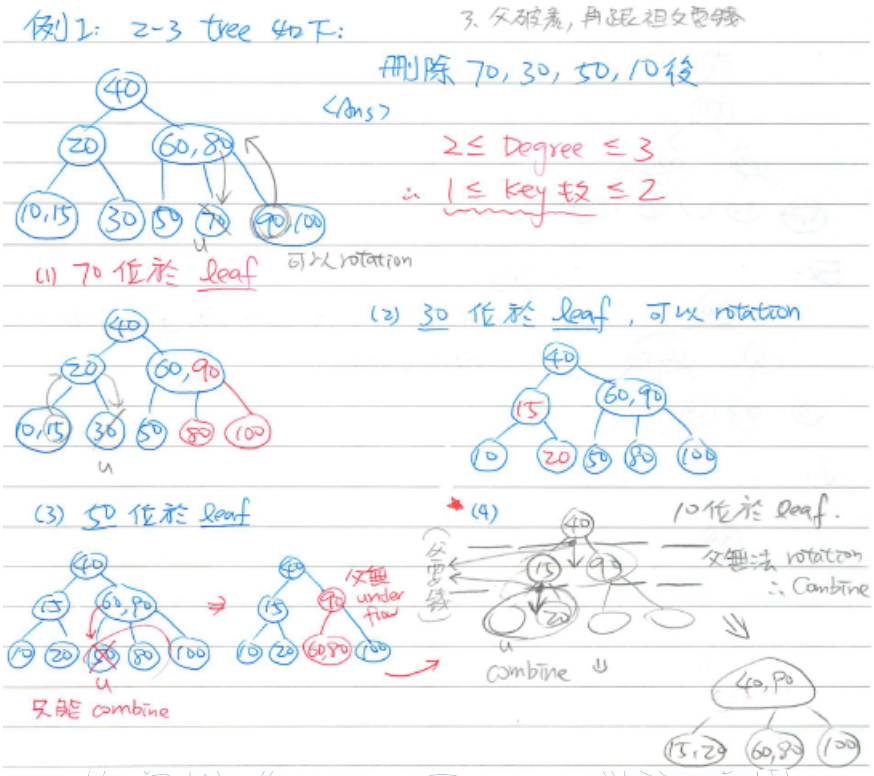


Figure 2: Example of B-tree deletion.

- (**FALSE**) For two functions $f(n)$ and $g(n)$, either $f(n) = O(g(n))$ or $f(n) = \Omega((f(n))$. Counterexample:

$$f(n) = \begin{cases} 1, & \text{if } n = 2k \\ 0, & \text{if } n = 2k+1 \end{cases}$$
$$g(n) = \begin{cases} 0, & \text{if } n = 2k \\ 1, & \text{if } n = 2k+1 \end{cases} \tag{55}$$

- For any uniform cost RAM program $T(n) = \Omega(S(n))$, where $S(n)$ is the space an algorithm uses for an input of size n.
- The capacity of each edge of a flow network can be floating-point, and it can be solved by linear programming.
- A flow network of multiple sources can be reduced to a single source.
- (**FALSE**) The value of any flow of a flow network is bounded by the capacity of only at most $O(n)$ cuts.

- (109NYCU-2) 2-coloring: $O(n^2)$, 3-coloring, 4-coloring: superpolynomial.
- Weighted-union heuristic: Append the **smaller** list onto the **longer** list, with ties broken arbitrarily.
- $n! \neq \Theta(n^n)$.
- A DAG with n vertices can **NOT** have more than $\binom{n}{2}$ edges.
- Longest palindrome subsequence:

$$L(i,j) = \begin{cases} 0 & ,i = j+1 \\ 1 & ,i = j \\ L(i+1,j-1)+2 & ,i < j \wedge s[i] = s[j] \\ \max(L(i+1,j), L(j,j-1)) & ,\text{otherwise} \end{cases} \tag{56}$$

where $L[1 \cdots n][1 \cdots n], s[1 \cdots n]$

- (102NTU-4) Minimum triangulation:

$$c(i,j) = \begin{cases} 0 & ,j < i+2 \\ \min\limits_{i<k<j} \{c(i,k) + c(k,j) + dist(i,j) + dis(j,k) + dist(k,j)\} & ,\text{otherwise} \end{cases} \tag{57}$$

```
double triangulation(Point P[], int n) {
    if (n < 3)
        return 0;

    double c[n][n];
    for (int gap = 0, gap < n; gap++) {
        for(int i = 0, j = gap; j < n; i++, j++) {
            if (j < i + 2)
                c[i][j] = 0.0;
            else {
                c[i][j] = MAX;
                for (int k = i + 1; k < j; k++) {
                    double val = c[i][k] + c[k][j] + wt(P, i, j,
                        k);
                    if (c[i][j] > val)
                        c[i][j] = val;
```

```
                    }
                }
            }
        }

        return c[0][n - 1];
    }
```

Listing 1: Minimum triangulation.

- Sort n integers ranged from 0 to $n^2 - 1$: 將 n 個整數表示成 **n 進位**數，每個數由 2-digit 表示，範圍 0 到 $n - 1$，再用 radix sort 對 2-digit 排序，共兩次。
- If max frequency is ≤ 2 times of min frequency, Huffman code is **NOT** always better than an ordinary fixed-length code.
- Amortized analysis 與 average-case analysis 無關。
- (**FALSE**) **If** a graph has a unique MST then, for every cut of the graph, there is a **unique light edge** crossing the cut.
- (**TRUE**) A graph has a unique MST **if**, for every cut of the graph, there is a **unique light edge** crossing the cut.
- The worst-case running time and expected running time are equal to within **constant** factors for any randomized algorithm.
- Selection problem: $T(n) = T(\frac{n}{5}) + T(\frac{3n}{4}) + O(n)$
- Given an **undirected** graph and a positive integer k, is there a path of length $\leq k$, which each edge has weight 1 and each vertex is visited **exactly** once: P, solved by Floyd-Warshall algorithm.
- Given an **undirected** graph and a positive integer k, is there a path of length $\geq k$, which each edge has weight 1 and each vertex is visited $\leq$ once: NPC.
- A flow network of multiple sources can be reduced to a single source.

- Subset sum:
  $s(i, j)$: sum j can be found in $\{a_1, \cdots, a_i\}$

$$s(i,j) = \begin{cases} 0 & , i = 0 \\ 1 & , j = 0 \\ s(i-1,j) \vee s(i-1, j-v_i) & , j \geq v_i \end{cases} \tag{58}$$

  result is

$$s(m, n) \tag{59}$$

- Hanoi tower:
  - Iterative version: Check if the **input number** n is even or odd.
    If n is even,

$$\begin{cases} A \leftrightarrow C \\ A \leftrightarrow B \\ C \leftrightarrow B \end{cases} \tag{60}$$

    If n is odd,

$$\begin{cases} A \leftrightarrow B \\ A \leftrightarrow C \\ B \leftrightarrow C \end{cases} \tag{61}$$

  - Convert to undirected graph and solved by Hamiltonian path problem:

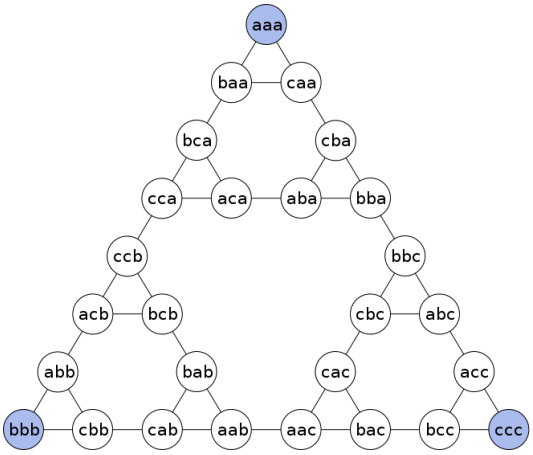


Figure 3: Example of Hanoi tower converted to undirected graph and solved by Hamiltonian path problem. For each node, disk positions from left to right in order of increasing size, and edges represent moves.

- Fibonacci search:

```
def fibSearch(arr, data):
    max = len(arr) - 1
    y = getY(fib, max + 1) # Find the largest index, which its value is
        smaller than data.
    m = max - fib[y]
    x = y - 1
    i = x
    if arr[i] < data: # Check at first.
        i += m
    while fib[x] > 0:
        if arr[i] < data:
            x -= 1
            i += fib[x]
        elif arr[i] > data:
            x -= 1
            i -= fib[x]
        else:
            return i
    return -1
```

Listing 2: Fibonacci search.

- Box stacking: create a stack of boxes which is as tall as possible, but you can only stack a box on top of another box if the dimensions of the 2-D base of the lower box

are each strictly larger than those of the 2-D base of the higher box.

1. Generate all 3 rotations of all boxes. We consider width as always smaller than or equal to depth.
2. Sort the above generated $3n$ boxes in **decreasing** order of **base area**.
3. $msh(i)$: Max possible stack height with box i at top of stack.

$$msh(i) = \{\max\{msh(j)\} + height(i)\}, \quad \forall j < i \wedge width(j) > width(i) \wedge depth(j) > depth(i) \tag{62}$$

result is

$$\max_{0<i<n}\{msh(i)\} \tag{63}$$

- Building bridge: connect as many north-south pairs of cities as possible with bridges such that no two bridges cross.
  1. Sort the north-south pairs on the basis of **increasing** order of **south** x-coordinates.
  2. Find **LIS** of north x-coordinates.
- Optimal strategy: play a game against an opponent by alternating turns. In each turn, a player selects either the first or last coin from the row, removes it from the row permanently, and receives the value of the coin. Determine the maximum possible amount of money we can definitely win if we move first.
  $f(i,j)$: max value the user can collect from i-th coin to j-th coin.

$$f(i,j) = \begin{cases} v_i & ,j = i \\ \max\{v_i, v_j\} & ,j = i+1 \\ \max\{v_i + \min\{f(i+2,j), f(i+1,j-1)\}, & \\ \quad v_j + \min\{f(i+1,j-1), f(i,j-2)\}\} & ,\text{otherwise} \end{cases} \tag{64}$$

- (TIOJ-1097) Find the largest square submatrix with all 0s in a 0/1 matrix:
  $dp(i,j)$: max square submatrix in $i \times j$ left upper submatrix.

$$dp(i,j) = \min\{dp(i-1,j-1), dp(i,j-1), dp(i-1,j)\} + 1 \tag{65}$$

- (UVA-10934) Dropping water balloons (k balloons and height n):

$dp(i,j)$: max height i balloons can be dropped j times.

$$dp(i,j) = \begin{cases} dp(i,j-1) + dp(i-1,j-1) + 1 & , arr(i,j) = 1 \\ 0 & , arr(i,j) = 0 \end{cases} \tag{66}$$

result is

$$\min_j \{dp(k,j) \geq n\} \tag{67}$$

- △ (TIOJ-1471) Skyline:
  $dp(i,j)$: number of legal path till the end through walking distance i and temporary height is j.

$$\begin{cases} dp(i,j) & = dp(i-1,j-1) + sum(j) \\ sum(j) & = sum(j) - dp(i-j,j) + dp(i,j) \end{cases} \tag{68}$$

  result is

$$\sum_j dp(n,j) \tag{69}$$

- (leetcode-84) Largest rectangle in histogram:
  - If the new element is higher than stack top element, push it; otherwise, pop and calculate the area until the new element is higher than stack top element.
  - Maximal rectangle: Similarly, for each column, the count of 1 of each row, can be seen as the element.
- AOV network topological order is **NOT** unique.
- (leetcode-97) Interleaving string:
  $dp(i,j)$: if $s_1[0:i-1]$ and $s_2[0:i-1]$ can be combined as $s_3[0:i+j-1]$.

$$dp(i,j) = \begin{cases} \text{true} & , i = j = 0 \\ (dp(i-1,j) \&\& (s_1(i-1) == s_3(i+j-1))) \;\; || & \\ (dp(i,j-1) \&\& (s_2(j-1) == s_3(i+j-1))) & , \text{otherwise} \end{cases} \tag{70}$$

- (leetcode-115) Distinct subsequences:

$dp(i,j)$: number of subsequence of $t[1:i]$ equals $s[1:j]$ (strings start from index 1).

$$dp(i,j) = \begin{cases} 1 & ,i = 0 \\ dp(i,j-1) + dp(i-1,j-1) & ,t[i] = s[j] \\ dp(i,j-1) & ,\text{otherwise} \end{cases} \tag{71}$$

result is

$$dp(n,m) \tag{72}$$

where n and m are lengths of strings t and s, respectively.

- △ (109NYCU-9) Prefix sum:
  $D(i,s)$: max number of elements that can be selected from first i integers with sum $\leq s$.

$$\begin{gathered} D(i,s) = \max\{D(i-1,s), D(i-1, \min(s - a_i, 6 \times a_i)) + 1\}, \\ \forall\ 1 < j \leq k,\ \text{s.t.}\ \sum_{l=1}^{j-1} a_{i_l} \leq 6 \times a_{i_j},\ 1 \leq i_1 < \cdots < i_k \leq n \end{gathered} \tag{73}$$

  result is

$$D(n, 6 \times a_{n+1}) \tag{74}$$

- (109NYCU-15) 0/1 Knapsack problem: if $W = \Theta(n^2)$, and $w_i \in \{1\} \vee w_i \in \{1,2\}$, then $T(n) = O(n)$.

- (leetcode-84) Largest Rectangle in Histogram:

```
def largestRectangleArea(self, heights):
    stack = []
    area = 0
    heights.append(0)
    n = len(heights)

    for i in range(n):
        if not stack or heights[i] > heights[stack[-1]]:
            stack.append(i)
        else:
            while stack and heights[i] <= heights[stack[-1]]:
                h = heights[stack[-1]]
                stack.pop()
                w = i if not stack else i - stack[-1] - 1
```

```
                area = max(area, h * w)
            stack.append(i)
    return area
```

Listing 3: Largest Rectangle in Histogram.

- – AVL trees are ideal for sorting items of an **ordered dictionary**.
  – Different number binary trees of height h:

$$H_n = \begin{cases} 2H_{n-1} \times \sum_{i=0}^{h-2} H_i + H_{i-1}^2 & , h \geq 2 \\ H_0 = 1, H_1 = 3 \end{cases} \tag{75}$$

(1) Prove that the inequality $\ell(u,w)+d(v,u)-d(v,w)\geq 0$ holds for any vertex $v\in V$ and any edge $(u,v)\in E$ on the graph G.

(2) Assume that a value $s(v)$ is attached to each vertex $v\in V$ on the graph G. Consider a new graph G' that comes from transforming G by replacing the length of each edge $(u,v)\in E$ with $\ell(u,v)+s(u)-s(v)\geq 0$. Prove that the shortest path on the graph G' between w and x is also the shortest path between w and x on the graph G.

(1) 因為 $d(v,w)$ 為 v 到 w 的 shortest path 長，因此任何一條自 v 到 w 的 path 之長均大於等於 $d(v,w)$ 。因為 $\ell(u,w)+d(v,u)$ 亦為一條自 v 到 w 之 path，因此：

$$\ell(u,w)+d(v,u)-d(v,w)\geq 0$$

(2) 考慮 $P=<u=v_0,v_1,\ldots,v_k=w>$ 為一條自 u 到 w 之 path，並令

$$\hat{w}(u,v)=\ell(u,v)+s(u)-s(v)$$

則：

$$\begin{aligned}\hat{w}(P)&=\sum_{i=1}^{k}\hat{w}(v_{i-1},v_i)\\&=\sum_{i=1}^{k}(\ell(u,v)+s(u)-s(v))\\&=\sum_{i=1}^{k}\ell(v_{i-1},v_i)+s(v_0)+-s(v_k)\\&=w(P)+s(u)-s(w)\end{aligned}$$

因此：

Shortest path P 可以使得 $\hat{w}(P)$ 最小 若且為若 此 P 可以使得 $w(P)$ 為最小。

(1)A company named TW Telecomm has a network of n switch stations connected by m high-speed communication links. Each customer's phone is directly connected to one station in his or her area. The engineers of TW Telecomm have developed a prototype video-phone system that allows two customers to see each other during a phone call. In order to have acceptable image quality, however, the number of links used to transmit video signals between the two parties cannot exceed 4. Suppose that TW Telecomm's network is represented by a graph. Design an efficient algorithm that computers, for each station, the set of stations it can reach using no more than 4 links. (12%)

(2)Please describe the running time of your algorithm. (3%)【94 年台師大資教】

解 (1)for $t \leftarrow 1$ to 3 {

for $i \leftarrow 1$ to n {

for $j \leftarrow 1$ to n {

$M^t[i, j] \leftarrow 0$;

for $k \leftarrow 1$ to n {

if ($M^{t-1}[i, j]=1$ or $M^{t-1}[i, k] \wedge M^{t-1}[k, j]=1$)

$M^t[i, j] \leftarrow 1$;

}

}

}

}

則與第 i 個 Station 不超過 4 個 link 的 Station 為 $\{k \mid M^4[i, k]=1\}$

Given an undirected graph $G=(V,E)$ with $n=|V|$ vertices, four vertices of G, say, u, v, x, and y, are said to form a 4-cycle if (u,v), (v,x), (x,y) and (y,u) are in E. Consider the problem of determining whether G contains a 4-cycle. A naïve method by checking all possible 4-combinations of the vertex set will need $\Omega(n^4)$ time to complete the job. Design a more efficient algorithm (i.e., the time complexity of your algorithm should be $O(n^k)$ with $k<4$) to solve the problem. Analysis the execution time of your algorithm.

令 M 為其 adjacent matrix

```
A ← O;
B ← M;
for i ← 1 to n {
  for j ← 1 to n {
    for k ← 1 to n {
      A[i, j] ← B[i, j] + B[i,k] * B[k, j];
    }
  }
}
```

```
for i ← 1 to n {
  for j ← 1 to n {
    if ( i ≠ j  and  B[i, j] ≥ 2 )
      return True;
  }
}
return False
```

Show how to determine in $O(n^2 \log n)$ time whether any three points in the set $S=\{(x_1,y_1),(x_2,y_2)...(x_n,y_n)\}$ are collinear. 【92 年台大資工所】

解 演算法設計如下：

```
for  i ← 1  to n
    begin
        在 S 中每一點依照與 (x_i, y_i) 所形成之向量依角度由小自大排序
        後，令其為 < p_1, p_2, ⋯, p_n >;
        for  j ← 1  to n
            if  m(p_j) = m(p_{j+1})
                return True;
    end
return False;
```

其精神在於只要有三點共線，則會算出兩個點與 (x_i,y_i) 所形成之向量之斜率相同。而經排序過後，只需檢驗第 j 個和第 $j+1$ 個向量之斜率有無相等即可。

範例 11

(1) The pseudocode _____ reduces the o-1 kanpsack decision problem to the 0-1 knapsack optimization problem.

KNAPSACKOPT(vi,wi,W) denotes an integer-valued function that solves the 0-1 knapsack optimization problem with the instance vi, wi and W.

KNAPSACKDEC(vi, wi, W, B) denotes a Boolean-valued function that solves the 0-1 knapsack decision problem with the instance vi, wi, W and B, where B is a positive integer lower bound.

(2) The pseudocode _____ reduces the 0-1 knapsack optimization problem to the 0-1 knapsack decision problem. 【98 年交大資工】

解 (1) M ← **KNAPSACKOPT**(vi, wi, W);

```
if(M>=B)
  return True;
else
return False;
```

(2) M ← 0;

```
for i ← 1 to n
  M ← M + vi;
for B ← 0 to M;
  if (KNAPSACKDEC(vi, wi, W, B)=False)
    return B-1;
```

(a) 給定一個 endpoint Hamiltoniam path problem 的 instance：(G,a,b)，則可以將其轉換成一個 Hamiltonian cycle problem：G'，其中 G'為 G 中多加入(a,b)這個邊。

當 G 有從 a 到 b 的 Hamiltonian path P，則 P 中加上(a,b)這個邊即為 G' 中的 Hamiltonian Cycle C。反之，若有 G'中有 Hamiltonian cycle C，則將(a,b)去掉後，則為 G 中從 a 到 b 的 Hamiltonian path。

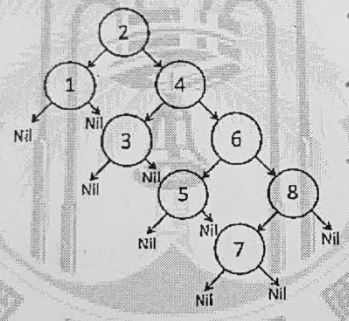
(b) 給定一個 Hamiltonian cycle problem 的 instance: G=(V,E)，則可以將其轉換成一個 Traveling salesman problem 的 instance: (G'=(V,E'), k=|V|)，其中每一個在 E 中的邊(u,v)，在 E'中之 weight 為 1；每一個不在 E 中的邊(u,v)，在 E'中之 weight 為 2。

當 G 有 Hamiltonian Cycle C 時，C 亦為 G'中 total distance <= k 的 Hamiltonian Cycle。反之，若 G'中有一個 Hamiltonain Cycle C 其 total distance <=k，則 C 中必不含 weight 為 2 的邊，則其為 G 中的 Hamiltonian cycle。

(c) 給定一個 Hamilonian path problem 的 instance: G, 則可以將其轉換成 Bounded degree spanning tree problem 的 instance: (G, D=2)。

當 G 有 Hamiltonian path，則 G 有 D=2 的 spanning tree。反之，當 G 中有 degree 均小於等於 2 的 spanning tree 時，tree 包含所有 G 上的點且 tree 上必只有兩個點的 degree 小於 2，此即為 G 的 Hamiltonian path。

2. (10 points) A red-black tree is a special type of binary search trees. Given the following binary search tree on 8 keys (i.e., integers 1 . . . 8), please answer the questions.



(a) (5 points) The tree cannot be simply transformed to a red-black tree by coloring its internal nodes. Explain what property of the red-black tree cannot be satisfied.

3. †† (13%) A Hamiltonian cycle of a graph G is a simple cycle that visits all nodes in G. Suppose there exists an $O(n^7)$-time algorithm that decides $\text{HamC}(G)$ for any n-node graph G.

$\text{HamC}(G)$

Input: a simple undirected graph G.

Output: "true," if G has a Hamiltonian cycle; "false," otherwise.

Complete Algorithm 1, an $O(n^7)$-time algorithm that uses HamC at most once to decide $\text{HamP}_{2\times3}$ for any n-node graph G, for any distinct nodes $a_1, a_2, x_1, x_2, x_3 \in G$.

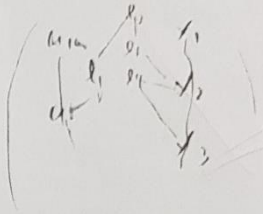
$\text{HamP}_{2\times3}(G = (V, E), a_1, a_2, x_1, x_2, x_3)$

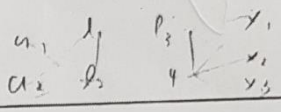
Input: a simple undirected graph $G = (V, E)$ that contains at least the five distinct nodes a_1, a_2, x_1, x_2, x_3.

Output: "true," if G has a simple path of length $|V| - 4$ that starts at a_i for some $i \in \{1, 2\}$, visits every node in $V \setminus \{a_1, a_2, x_1, x_2, x_3\}$ exactly once, and finally stops at x_j for some $j \in \{1, 2, 3\}$; otherwise, "output false."

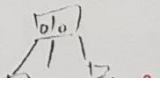
Algorithm 1: $\text{HamP}_{2\times3}(G = (V, E), a_1, a_2, x_1, x_2, x_3)$

```
1 U ← V ∪ {ℓ1, ℓ2, ℓ3, ℓ4};
2 F ← E;
  /* Add some edges incident to node ℓ1 to F.   */
3 F ← F ∪ {      };
  /* Add some edges incident to node ℓ2 to F.   */
4 F ← F ∪ {      };
  /* Add some edges incident to node ℓ3 to F.   */
5 F ← F ∪ {      };
  /* Add some edges incident to node ℓ4 to F.   */
6 F ← F ∪ {      };
7 return HamC(H = (U, F));
```





- Which of the following undirected edges shall be placed in the missing part ⑥ of Line 3?
  (A) (ℓ_1, ℓ_2) (B) (ℓ_1, a_1) (C) (ℓ_1, a_2) (D) (ℓ_1, x_2)
- Which of the following undirected edges shall be placed in the missing part ⑦ of Line 4?
  (A) (ℓ_2, a_1) (B) (ℓ_2, a_2) (C) (ℓ_2, x_2) (D) (ℓ_2, x_3)
- Which of the following undirected edges shall be placed in the missing part ⑧ of Line 5?
  (A) (ℓ_3, ℓ_1) (B) (ℓ_3, ℓ_2) (C) (ℓ_3, x_1) (D) (ℓ_3, x_2)
- Which of the following undirected edges shall be placed in the missing part ⑨ of Line 6?
  (A) (ℓ_4, ℓ_3) (B) (ℓ_4, x_1) (C) (ℓ_4, x_2) (D) (ℓ_4, x_3)



16. †† (8%) A Hamiltonian cycle of a graph G is a simple cycle that visits all the nodes in G. Suppose that there is an $O(n^7)$-time algorithm that decides HamC(G) for any n-node graph G.

HamC(G)

Input: a simple undirected graph G

Output: "true," if G has a Hamiltonian cycle; "false," otherwise.

Complete Algorithm 5, which is an $O(n^7)$-time algorithm that uses HamC(G) at most once to decide HamC3(G, x, y, z) for any n-node graph G, for some distinct nodes $x, y, z \in G$.

HamC3($G = (V, E), x, y, z$)

Input: a simple undirected graph $G = (V, E)$ of $|V| \geq 3$ and three distinct nodes $x, y, z \in G$.

Output: "true," if G has a Hamiltonian cycle C on which x, y, z are consecutive nodes in an arbitrary order; "false," otherwise.

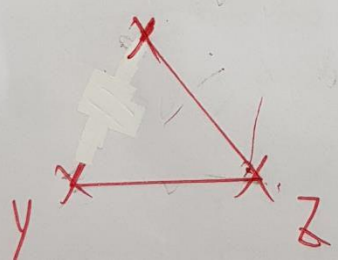
Algorithm 5: HamC3($G = (V, E), x, y, z$)

```
1  U ← V ∪ {a, b};
2  F ← E;
3  if at least two of edges (x,y),(y,z),(z,x) are not contained in E then
4      return _____;
5  else if exactly one of edges (x,y),(y,z),(z,x) is not contained in E then
       /* Assume w.l.o.g. that (x,y) ∉ E */
6      F ← F ∪ {_____};
7  else
8      F ← F ∪ {_____};
9  end
10 return HamC(H = (U, F));
```

- Which of the following shall be placed in the missing part (36) of Line 4?
  (A) true (B) false (C) HamC($G = (V, E)$) (D) HamC($H = (U, F)$) (E) none of the above

- Which of the following shall be placed in the missing part (37) of Line 6?
  (A) $(a, x), (b, y)$ (B) $(a, y), (b, z)$ (C) $(a, z), (b, x)$ (D) (a, b) (E) none of the above

- Which of the following shall be placed in the missing part (38) of Line 8?
  (A) $(a, x), (b, y)$ (B) $(a, y), (b, z)$ (C) $(a, z), (b, x)$ (D) (a, b) (E) none of the above



```
Cycle-Removal(G,u)
 Initialize S as a stack
 Label all vertices in G as Type-0
 Label u as Type-1
 S.push(u)
 while S is not empty
   v = the top vertex of S // Comment: not S.pop() here
   if v has a Type-A neighbor w in G
     Remove (v,w) from G
   else if v a Type-B neighbor w in G
     Label w as Type-C
     S.push(w)
   else
     Label v as Type-D
     S.pop()
   end if
 end while
 output G
```

white gray gray white gray black

A : 1
B : 0
C : K
D : 2

Thm. Vertex Cover $\leq_P$ Set Cover

Proof. Let $G = (V, E)$ and k be an instance of VERTEX COVER. Create an instance of SET COVER:

- $U = E$
- Create a S_u for for each $u \in V$, where S_u contains the edges adjacent to u.

U can be covered by $\leq k$ sets iff G has a vertex cover of size $\leq k$.

Why? If k sets $S_{u_1}, \ldots, S_{u_k}$ cover U then every edge is adjacent to at least one of the vertices $u_1, \ldots, u_k$, yielding a vertex cover of size k.

If $u_1, \ldots, u_k$ is a vertex cover, then sets $S_{u_1}, \ldots, S_{u_k}$ cover U. □

4. (20 points) Find a shortest path in a DAG with given constraints. A graph is a *DAG* if it is *directed* and it has *no* cycle. Every edge has a positive integer distance, and is either *thick* or *thin*. Figure 2 illustrates such a DAG. For example, the edge from A to B is thick and with a distance 1, and the edge from A to C is thin and has a distance 2.

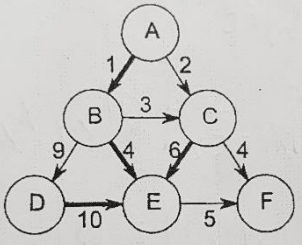


Figure 2: A DAG

Now given nodes A and F, and two integers k, n, find the shortest path from A to F that goes through exactly k thick edges and n thin edges, and does not go through any node twice. For example, the shortest distance from A to F that goes through two thick and two thin edges is 15 by going through A-B-C-E-F. The path A-C-E-F is not a valid solution because it only goes through one thick edge. The path A-B-D-E-F does go through the right number of edges, but its total distance is 25, which is not the minimum.

Please describe the key ideas in your algorithm *without using pseudo code*. Finally analyze the time complexity of your algorithm. We assume that there are N nodes and M edges in the DAG.

Your entire description and analysis can only use at most *400* Chinese characters or *400* English words. *Over-sized description or analysis will be ignored. Pseudo code will also be ignored.*

4. $V' := \text{Topological-sort}(V)$

$E_1 :=$ thick edges

$E_2 :=$ thin edges

$DAG(i, x, y)$ 為從 v_1 走到 v_i 並將過 x thick edges 和 y thin edges 的 shortest path.

若 v_1 到 v_i 的最後一條路為 (v_j, v_i)，則

$$D(i,x,y) = \begin{cases} w(j,i) + D(i,x-1,y) & \text{if } (v_j,v_i) \in E_1 \\ w(j,i) + D(i,x,y-1) & \text{if } (v_j,v_i) \in E_2 \end{cases}$$

$$\hookrightarrow D(i,x,y) = \min\left\{ \min_{(v_j,v_i)\in E_1} \{D(i,x-1,y) + w(j,i)\}, \min_{(v_j,v_i)\in E_2} \{D(i,x,y-1) + w(j,i)\} \right\}$$

$(2 \le i \le N,\ 0 \le x \le |E_1|,\ 0 \le y \le |E_2|$，$x, y$ 不能為 0)

Initial condition $D(1,0,0) = 0$

最終可得 $D(N,k,n)$ 即 v_1 走到 v_N 中經過 k thick edges 和 n thin edges 所得 shortest path.

Time complexity : $O(N+M) + kn \times O(N+M)$

Topological-sort $\le kn$ edges　traverse graph

$= O(kn(N+M))$

A sequence $x_1, x_2, \ldots, x_n$ is said to be cyclically sorted if the smallest number in the sequence is x_i for some unknown i, and the sequence $x_i, x_{i+1}, \ldots, x_n, x_1, \ldots, x_{i-1}$ is sorted in increasing order. Please design and illustrate an O(log n) algorithm to find the position of the minimal element in a cyclic sorted sequence of n elements by using the example sequence 45, 52, 66, 72, 3, 29, 38.

【98 年政大資科】

解 **Procedure** FindSmall(X)

```
{
 i ← 1;
 j ← n;
 while(i<j)
 {
    k ← (i+j)/2;
    if (X[k]<X[j])
        j ← k;
    else
        i ← k+1;
 }
 return i;
}
```

→ i=j 時就要 return，∴要記到子串開頭名

Let $A[1..n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A.

(1) List the five inversions of the array <2,3,7,6,1>

(2) What array with elements from the set $\{1,2,\ldots,n\}$ has the most inversions? How many does it have?

(3) Give an algorithm that determines the number of inversions in any permutation on n distinct numbers in $O(n \log n)$ worst-case time. (Hint: Modify merge sort)

【90 年台大資工所】

解 (1) (1,5), (2,5), (3,4), (3,5), (4,5)

(2) $<n, n-1, \ldots, 2, 1>$最多，因為任取兩個元素均可形成 inversion，所以共有 $\binom{n}{2}$ 個 inversion。

(3) Inversion 數計算可由修改 merge sort 得知。→ Merge 時，右半部 merge 皆需加上左半部當時長度

【程式說明】

p：該 partition 開始的位置
q：該 partition 中間的位置
r：該 partition 最後的位置
c：計算 inversion 數目
L 矩陣：存 A 的左半
R 矩陣：存 A 的右半

【Pseudo code】

```
Merge-Sort(A, p, r)
    if p < r
        then
            q ← FLOR( (p+r)/2 );
            return Merge-Sort (A, p, q) + Merge-Sort (A, q+1, r) +
                   merge(A, p, q, r);
    else
        return 0;
```

```
merge (A, p, q, r)
    n1 ← q−p+1;
    n2 ← r−q;

    for i ← 1 to n1
        L[i] ← A[p+i−1];
    for j := 1 to n2
        R[j] ← A[q+j];

    L[n1+1] ← ∞;
    R[n2+1] ← ∞;
    c ← 0, i ← 1, j ← 1;

    for k ← p to r
        if ( L[i] <= R[j] )          //左邊的比右邊大，所以無 inversion
            then A[k] ← L[i];
                 i ← i+ 1;
        else A[k] ← R[j];            //右邊比左邊大，有 inversion
             c ← c + (n1−i+1);       //因為 L 已經排序好，L[i] <= R[j]
             j ← j+1;                //則 L[i+1]~L[n1]都比 R[j]大，所以 c=c+(n1−i+1)
    return c;
```

右>左

左>右

例 3.1

Given n items of values V_1, V_2,…,V_n, and weights W_1, W_2,…, W_n, and a knapsack of capacity C, the so-called knapsack problem is to find X_1, X_2,…, X_n, where $0 \leqq X_i \leqq 1$, such that $\sum W_i X_i \leq M$. Please give a greedy method to find an optimal solution of the knapsack problem, analyze its time complexicy and prove its correctness. 【94 年交大生資所】

解 演算法及時間複雜度請參考上述之演算法。

正確性證明如下：

不失一般性令物品價值 $v_1 \geqq v_2 \geqq \ldots \geqq v_n$。令 $X =< x_1,\ldots,x_n >$ 代表用 greedy 得到的解向量，其中 $0 \leqq x_i \leqq 1$ 為取物時取物品 i 之重量為 $w_i * x_i$。

設 j 為第一個 index 使得 $x_j < 1$。

令 $Y =< y_1,\ldots,y_n >$ 為任何一解向量且 X 不等於 Y。

因為負重不可超過 W，$\sum_{i=1}^{n} w_i * y_i \leq W$。

另外，由演算法可知：$\sum_{i=1}^{n} w_i * x_i = W$，因此 $\sum_{i=1}^{n} w_i * (x_i - y_i) \geq 0$。

考慮 X 和 Y 取物的總值差：

$\sum_{i=1}^{n} v_i * (x_i - y_i)$，因為 $\frac{v_i}{w_i}(x_i - y_i) \geq \frac{v_j}{w_j}(x_i - y_i)$，所以我們可以推得：

$$\sum_{i=1}^{n} v_i * (x_i - y_i) = \sum_{i=1}^{n} v_i * \frac{w_i}{w_i}(x_i - y_i) \geq \frac{v_j}{w_j} \sum_{i=1}^{n} w_i (x_i - y_i) \geq 0$$

因此，X 為最佳解。

例 3.3

Let $G=(V,E)$ be a connected undirected graph with a weight function $w: E \to R$, where R is the set of real numbers and $|V|=n$. Kruskal's algorithm for finding minimum spanning tree can be summarized as follows. First, choose an edge in the graph with minimum weight. Sucessively add edge with minimum weight that do not from a simple cucle with those edges already chosen. Stop after $n-1$ edge have been chosen.

Prove correctness of Kruskal's algorithm. 【95 年交大資工所】

解 令 T 為用 Kruskal 演算法找出之 Spanning Tree，T' 為 G 之 Minimum Spanning Tree。

若 $T=T'$，則得證。

否則，因為 $E(T) \neq E(T')$，所以在 $E(T)-E(T')$ 中取一個最小 weight 的邊 e。

將 e 加入 T' 後，會形成一個 cycle C。

則 C 中任一個不屬於 $E(T)$ 之邊 e' 之 weight 必大於等於 e

(因為若 e' 之 weight 小於 e，則 Kruskal 在選擇時必會先選 e'，如此一來會導致矛盾)

因此，可造一個 Spanning Tree $T''=(V, E'-\{e'\}\cup\{e\})$

而 weight$(T'') \leqq$ weight(T')

重覆上述步驟，我們便可以將 T' 變成 T 而不增加其 weight。又 T' 為 Minimum Spanning Tree，所以可知 weight(T) = weight(T')。即 T' 為 Minimum Spanning Tree。

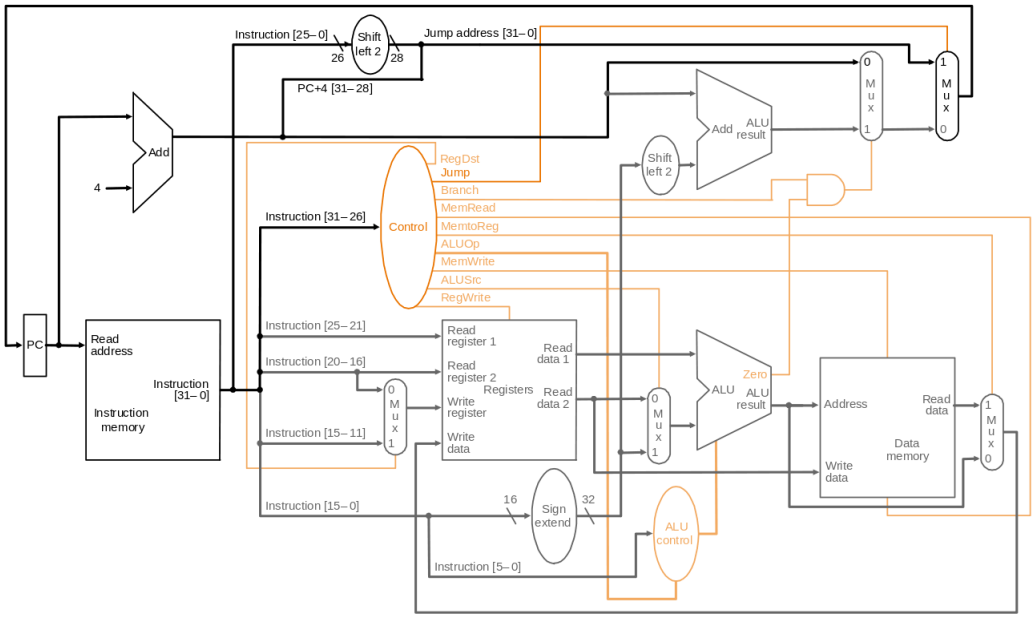


Figure 4: Single-cycle CPU with jump and branch.

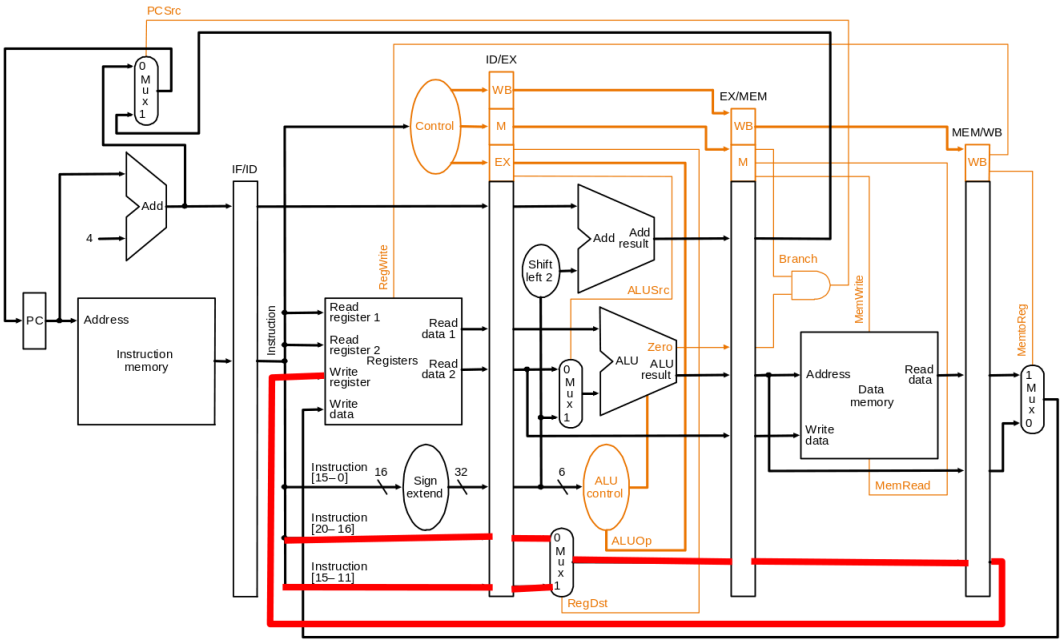


Figure 5: Original pipeline.

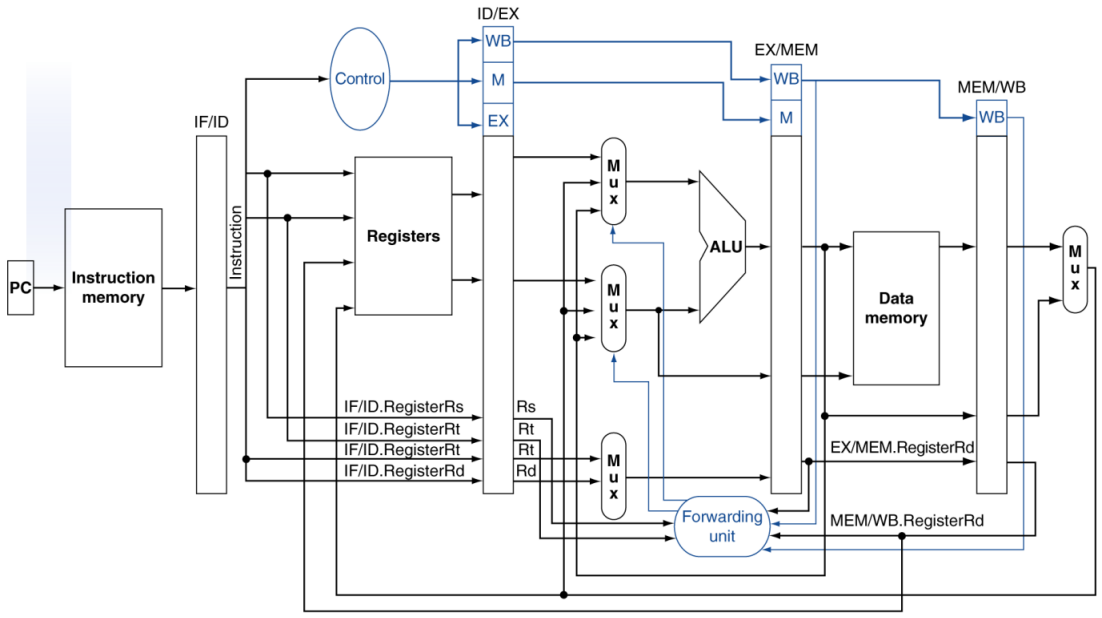


Figure 6: Pipeline with forwarding.

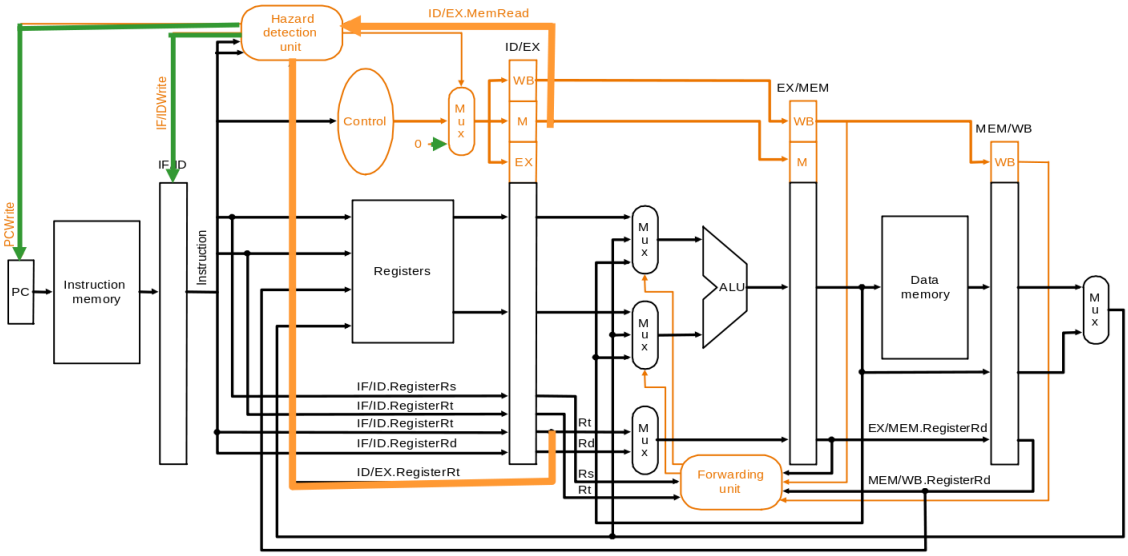


Figure 7: Pipeline with hazard detection and forwarding units.

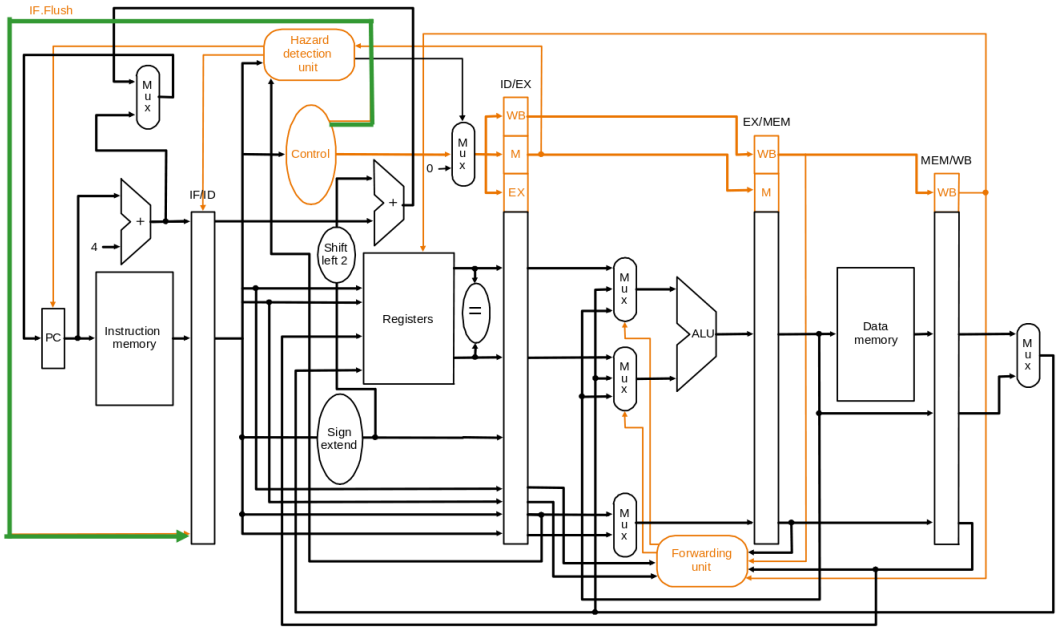


Figure 8: Pipeline with hazard detection, forwarding units and flush.

- Endianness：
  - Big Endian：最左邊或 MSB 放在最低 address，e.g. MIPS。
  - Little Endian：最右邊或 LSB 放在最低 address，e.g. x86。
-
  - `srl/sll rd, rt, shamt # rs = 5'0`
  - `lw/sw rt, imm(rs)`
  - `beq/bne rs, rt, addr`
  - `addi rt, rs, imm`
  - `lb`：Load 最高 address（LSB）到最高 address（LSB），`sb`：Store 最高 address（LSB）到最低 address（MSB）。
  - `jr rs # rt = rd = shamt = 5'0`：R-type
-

```
int fact (int n) {
    if (n < 1)
        return 1;
```

```
        else
            return (n * fact(n - 1));
    }
```

```
    fact:
        addi $sp, $sp, -8
        sw $ra, 4($sp)
        sw $a0, 0($sp)
        slti $t0, $a0, 1
        beq $t0, $zero, L1
        addi $v0, $zero, 1
        addi $sp, $sp, 8
        jr $ra
    L1:
        addi $a0, $a0, -1
        jal fact
        lw $a0, 0($sp)
        lw $ra, 4($sp)
        addi $sp, $sp, 8
        mul $v0, $a0, $v0
        jr $ra
```

- 浮點數：

Single precision		Double precision		Representation
Exponent	Fraction	Exponent	Fraction	
0	0	0	0	± 0
0	$\neq 0$	0	$\neq 0$	$\pm$ denormalized number
$1 \sim 254$	$\times$	$1 \sim 2046$	$\times$	$\pm$ floating-point number
255	0	2047	0	$\pm\infty$
255	$\neq 0$	2047	$\neq 0$	NaN

- Overflow detection：
  - 有號數：

```
        addu $t0, $t1, $t2
        xor $t3, $t1, $t2
        slt $t3, $t3, $zero # $t3 = 1 if sign differs
        bne $t3, $zero, NO_OVERFLOW
        xor $t3, $t0, $t1 # Check if the sum sign differs
        slt $t3, $t3, $zero
        bne $t3, $zero, OVERFLOW
```

– 無號數：

```
addu $t0, $t1, $t2
nor $t3, $t1, $zero # 2^32 − $t1 − 1
sltu $t3, $t3, $t2 # 2^32 − $t1 − 1 < $t2 → 2^32 − 1 < $t1 + $t2
bne $t3, $zero, OVERFLOW
```

Instruction	ALUOp1	ALUOp2
lw/sw	0	0
beq	×	1
R-type	1	×

•

• Data hazards：

– Forwarding：Combinational units，放在 EX 因為 ALU。

```
if (EX/MEM.RegWrite ∧ (EX/MEM.Rd ≠ 0) ∧
    (EX/MEM.Rd = ID/EX.Rs/Rt))
    ForwardA/B = 10
```

Listing 4: EX hazard.

```
if (MEM/WB.RegWrite ∧ (MEM/WB.Rd ≠ 0) ∧
    (¬ EX_hazard) ∧ (MEM/WB.Rd = ID/EX.Rs/Rt))
    ForwardA/B = 01
```

Listing 5: MEM hazard.

– Stall：

```
if (ID/EX.MemRead ∧ (ID/EX.Rt = IF/ID.Rs/Rt))
    IF/ID.Write := 0
    PC.Write := 0
```

Listing 6: Stall.

• Control hazards：

– 若分支指令與**前一個 ALU 指令**或**前面第二個** lw 有 data dependency，必須 stall 1 CC。

– 分支指令通過 xor 再 nor 比較是否相同。

– Delayed branch：

  * NOT suitable for deep pipeline.
  * From before：最佳方法，不管跳或不跳皆提升。
  * From target：用於 branch 發生機率高，有跳才提升。
  * From fall through：用於 branch 發生機率低，不跳才提升。

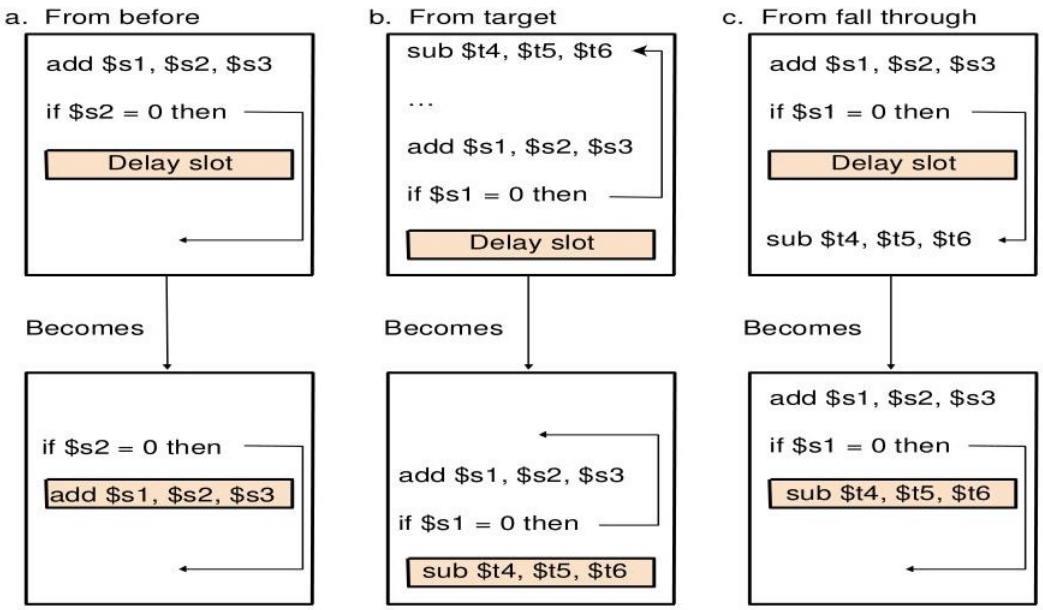


Figure 9: Example of delayed branch.

•

– Intel IA-64 (EPIC)：

  * 支援利用 compiler 開發的平行度。
  * 可以猜測，並利用 if-else 取代 branch。
  * Registers 比 MIPS 多很多。
  * Instruction group is a sequence of instructions which does **NOT** have data dependency and can be executed parallelly.

– Speculation 錯誤復原：

  * 軟體提供修補程式。
  * 硬體 CPU 將猜測結果暫時儲存，若正確，則將猜測結果寫回 register 或 memory，否則 flush buffer。

Advanced pipeline		
Technique	Hardware	Software
Branch prediction	√	√
Speculation	√	√
Intel IA-64 (EPIC)	√	√
Register renaming	√	√
Prediction		√

- MIPS exception handling：
  - Flush the instruction and let all preceding instructions complete if they can.
  - 利用 cause register 儲存 exception 原因。
  - 將造成 exception 的 instruction memory address 存在 EPC ($PC + 4$)。
  - 使用 entry point switch to kernel。
  - Exception handling routine 須將 $PC - 4$。
- Non-blocking cache：
  - Does **NOT** allow **miss under hit** to hide miss latency.
  - **Miss under miss** allows multiple outstanding cache misses.
  - Allow a **load** instruction to access the cache if the previous **load** is a cache miss.
- RAID：
  - RAID 2：Hamming code，Write 需要讀取所有 disks，從新計算 Hamming code 並寫入 ECC disks，效率差，$2n - 1$ disks。
  - RAID 3：
    - Reliability 和 RAID 2 相同。
    - 不做備份，花費較多時間恢復 data，$n + 1$ disks。
    - 當 1 個 disk 出錯可救回來，多個則否。
    - Availability cost 為 $\frac{1}{N}$，其中 N 為 protection group disks 數量。
    - Parity 集中存放一個 disk。
  - RAID 4：
    - 只對 protection group 其中一 disk 做 small reads。
    - $n + 1$ disks，parity 集中存放一個 disk。

    - 當 1 個 disk 出錯可救回來，多個則否。
  - RAID 5：
    - Write 就不會有單一 disk 瓶頸。
    - $n+1$ disks，parity 被分散到所有 disks。
    - 可允許 1 個 disk 故障。
  - RAID 6：
    - 與 RAID 5 相比，增加第二個獨立的 parity block。
    - 通常通過硬體實現。
    - $n+2$ disks。
    - 可允許 2 個 disk 故障。
  - RAID 3 has **worst** throughput for **small writes**.
  - RAID 3 has **best small writes latency**.
  - RAID 3, 4, 5 have same throughput for **large writes**.
  - RAID 1 can **NOT** have **small writes** in parallel.
  - RAID 3 can **NOT** have **small writes or reads** in parallel.
  - RAID 4, 5 perform same for parallel **small reads and writes**.
  - RAID 4 does **NOT** have better **big reads** performance than RAID 3.
  - RAID 1+0 has better **write throughput** than RAID 0+1.
-

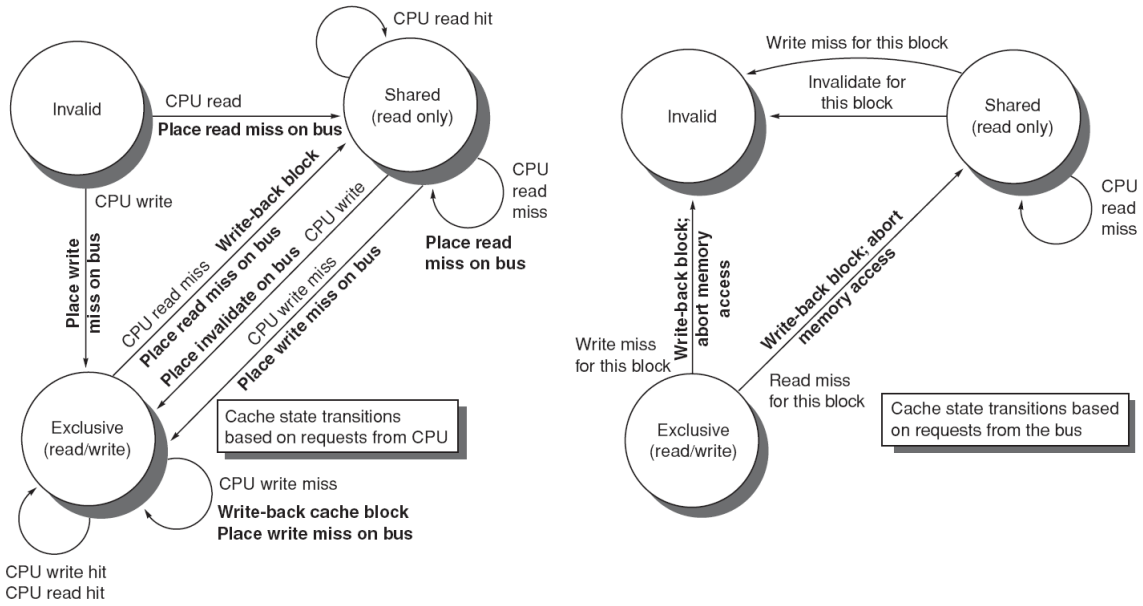


Figure 10: Snooping states.

- Multithreading：
  - Coarse-grained multithreading: Switch threads only on costly stalls, such as L2 cache misses. Pipeline **start-up** costs.
  - Fine-grained multithreading: Switch between threads on each instruction packs. It can hide the throughput losses.
  - SMT: **ILP** and **TLP**; coarse-grained and fine-grained: **TLP only**.
- Network topology：
  - Performance measure：
    - Network bandwidth.
    - Bisection bandwidth：平均切為二所減少的 bandwidth，越高容錯力越高。
    - Diameter：任兩點最短路的最大值，越低越好。
    - Nodal degree：CPU degree，越高容錯力越高。
  - Omega network hardware: $2n \log_2 n$.
  - Crossbar network hardware: n^2.
-

  - Time-sharing (Multitasking)：使用 virtual memory 以及 spooling，且對所有 users 公平對待。
  - Real-time：
    * **Hard** real-time disk 少用，不使用 virtual memory；但 **soft** real-time 可，但 real-time processes 的 pages 在完成前不能被 swapped out。
    * **Hard** real-time 不與 time-sharing 並存；但 **soft** real-time 可。
    * 減少 kernel 干預時間，因為 Linux kernel 在執行某些 system process 時，不允許 user process preempts kernel，防止 race condition。

-
  - Interrupt：Hardward-generated, e.g. I/O-complete, Time-out.
  - Trap：Software-generated。Catch arithmetic error 或重大 error，例如 Divide-by-zero，以及 process 需要 OS 提供服務，會先發 trap 通知 OS。

- Scheduler：
  - Long-term (Job) scheduler：通常僅 **batch system 採用**，從 job queue 中選 jobs 載入 memory。執行頻率最低，可以調控 multiprogramming degree 與 CPU-bound 與 I/O-bound jobs 的比例。
  - Short-term (CPU, process) scheduler：從 ready queue 選擇一個 process 分派給 CPU 執行。**所有系統都需要**，執行頻率最高，**無法**調控 multiprogramming degree 與 CPU-bound 與 I/O-bound jobs 的比例。
  - Medium-term scheduler：Memory space 不足且有其他 processes 需要更多 memory 時執行，選擇 Blocked 或 lower priority process swap out to disk。僅 **Time-sharing system 採用**，batch 和 real-time systems 不採用，可以調控 multiprogramming degree 與 CPU-bound 與 I/O-bound jobs 的比例。

- Dispatcher：
  - 將 CPU 真正分配給 CPU scheduler 選擇的 process。
  - Context switching.
  - Switch mode to user mode.
  - Jump to execution entry of user process.

- CPU scheduling：

- Non-preemptive SJF 不適合用在 **short-term** scheduler，因為很難在短時間算出 next CPU burst；long-term scheduler 較合適。
- MFQ 雖然不公平，但 **NO** starvation。
- Linux 指定 processes 不要移轉到某些 processors。
- Worst-case CPU utilization for scheduling n processes using Rate-monotonic:

$$\begin{aligned} & 2 \times (2^{\frac{1}{n}} - 1) \\ \Rightarrow & (n \to \infty) = 69\% \end{aligned} \tag{76}$$

- Dispatch latency：
  - Conflict phase：preempts kernel，並且 low-priority process releases needed resources for high-priority process。
  - Dispatch phase：Context switching, change mode to user mode, and jump to the user process.

- Deadlock avoidance：
  - 若 n processes，m resources（單一種類），若滿足

$$\begin{aligned} & 1 \leq Max_i \leq m \\ & \sum_{i=1}^{n} Max_i < n + m \end{aligned} \tag{77}$$

則 NO deadlock。

Proof. 若所有資源都分配給 processes，即

$$\sum_{i=1}^{n} Allocation_i = m \tag{78}$$

又

$$\begin{aligned} & \sum_{i=1}^{n} Need_i = \sum_{i=1}^{n} Max_i - \sum_{i=1}^{n} Allocation_i \\ \to & \sum_{i=1}^{n} Max_i = \sum_{i=1}^{n} Need_i + m \end{aligned} \tag{79}$$

根據第二條件，有

$$\begin{aligned} &\sum_{i=1}^{n} Max_i < n + m \\ \rightarrow &\sum_{i=1}^{n} Need_i < n \end{aligned} \tag{80}$$

$\exists$ process P_i，$Need_i = 0$，又

$$\begin{aligned} &Max_i \geq 1 \wedge Need_i = 0 \\ \rightarrow &Allocation_i \geq 1 \end{aligned} \tag{81}$$

在 P_i 完工後，會產生 ≥ 1 resources 給其他 processes 使用，又可以使 ≥ 1 processes P_j 有 $Need_j = 0$，依此類推，所有 processes 皆可完工。

- Critical section：
  - 在 critical section，CPU 也可能被 preempted。
  - 滿足：
    - Mutual exclusion：同一時間點，最多 1 process 在他的 critical section，不允許多個 processes 同時在**各自**的 critical section。
    - Progress：不想進入 critical section 時，不能阻礙其他想進入 critcal section 的 process 進入，即不能參與進入 critical section 的 decision，且必須在有限時間內決定進入 critical section 的 process。
    - Bounded waiting：Process 提出申請進入 critical section 後，必須在有限時間內進入，即公平，NO starvation。
- Two processes solution (Peterson's solution)：
  - 共享變數：

```
int turn = i ∨ j;
bool flag = False;
```

Listing 7: Shared variables of Peterson's solution (two processes solution).

  - $flag$ 或 $turn$ 或兩者值皆互換依然正確，但若將前兩行賦值順序對調，則因為 **mutual exclusion 不成立**，而不正確。
  - Peterson's solution is NOT guaranteed to work on modern PC, since processors and compilers may reorder read and write operations that have NO dependencies.

Algorithm 1 P_i of Peterson's solution (two processes solution).

```
1: function P_i
2:     repeat
3:         flag[i] := True
4:         turn := j
5:         while flag[j] ∧ turn = j do
6:         end while
7:         Critacal section.
8:         flag[i] := False
9:         Remainder section.
10:    until False
11: end function
```

- Test-and-Set：
  - 共享變數：

```
bool lock = False;
/*
True, 表示想進但在等;
False, 表示已在critical section或是初值。
*/
bool waiting[0 ··· (n − 1)] = False;
```

Listing 8: Shared variables of TEST-AND-SET solution.

  - 若移除第八行 `waiting[i] := False`，則 **progress 不成立**，若僅 P_i 和 P_j 想進入 critical section，此時 `waiting[i], waiting[j] = True`，且 P_i 先進入 critical section，有 `lock, waiting[i] = True`；當 P_i 離開 critical section 後，將 `waiting[j] := False`，P_j 進入 critical section；當 P_j 離開 critical section 後，因為 `waiting[i] = True`，P_j 將 `waiting[i] := False`，但 `lock = True`，未來沒有 process 可以再進入 critical section，**deadlock**。

Algorithm 2 P_i (Test-and-Set).

```
1: function P_i
2:     repeat
3:         waiting[i] := True
4:         key := True                                   ▷ Local variable.
5:         while waiting[i] ∧ key do
6:             key := TEST-AND-SET(&lock)
7:         end while
8:         waiting[i] := False
9:         Critical section.
10:        j := i + 1 (mod n)
11:        while j ≠ i ∧ ¬waiting[j] do                  ▷ 找下一個想進入的 P_j。
12:            j := j + 1 (mod n)
13:        end while
14:        if j = i then                                 ▷ 沒有 P_j 想進入 critical section。
15:            lock := False
16:        else
17:            waiting[j] := False
18:        end if
19:        Remainder section.
20:    until False
21: end function
```

- Producer-consumer problem：
  - 共享變數：

```
semaphore mutex = 1;
semaphore empty = n; // buffer空格數。
semaphore full = 0; // buffer中item數。
```

Listing 9: Shared variables of Producer-consumer problem.

  - 若將其中一個或兩個程式的兩行 `wait` 對調，可能會 **deadlock**。

Algorithm 3 Producer.

```
1: function PRODUCER
2:     repeat
3:         Produce an item.
4:         WAIT(empty)
5:         WAIT(mutex)
6:         Add the item to buffer.
7:         SIGNAL(mutex)
8:         SIGNAL(full)
9:     until False
10: end function
```

Algorithm 4 Consumer.

```
1: function CONSUMER
2:     repeat
3:         WAIT(full)
4:         WAIT(mutex)
5:         Retrieve an item from buffer.
6:         SIGNAL(mutex)
7:         SIGNAL(empty)
8:         Consume the item.
9:     until False
10: end function
```

- Reader/Writer problem：
  - R/W 和 W/W 皆要互斥。
  - First readers/writers problem：
    - 共享變數：

```
// R/W和W/W互斥控制，同時對writer不利之阻擋。
semaphore wrt = 1 ;
int readcnt = 0;
semaphore mutex = 1; // readcnt互斥控制。
```

Listing 10: Shared variables of First Reader/Writer problem.

Algorithm 5 Writer (First Reader/Writer problem).

```
1: function WRITER
2:     repeat
3:         WAIT(wrt)
4:         Writing.
5:         SIGNAL(wrt)
6:     until False
7: end function
```

Algorithm 6 Reader (First Reader/Writer problem).

```
1: function READER
2:     repeat
3:         WAIT(mutex)
4:         readcnt := readcnt + 1
5:         if readcnt = 1 then                ▷ 表示第一個 reader 需偵測有無 writer 在。
6:             WAIT(wrt)
7:         end if
8:         SIGNAL(mutex)
9:         Reading.
10:        WAIT(mutex)
11:        readcnt := readcnt − 1
12:        if readcnt = 0 then                ▷ No reader.
13:            SIGNAL(wrt)
14:        end if
15:        SIGNAL(mutex)
16:    until False
17: end function
```

- Second Reader/Writer problem：
  - 共享變數：

```
int readcnt = 0;
semaphore mutex = 1; // readcnt互斥控制。
semaphore wrt = 1; // R/W和W/W互斥控制。
int wrtcnt = 0;
semaphore y = 1; // wrtcnt互斥控制。
semaphore rsem = 1; // 對reader不利之阻擋。
semaphore z = 1; // reader的入口控制，可有可無。
```

Listing 11: Shared variables of Second Reader/Writer problem.

Algorithm 7 Writer (Second Reader/Writer problem).

```
1: function WRITER
2:     repeat
3:         WAIT(y)
4:         wrtcnt := wrtcnt + 1
5:         if wrtcnt = 1 then                    ▷ 表示第一個 writer 需阻擋 readers。
6:             WAIT(rsem)
7:         end if
8:         SIGNAL(y)
9:         WAIT(wrt)
10:        Writing.
11:        WAIT(y)
12:        wrtcnt := wrtcnt − 1
13:        if wrtcnt = 0 then
14:            SIGNAL(rsem)                      ▷ No writer.
15:        end if
16:        SIGNAL(wrt)
17:        SIGNAL(y)
18:    until False
19: end function
```

Algorithm 8 Reader (Second Reader/Writer problem).

```
1: function READER
2:     repeat
3:         WAIT(z)
4:         WAIT(rsem)
5:         WAIT(mutex)
6:         readcnt := readcnt + 1
7:         if readcnt = 1 then
8:             WAIT(wrt)
9:         end if
10:        SIGNAL(mutex)
11:        SIGNAL(rsem)
12:        SIGNAL(z)
13:        Reading.
14:        WAIT(mutex)
15:        readcnt := readcnt − 1
16:        if readcnt = 0 then
17:            SIGNAL(wrt)
18:        end if
19:        SIGNAL(mutex)
20:    until False
21: end function
```

- The sleeping barber problem：
  - 共享變數：

```
semaphore customer = 0; // 強迫barber sleep。
// 強迫customer sleep if barber is busy。
semaphore barber = 0;
int waiting = 0; // 正在等待的customers個數。
semaphore mutex = 1; // waiting互斥控制。
```

Listing 12: Shared variables of The sleeping barber problem.

  - 若將 BARBER 將兩行 `wait` 對調，可能會 **deadlock**。

Algorithm 9 Barber.

```
1: function BARBER
2:     repeat
3:         WAIT(customer)                    ▷ Barber go to sleep if no customer.
4:         WAIT(mutex)
5:         waiting := waiting − 1
6:         SIGNAL(barber)                    ▷ Wake up customer.
7:         SIGNAL(mutex)
8:         Cutting hair.
9:     until False
10: end function
```

Algorithm 10 Customer.

```
1: function CUSTOMER
2:     repeat
3:         WAIT(mutex)
4:         if waiting < n then                          ▷ 入店。
5:             waiting := waiting + 1
6:             SIGNAL(customer)                         ▷ Wake up barber.
7:             SIGNAL(mutex)
8:             WAIT(barber)                             ▷ Customer go to sleep if barber is busy.
9:             Getting cut.
10:        else
11:            SIGNAL(mutex)
12:        end if
13:    until False
14: end function
```

- The dining-philosophers problem：

– 五位哲學家兩兩間放一根筷子吃中餐（筷子），哲學家需取得左右兩根筷子才能吃飯。若吃西餐（刀叉），必須**偶數**個哲學家，

– Algorithm 1：

* 根據公式 (77)，人數必須 < 5 才不會 deadlock。
* 共享變數：

```
semaphore chopstick[0 ··· 4] = 1;
// 可拿筷子的哲學家數量互斥控制。
semaphore no = 4;
```

Listing 13: Shared variables of The dining-philosophers problem.

Algorithm 11 P_i of Algorithm 1 (The dining-philosophers problem).

```
1: function P_i
2:     repeat
3:         WAIT(no)
4:         Hungry.
5:         WAIT(chopstick[i])
6:         WAIT(chopstick[(i + 1 (mod 5))])
7:         Eating.
8:         SIGNAL(chopstick[i])
9:         SIGNAL(chopstick[(i + 1 (mod 5))])
10:        Thinking.
11:        SIGNAL(no)
12:    until False
13: end function
‘’
```

– Algorithm 2：只有能夠同時拿左右兩根筷子才允許持有筷子，否則不可持有任何筷子，**破除 hold and wait**，不會 deadlock。

– Algorithm 3：當有**偶數**個哲學家時，偶數號的哲學家先取左邊，再取右邊，奇數號的則反之，**破除 circular wait**，不會 deadlock。與吃西餐先拿刀再拿叉相似。

- Binary semaphore 製作 counting semaphore（若為 $-n$ 表示 n 個 process 卡在 `wait`）：

– 共享變數：

```
int c = n; // Counting semaphore號誌值。
semaphore s_1 = 1; // c互斥控制。
```

```
binary_semaphore s2 = 0; // c < 0時卡住process
```

Listing 14: Shared variables of The dining-philosophers problem.

Algorithm 12 *wait*(c) (counting semaphore).

```
1: function WAIT(c)
2:     WAIT(s1)
3:     c := c − 1
4:     if c < 0 then
5:         SIGNAL(s1)
6:         WAIT(s2)                          ▷ Process 卡住。
7:     else
8:         SIGNAL(s1)
9:     end if
10: end function
```

Algorithm 13 *signal*(c) (counting semaphore).

```
1: function SIGNAL(c)
2:     WAIT(s1)
3:     c := c + 1
4:     if c ≤ 0 then                         ▷ 先前有 process 卡住。
5:         SIGNAL(s2)
6:     end if
7:     SIGNAL(s1)
8: end function
```

- Process is NOT active：
  - Process 呼叫的 function 執行完畢。
  - Process 執行 `wait()` 被 blocked。
- Monitor 解 The dining philosophers problem：

```
Monitor Dining-ph {
    enum {
        thinking, hungry, eating
    } state[5];
}
Condition self[5];
```

Listing 15: Data structure (The dining philosophers problem (Monitor)).

Algorithm 14 *pickup*(i).

```
1: function PICKUP(i)
2:     state[i] := hungry
3:     TEST(i)
4:     if state[i] ≠ eating then
5:         self[i].WAIT
6:     end if
7: end function
```

Algorithm 15 *test*(i).

```
1: function TEST(i)
2:     if state[(i+4) (mod 5)] ≠ eating ∧ state[i] = hungry ∧ state[(i+1) (mod 5)] ≠ eating
   then
3:         state[i] := eating
4:         self[i].SIGNAL
5:     end if
6: end function
```

Algorithm 16 *putdown*(i).

```
1: function PUTDOWN(i)
2:     state[i] := thinking
3:     TEST((i + 4) (mod 5))
4:     TEST((i + 1) (mod 5))
5: end function
```

Algorithm 17 *initialization_code*().

```
1: function INITIALIZATION_CODE                    ▷ For non-Condition type.
2:     for i := 0 to 4 do
3:         state[i] := thinking
4:     end for
5: end function
```

Algorithm 18 P_i (The dining philosophers problem (Monitor)).

```
1: function P_i
2:     DINING_PH dp                                   ▷ Shared variable.
3:     repeat
4:         Hungry.                                    ▷ No active.
5:         dp.PICKUP(i)          ▷ Running: active; Blocked: NOT active.
6:         Eating.                                    ▷ No active.
7:         dp.PUTDOWN(i)                              ▷ Active.
8:         Thinking.                                  ▷ No active.
9:     until False
10: end function
```

- 使用 semaphore 製作 monitor：
  - 共享變數：

```
semaphore mutex = 1;
// Block process P if P call signal.
semaphore next = 0;
// 統計process P 那種特殊waiting processes的個數。
int next_cnt = 0;
// Block process Q if Q call wait.
semaphore x_sem = 0;
// 統計一般waiting processes的個數。
int x_cnt = 0;
```

Listing 16: Shared variables of making monitor using semaphore.

  - 在 function body 前後加入控制碼，類似 Entry section 和 Exit section。

Algorithm 19 f (Example for adding control code before and after function body).

```
1: function F
2:     WAIT(mutex)
3:     Function body.
4:     if next_cnt > 0 then
5:         SIGNAL(next)
6:     else
7:         SIGNAL(mutex)
8:     end if
9: end function
```

Algorithm 20 $x.wait$.

```
1: function x.WAIT
2:     x_cnt := x_cnt + 1
3:     if next_cnt > 0 then
4:         SIGNAL(next)
5:     else
6:         SIGNAL(mutex)
7:     end if
8:     WAIT(x_sem)                          ▷ Q 自己卡住。
9:     x_cnt := x_cnt − 1                   ▷ Q 被救。
10: end function
```

Algorithm 21 $x.signal$.

```
1: function x.SIGNAL
2:     if x_cnt > 0 then
3:         next_cnt := next_cnt + 1
4:         SIGNAL(x_sem)
5:         WAIT(next)                       ▷ P 自己卡住。
6:         next_cnt := next_cnt − 1         ▷ P 被救。
7:     end if
8: end function
```

–

  * Dynamic binding 由 MMU 負責。
  * Dynamic loading 由 programmer 負責，OS 無負擔。
  * Dynamic linking 需要 OS 支持。
  * 必須支援 dynamic binding 才可以在 execution time compaction。

– Process 可分配 frames 數量由 hardware 決定，最多為 physical memory size，最少須讓任一 machine code 完成，即週期中最多可能 memory access 數量，e.g. IF, MEM, WB 共三次。

– Dirty bit：

  * MMU：from 0 to 1.
  * OS：from 1 to 0.

–

$$\text{TLB reach} = \text{TLB entries} \times \text{Frame size} \tag{82}$$

–

    * Solaris ZFS uses **checksums** to provide fault-tolerance in case pointers are wrong.
    * NFS:
        · Using RPC for remote file operations.
        · Writing to a file by a user are immediately visible to other users, since it does **NOT** support session semantics.
        · Does **NOT** support `open()` and `close()` operations.
        · Each request must provide a full set of arguments.
        · Supported file operations must be idempotent.
        · **NO** special measures are needed to recover a server from crash.
-
    * Seek time：head 移到 **track** 的時間。
    * Latency (Rotation) time：**sector** 移到 head 的時間。
- NAS vs SAN：
    * NAS operates at **file** level while SAN operates at **block** level.
    * CIFS/SMB and NFS are examples of NAS.
    * SAN is often the preferred choice over NAS.
    * Almost any machine running Microsoft Windows with LAN connectivity can be configured to access a NAS.
- Log-structured file system：
    * 將要 write 的 data 合成一串，再一次 write。
    * Read 都在 cache，因為 cache 夠大。
    * Disk access 的 seek 和 rotation 是 bottleneck，sequential access 比 random access 好。
-
    * Meltdown: read arbitrary kernel memorya, and it does **NOT** rely on software vulnerabilities.
    * Spectre: Making other applications to access arbitrary contents in memory.
    * Both belongs to **side channel attacks**.
    * Does **NOT** leave records in traditional log.
    * Hard for antivirus software to detect them.

    - Processors which are able to implementat out-of-order execution is risky.
    - IA-64 is immune to Spectre and Meltdown.
  - Power:
    - CMOS does **NOT** consume power when it's **static** ($power_{static} = 0$), so it can decrease **frequency** to save power.
    - **Static** power dissipation occurs because of leakage current that flows even when a transistor is **off**.
    - Computers at **lower utilization** does **NOT** use less power proportionally.
    - The main reason for the switch from high-performance uniprocessors to multiprocessors with simpler cores and lower clock rates in recent years is the **power limit** and **memory gap**.
  - Cache:
    - L1 data cache is usually seperated from L1 instruction cache to **increase bandwidth**.
    - Data cache is usually deployed at **MEM** stage.
    - In modern preocessors, **L1** data and instruction caches are split, but L2 does **NOT**. Both L1 and L2 caches are **write-back**.
    - Physical caches do NOT flush at **context switching**.
    - The TLB cache may require a flush after a page table update.
    - Cache memories are usually hardware controlled, and OS may **NOT** even need to know their existence.

- Branch prediction:
  - Branch target buffer is used by **CPU**, which is checked at **IF** stage.
  - Branch prediction buffer is good to predict the **branch outcome**, but it does **NOT** help in predicting the **branch target**.
  - Indirect branch prediction: Dynamic: hybrid predictor; Static: Neural branch predictor.
  - **Virtual program counter prediction** is often used to predict **conditional/unconditional indirect** branch, which treats indirect branches as **multiple conditional branches**.

- Hazards:
  - **Memory hazard** do NOT cause stall, e.g. `sw` after `lw`.
  - Control hazards can **NOT** be avoided.
- Page table:
  - In hash-based page tables using **linked list** to solve collision, **each element** contains a frame number and a page number.
  - MIPS uses **two** seperated page tables and two limit registers, one for **stack** and the other for **heap**.
  - (**FALSE**) Use of shared memory can reduce the number of page table entries.
  - (**FALSE**) The page table of Linux process is managed by the C runtime library (.so) in the process.
  - For the **unused regions** in the virtual address space, the space overhead of the corresponding **page table entries** can be negligible.
- Arithmetic:
  - Increasing number of **used sticky bits** do NOT improve accuracy.
  - Conversion from single-precision to double-precision causes loss of precision.
  - Ripple Carry Adder: Critical path delay is $2N$ gate delay (carry out), and sum delay is $2N + 1$ gate delay (actual sum).
  - Converting an integer variable to a **single** precision FP number will lose precision, but **double** precision does **NOT**.
- Multi-threading:
  - GPGPU usually runs **SPMT** (Single Program Multiple Thread), and GPU runs SIMT.
  - Vector processors need **less bandwidth** than conventional processors.
  - GPUs do **NOT** rely on **multilevel caches**.
- Storage:
  - Smartphones normally do **NOT** have HDDs.

  - Secondary storage is normally **non-volatile**.
  - Wearable devices are normally equipped with **hard disks** to increase its storage space.

- Disk：
  - **High-level** formatting creates a file system on a disk partition.
  - A disk sector contains a header, a data area, and a trailer.
  - In UNIX, disk scheduling algorithm is performed in the **disk driver**.
  - A file system can be created across **multiple disk partitions**.
  - **Disk device driver** can **NOT** be paged out, but page tables, memory-mapped files, shared memory can.
  - Moving files between directories on the **same** disk partition and **deleting** files on a hard disk cause little overhead, but moving files between directories on **different** disk partitions cause much.
  - The variation of disk I/O **latencies** under SSTF can be very high.

- Cybersecurity：
  - Trojan Horse is a code segment that **misuses** its environment.
  - Installing antivirus software is **NOT** an example of least privileges.
  - Many routers are equipped with **firewall** and **VPN** functions.
  - Via HTTPS, ISPs can know the browsing website, but can **NOT** know the content.

- Cryptography：
  - Public-key (asymmetric) cryptography 提供 digital signature 功能。
  - AES：Symmetric, block cipher.
  - DES：Symmetric, block cipher.
  - RC4：Symmetric, stream cipher.
  - RSA：Asymmetric，只要鑰匙夠長，沒有任何可靠的攻擊方法。
    * Authentication：將 message 與 hash 過再用 private key 加密的 message 串接。e.g. $M||\{h(M)\}_{K_{sa}}$.

    - Confidentiality：將用 one-time AES key 加密的 message 與用 public key 加密的 one-time AES key 串接。e.g. $\{M\}_{K_{da}}||\{K_{da}\}_{K_{pb}}$.
    - Confidentiality and authentication：將 authentication 的內容用 one-time AES key 加密，再與用 public key 加密的 one-time AES key 串接。e.g. $\{M||\{h(M)\}_{K_{sa}}\}_{K_{da}}||\{K_{da}\}_{K_{pb}}$.
  - Digital certificate contains **private key** signed by the user.

- Kernel：
  - Monolithic：UNIX, UNIX-like, Windows 9x, Android.
  - Microkernel：Mach.
  - Hybrid：Windows NT, Windows XP, macOS.
  - Kernel processes are **NOT** allocated through paging and virtual memory interface.
  - A **non-preemptive** kernel is free from race conditions on kernel data structures.
  - **Preemptive** kernel design can **NOT** prevent the deadlock problem with kernel data structures from occurring in the kernel.
  - Linux kernel is a **preemptive** kernel and a process running in a kernel mode could **NOT** be preempted.

- UID：
  - Real UID: identify the real owner of the process and affect the permissions for sending signals.
  - Effective UID: used for most access checks, including creating and accessing to a file.
  - Saved UID: used when a program running with elevated privileges needs to do some unprivileged work temporarily.

- I/O：
  - Buffered I/O: Read one block to cache when R/W, then copy from cache and return to reduce number of system call. Totally 2 copy operations.
  - Unbuffered I/O: Directly transfer from disk without caching. Caching is conducted by the application. Number of copy operations is determined by the transfering method, and it's only 1 copy operation for block-transfering.

  - A program using asynchronous I/O system calls in **NOT** simpler to write than using synchronous I/O system calls.

- File system：
  - devfs：**Virtual** fs。一個 file 一個 device，但該 device 未必存在，**不確定 device mapping**。
  - sysfs：**Virutal** fs。將 real connected devices 組織成**分階層**的 file directory，每個 device 有**唯一**對應的 directory。
  - Device tree：每個 node 用 key 對應 value 方式紀錄 device properties，其中 value 可為空。

- GCD (Grand Central Dispatch)：
  - 自動利用更多 CPU cores。
  - 自動管理 thread life cycles。
  - Move thread pool out of hand of developers and closer to OS.
  - Dispatch tasks 時，可分在相同或不同 queues，分別稱作 serial 和 concurrent。Queues 間可分為 sync 和 async，前者同時間只允許一個 queue 執行，後者允許多個 queues 執行。

- Container：
  - 所有 containers 共用 host OS。
  - 相較 VM，不須打包 OS 就能執行，速度較快且空間小。

- MBR, BIOS, GPT, UEFI：
  - BIOS 無法辨識 GPT（GUID Partition Table）。
  - UEFI 用來定義 OS 和 firmware 間的 software interface。
  - UEFI 是用模組化，動態連結的形式構建的系統，較 BIOS 而言更易於實現，容錯和糾錯特性更強，縮短了系統研發的時間。
  - UEFI（Unified Extensible Firmware Interface）預啟動時就 load OS，且可以同時識別 MBR 和 GPT。
  - GPT 使用 LBA（Logical Block Address）取代早期 CHS（Cylinder-head-sector）定址方式。

  - GPT 的分割區表的位置資訊儲存在 GPT header 中，但第一個磁區仍然用作 MBR，之後才是 GPT header。

- Thread：
  - **Native Windows threads** cause a user-mode to kernel-mode.
  - Hyper-threading is **superscalar** and it can speedup **context switching**.
  - Each thread of the program receives a **larger** CPU time with **many-to-one** thread model.
  - Most operating systems **downgrade** the thread priority when it runs out of time quantum, but **boost** the priority when it returns from an I/O request.
- Allocation：
  - There is **NO** optimum solution to allocate contiguous memory from free holes.
  - Extent allocation uses **contiguous physical** blocks, and it also needs defragmentation.
  - Contiguous allocation offers the best R/W performance for **large** files.
- CPU scheduling：
  - FIFO can outperform LRU.
  - FIFO may have Convoy effect, which causes low **I/O** utilization.
  - After making system calls, the process is still in running state.
  - (**FALSE**) In a time-sharing system, a process does **NOT** leave running state unless it terminates or is preempted through a timer interrupt.
- Synchronization：
  - Test-and-Set still wastes cycles when a process can **NOT** acquire a lock.
  - To use shared memory, several system calls have to be invoked.
  - Test-and-Set can be implementated in **user space**, provided that the lock variable is in a shared memory region.
  - **Two-phase locking protocol (2PL)** ensures **conflict serializability**, but it may result in **deadlock**.

  - OS does **NOT** need to estimate MAX when a process enters ready queue.

- Out-of-order execution in **cache** level do NOT fail.
- Program is a **passive** entity, process is an **avtive** entity.
- Multiple-cycles CPU requires **minimum function units**.
- Compiler identifies **basic blocks** for code optimization.
- To form the machine code, the value of label of branch instructions is computed by **linker** when the label is an **external** reference.
- **NOT** each computer support **direct addressing mode**.
- **Conflict** misses do **NOT** occur in **fully associative** caches.
- **MIPS** and **ARM** use **memory-mapped I/O**.
- Writes are much **slower** than reads for flash. NAND flash is **cheaper** than NOR flash.
- Difficulty to handle **exceptions** (from most difficult to simplest):

Superscalar
Speculative
Out-of-order
Pipelined
Single-issue in-order processor
Hierarchical data caches

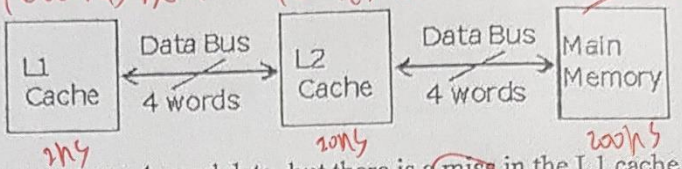
- Difficulty to handle **interrupts** (from most difficult to simplest):

GPGPU
Containers
Virtual machines
Hyper-threaded processor
Superscalar
Pipelined

- Data fault: Access invalid data memory, which is signaled by **MMU**.
- NUMA is intrinsic in Von Neumann's computer model.

- `kmalloc` : physically contiguous; `vmalloc` : virtually contiguous; `malloc` : no constraints.
- `strncpy` 相較 `strcpy` 安全，且需要**預留一格**，可防止 buffer overflow。
- Java **interprets** Java bytecode operations **one at a time**.
- CLR, which is the implementation of .NET VM, **compiles** Microsoft intermediate language instructions **one at a time**.
- Normal instructions for the VM can execute **directly on the hardware** and **only the privileged instructions** must be simulated.
- Named pipes are referred to as **FIFOs** in UNIX systems. Once created, they appear as typical **files** in the file systems.
- Permission bits are stored at **inodes**.
- Five classic components: datapath, control unit, memory, input, and output.
- Data center cares more about **throughput** than response time.
- Memory blocks on the **stacks** can **NOT** be freed at any time, but **heaps** can.
- **Stack** is good for locality.
- (**FALSE**) Programs written in different assmebly languages can ONLY be executed on specific hardware.
- Computer system can be divided into four components including hardware, OS, application programs, and users.
- Normal instructions for the virtual machines can execute directly on the hardware and ONLY the privileged instructions must be simulated.
- Bitmap is NOT a file.
- data section 存 global 和 static variables。
- When the block size is very large, the **spatial locality** within the block is lower.

B. A computer system has an L1 cache, an L2 cache, and a main memory unit connected as shown below. The block size is 16 words for the L1 cache, and is 4 words for the L2 cache; the main memory is 4-word wide. The access times are 2 ns, 20 ns, and 200 ns for the L1 cache, L2 cache, and main memory, respectively.



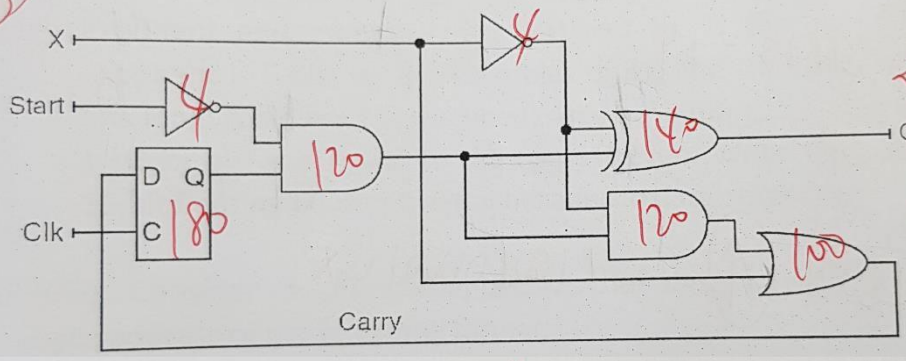
24. When the processor requests some 4-word data, but there is a miss in the L1 cache and a hit in the L2 cache, how much is the total required time for data transfer upon this request?
(a) 20 ns
(b) 22 ns
(c) 80 ns
(d) 82 ns

d

25. When the processor requests some 4-word data, but there is a miss in both of the L1 cache and the L2 cache, and then a hit in the main memory, how much is the total memory access time for this request?
(a) 220 ns
(b) 222 ns
(c) 282 ns
(d) 880 ns

C

10. a. (4%) What is the function for the following circuit diagram?



$Q^+ = X + \bar{X}Q$
$= (X + \bar{X})(X + Q)$
$= X + Q$
$Out = \bar{X} \oplus Q$

10.(a)

X	Q	Out	Q^+
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	1

X	Q	Out
0011	1110	0010
1100	1000	1011
0101	1110	0100
1010	1100	1001

→ decrement by 1

14. (6%) Consider the target application with the following statistics for your team to design a cache:

230 data reads per 1000 instructions; → 0.23

120 data writes per 1000 instructions. → 0.12

With the block size of 128 bytes, the instruction cache miss rate is 0.4%, and the data cache miss rate is 2%. Note that the minimum CPI is 1 (i.e., at most one instruction is fetched per cycle). Assume that each miss generates a request for one block.

a. (3%) The read and write bandwidths between RAM and the cache is 1 byte/cycle in the initial design. For a write-through, write-allocate cache, what is the expected CPI?

b. (3%) Your team wants to improve the CPI to less than 1.5 by increasing the read and write bandwidth between RAM and the cache. However, the bandwidth is restricted to 2^k bytes/cycle, where k is an integer and $k \geq 0$. What is the smallest k to achieve the goal? What is the resultant CPI?

15.(a) $6 \times 10^2 + 3 \times 10 = 630$

→ $1 / \left(\frac{600}{630} / 1\right) + 30/630$

inst. length

$32\text{bit}/1\text{B} = 4$

15.(b) $1 / \left(\frac{600}{630} / \frac{1}{(0.\frac{1}{2})}\right) + 30/630$

data cache read miss

data cache write miss

inst. cache read miss

14.(a) $1 + 1 \times 0.004 \times 128 + 0.23 \times 0.02 \times 128 + 0.12 \times 0.02 \times 128 + 0.12 \times 0.98 \times 4$ → data cache write hit

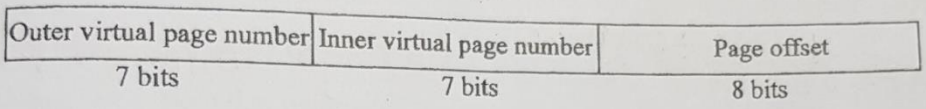
14.(b) $1.5 - 1 > (1 \times 0.004 \times 128 + 0.23 \times 0.02 \times 128 + 0.12 \times 0.02 \times 128 + 0.12 \times 0.98 \times 4) / 2^k$

→ $2^k > 3.76$ → $k = 2$ (min)

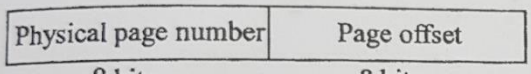
→ New CPI $= 1 + (0.512 + 0.368 + 0.307 + 0.492) / 2^2$

$= 1.47$

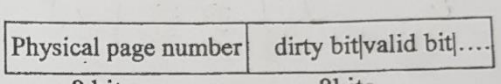
4. (11%) Consider a two-level page table memory management scheme that translates 22-bit virtual addresses to 16-bit physical addresses using page tables with 16-bit table entries. All the address formats are shown below:



22-bit virtual address



16-bit physical address



16-bit page table entry

a. (2%) Explain what the purpose of dirty bit is.

b. (2%) Give a logical reason why the designer might have made the virtual page number field 7 bits each for the inner and outer page tables.

剛好容納一個 page table.

→ page: $2^8 = 256B$ PT entry: 8+8 bits = 2B → $\frac{256B}{2B} = 128 = 2^7$ → 7 bits

9. (3 points) Answer **TRUE** or **FALSE**: Consider the L1 Cache in Intel i7-6700 (Skylake) with 32 KB, 64 B/line and 8-WAY L1 cache, 4 KB pages mode. If Virtual Address Indexed, Physical Address Tagged cache (VIPT) is used, one RAM data may be repeated placed in different cache slots at the same time.

F

~~Cache blocks~~ :

index : $\frac{32KB}{64B \times 8} = 2^6$

offset : $64B = 2^6$

4KB page : 2^{12}

→ $6+6 \leq 12$

→ Virtual address = physical address

→ Impossible to have one RAM data be placed in different caches at the same time.

接次頁