

輕鬆學演算法

從經典演算法到量子演算法

江振瑞

June 15, 2021

第零章 量子演算法

— 從位元到量子位元

Contents

0.1 位元與量子位元	1
0.2 狄拉克記號 (Dirac notation)	4
0.3 量子位元表示法	8
0.4 量子邏輯閘 (quantum logic gate)	8
0.5 Deutsch 演算法	12
0.6 Deutsch–Jozsa 演算法	14

0.1 位元與量子位元

古典計算模式以位元 (bit, or binary digit) 為基礎進行計算。一個位元在邏輯上不是 1 就是 0，在實體上則是以電晶體 (transistor) 的開或關的狀態來表示。現今我們經常使用具有計算能力的設備，如超級電腦、伺服器、桌機、筆電、平板、手機及微處理器都是使用這種計算模式的設備。這些設備通常可以在室溫運作，而且具有極低錯誤率。一般而言，古典計算模式的計算能力與每個步驟能夠處理的位元數呈現線性正比關係。目前人們每天日常工作、教育、娛樂與生活事務處理等都可透過古典計算模式得到良好的處理。

量子計算模式以量子位元 (qubit, qbit, or quantum bit) 為基礎進行計算。一個量子位元在邏輯上是 1 和 0 同時存在，可以被同時處理的疊加 (superposition) 狀態，在實體上則是以雙態 (two-state) 量子力學系統 (quantum mechanical system) 來實現，常見的範例包括使用電子自旋 (electron spin) 的上自旋 (spin up) 及下自旋 (spin down) 或是使用單光子偏振 (photon polarization) 的垂直偏振 (vertical polarization) 和水平偏振 (horizontal polarization) 等方式來實現量子位元又 1 又 0 的疊加狀態。

除了量子疊加狀態的特質之外，量子位元還具有量子糾纏 (entanglement) 狀態的特質。如文獻 [1] 所敘述的，為了使量子位元能夠被運用，量子必須達到量子疊加狀態和量子糾纏狀態，才能做為量子計算模式的基本單元。量子「量子糾纏」這一詞由薛丁格在文獻 [2] 提出，而愛因斯坦稱認為這種特質為「鬼魅般的超距作用 (spooky action at a distance)」，也就是說，如果在量子之間形成糾纏狀態，則量子即使處於相距相當遠的不同空間，卻仍然可以在沒有任何時間差的情況下維持所有量子間固定的特定關係。舉例而言，若有兩個狀態相反的量子處於糾纏狀態中，則若其中第一個量子被觀察到在邏輯上是 1，則另外一個量子在邏輯上一定是 0。反之，若其中第一個量子被觀察到在邏輯上是 0，則另外一個量子在邏輯上一定是 1。

因此，1 個量子位元可以同時呈現 0 與 1 的疊加狀態，2 個量子位元就可以同時呈現 00、01、10 與 11 的疊加狀態、...、 n 個量子位元就能夠同時呈現 2^n 種疊加狀態。所有疊加狀態以隨機的狀態呈現，而且可以透過量子計算系統使用量子邏輯閘同時處理 (計算)，最後再以量測的方式得到每種狀態產生的機率，以機率最高的狀態為最終的計算結果。另一方面，由於 n 個古典位元同一時間只能呈現 2^n 種狀態中的一種，然後這個呈現的狀態可以透過古典計算系統使用布林邏輯閘處理 (計算)，因此需要進行 2^n 次處理才能得知所有 2^n 種狀態的處理結果。綜合而言，使用 n 個量子位元的量子計算系統相較於使用 n 個位元的古典計算系統具有指數量級 ($O(2^n)$) 的計算加速。

目前較著名的量子計算系統包括：加拿大的 D-Wave、美國的 IBM Q System 以及美國的 Google Sycamore 等。如前所述，量子計算模式的計算能力與每個步驟能夠處理的量子位元數呈現指數次方正比關係，因此非常適合處理需要大量計算資源的組合最佳化 (combinatorial optimization) 問題，例如氣象預

報、投資規劃、交通控制、化學製藥、新蛋白合成、行程排班等。然而，這些系統必須在極低甚或接近絕對零度 (-273.15°C 或是 0K) 的溫度下運作，而且很容易由於外在環境的影響產生量子退相干 (quantum decoherence) 現象，進而使系統的量子行為變遷成為古典行為，產生量子至古典變遷 (quantum-to-classical transition)，這使得目前的量子計算系統具有高錯誤率。因此，如何在相對於絕對零度下的較高溫度就能達成量子疊加與糾纏狀態，同時具有較長的退相干時間 (也就是維持量子行為的時間)，並能夠處理高錯誤率的問題就成為先進量子計算系統追求的目標。

圖1是使用古典計算模式的台灣杉二號 (Taiwania 2) 超級電腦，根據維基百科的說明：台灣杉二號是由國家高速網路與計算中心結合廣達、台灣大、華碩等三大國內企業共同建造的超級電腦，硬體規格由 9072 顆 CPU 及 2016 個 GPU 組成，以每秒執行 9 千兆次浮點運算 (9 peta Floating-point Operations Per Second, 9 petaFLOPS) 的速度，在 TOP 500 於 2018 年底發布的世界五百大超級電腦排名中，排名第 20 名。這部超級古典電腦可以非常有效率地執行解決複雜人工智慧問題的演算法。而目前速度最快的古典電腦為日本的富岳 (Fugaku)，如圖2所示。富岳電腦由富士通與日本理化研究所 (Institute of Physical and Chemical Research) 共同建造，使用 ARM A64FX 處理器，具有 7,630,848 核心。富岳電腦的計算效能為 442 petaFLOPS，遠高於第 2 名由 IBM 為美國橡樹嶺國家實驗室 (Oak Ridge National Laboratory, ORNL) 打造的 Summit 電腦的 148.8 petaFLOPS。



圖 1: 台灣杉二號超級古典電腦
(資料來源: <https://www.ithome.com.tw/news/126983>)



圖 2: 世界最快的超級古典電腦-富岳 (Fugaku)
(資料來源: <https://news.yahoo.co.jp/articles/50b42cc2e9bc6b5f559f446b9f82500857cbf7ad>)

圖3是使用量子計算模式的 D-WAVE 2000Q 量子電腦，擁有 2000 量子位元，而目前最新的 D-WAVE 量子電腦擁有 5000 量子位元。D-WAVE 量子電腦雖然只能執行量子退火 (quantum annealing) 演算法，還不是通用型量子電腦 (universal quantum computer)，但是已經比現在全世界最高速的超級電腦能更快的找出許多組合最佳化問題的最佳解答。另一方面，IBM Q(如圖4) 以及 Google Sycamore(如圖5) 量子電腦則是朝通用型量子電腦的方向發展，目標可以執行一般的量子演算法，例如，解決質因數分解問題的秀爾演算法 (Shor's algorithm) [3] 以及解決搜尋問題的格羅弗演算法 (Grover's algorithm) [4] 等演算法。

本章將介紹 Deutsch-Jozsa 演算法，由這個演算法，讀者可以很容易看出量子計算模式比古典計算模式來得更有效率。為了這說明這個演算法，以下我們介紹相關的數學知識，包括狄拉克記號 (Dirac notation)、量子位元表示法、量子邏輯閘 (quantum logic gate) 以及么正矩陣 (unitary matrix) 等概念。



圖 3: D-Wave 2000Q 量子電腦
(資料來源: <https://www.dwavesys.com>)



圖 4: IBM-Q 量子電腦
(資料來源: <https://ml2quantum.com/q-experience/>)



圖 5: Google Sycamore 量子電腦

(資料來源: <https://www.theverge.com/2019/9/23/20879485/google-quantum-supremacy-qubits-nasa>)

0.2 狄拉克記號 (Dirac notation)

狄拉克記號 (Dirac notation) 由狄拉克於 1939 年提出，是複數希爾伯特空間 (complex Hilbert space) 的向量表示法。狄拉克將括號 (bracket) 這個字拆成包量或左向量 (bra) $\langle\psi|$ 與括量或右向量 (ket) $|\psi\rangle$ ，可以用來將量子態描述為希爾伯特空間中的向量。希爾伯特空間是有限維度歐幾里得向量空間的拓展，由有限維度拓展到無限且連續維度，而向量方面則由常實數向量拓展到複數函數向量，非常適合用在量子力學中，不過因為篇幅的關係，在本節並無再進一步的介紹。以下，本節先介紹右向量及左向量記號，然後再介紹一些相關的操作定義，如內積、共軛轉置、範、以及外積等。

右向量 (ket) 也稱為括量或右矢，可以表示希爾伯特空間的行向量 (column vector)，定義如下：

定義 0.1. 右向量 (ket):

$$|\psi\rangle = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}$$

其中 n 為維度 (dimension)， $\psi_1, \psi_2, \dots, \psi_n \in \mathbb{C}$ 。

右向量 (ket) 也可以表示為 $(\psi_1, \psi_2, \dots, \psi_n)^T$ ，其中 $\psi_1, \psi_2, \dots, \psi_n$ 是複數，也可以是實數 (複數中虛部為 0)。以下是右向量 ket 的範例：

$$|\psi\rangle = (1, 2, 3, 4)^T = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

$$|\psi\rangle = (1 + 2i, 3 - 4i, 5 + 6i)^T = \begin{pmatrix} 1 + 2i \\ 3 - 4i \\ 5 + 6i \end{pmatrix}$$

$$|\psi\rangle = (1, -2i, 3, 4i)^T = \begin{pmatrix} 1 \\ -2i \\ 3 \\ 4i \end{pmatrix}$$

另一方面，左向量 (bra) 也稱為包量或左矢，可以用來表示希爾伯特空間的列向量 (row vector)，定義如下：

定義 **0.2**. 左向量 (bra):

$$\langle\psi| = (\psi_1, \psi_2, \dots, \psi_n)^* = (\psi_1^*, \psi_2^*, \dots, \psi_n^*)$$

其中 n 為維度 (dimension), $\psi_1, \psi_2, \dots, \psi_n \in \mathbb{C}$, 星號代表共軛 (conjugate) 複數操作。

以下是左向量 bra 的範例:

$$\langle\psi| = (1, 2, 3, 4)^* = (1, 2, 3, 4)$$

$$\langle\psi| = (1 + 2i, 3 - 4i, 5 + 6i)^* = (1 - 2i, 3 + 4i, 5 - 6i)$$

$$\langle\psi| = (1, -2i, 3, 4i)^* = (1, 2i, 3, -4i)$$

左向量 (bra) 稱為右向量 (ket) 的伴隨向量 (co-vector)，而左向量與右向量的關係如下：

$$|\psi\rangle^\dagger = \langle\psi|$$

或是

$$\langle\psi|^\dagger = |\psi\rangle$$

其中 \dagger 代表共軛轉置 (conjugate transpose)，也稱為赫米特共軛 (Hermitian conjugate)(註: \dagger 為 dagger 符號，隨身匕首之意)。

以下是更具體的左向量與右向量的關係描述:

$$|\psi\rangle^\dagger = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}^\dagger = (\psi_1^*, \psi_2^*, \dots, \psi_n^*) = \langle\psi|$$

或是

$$\langle\psi|^\dagger = (\psi_1^*, \psi_2^*, \dots, \psi_n^*)^\dagger = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} = |\psi\rangle$$

以下是右向量 (ket) 及其對應的左向量 (bra)(或對應的伴隨向量) 的範例:

$$\begin{array}{llll} \text{右向量 (ket):} & \begin{pmatrix} 1+i \\ 1-i \end{pmatrix} & \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ \text{左向量 (bra):} & (1-i, 1+i) & \frac{1}{\sqrt{2}}(1, 0) & (1, 0, 0, 0) \end{array}$$

以下我們以狄拉克記號的型式, 定義向量的內積 (inner product)、範 (norm) 與外積 (outer product)。

定義 **0.3.** 內積 (inner product): 給定兩個右向量 $|\psi\rangle = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}$ 與 $|\phi\rangle = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{pmatrix}$, 則此二向量的內積定義為: $\langle\psi|\phi\rangle = |\psi\rangle^\dagger|\phi\rangle = \psi_1^*\phi_1 + \psi_1^*\phi_1 + \cdots + \psi_n^*\phi_n$

內積的詳細推導如下所示:

$$\begin{aligned} \langle\psi|\phi\rangle &= |\psi\rangle^\dagger|\phi\rangle = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}^\dagger \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{pmatrix} = (\psi_1, \psi_2, \dots, \psi_n)^* \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{pmatrix} = (\psi_1^*, \psi_2^*, \dots, \psi_n^*) \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{pmatrix} \\ &= \psi_1^*\phi_1 + \psi_1^*\phi_1 + \cdots + \psi_n^*\phi_n \end{aligned}$$

內積的計算結果是一個純量 (scalar), 因此也稱為純量積 (scalar product), 或稱為點積 (dot product)。以下是一個內積的範例:

令 $|\psi\rangle = (1 + 2i, 3 - 4i, 5 + 6i)^T = \begin{pmatrix} 1 + 2i \\ 3 - 4i \\ 5 + 6i \end{pmatrix}$, 且令 $|\phi\rangle = (1, -2i, 3)^T = \begin{pmatrix} 1 \\ -2i \\ 3 \end{pmatrix}$, 則 $|\psi\rangle$ 與 $|\phi\rangle$ 的內積

$\langle\psi|\phi\rangle$ 的計算如下:

$$\begin{aligned} \langle\psi|\phi\rangle &= |\psi\rangle^\dagger|\phi\rangle = (1 + 2i, 3 - 4i, 5 + 6i)^* \begin{pmatrix} 1 \\ -2i \\ 3 \end{pmatrix} = (1 - 2i, 3 + 4i, 5 - 6i) \begin{pmatrix} 1 \\ -2i \\ 3 \end{pmatrix} \\ &= (1 - 2i)(1 + 0i) + (3 + 4i)(0 - 2i) + (5 - 6i)(3 + 0i) = 1 - 8i^2 + 15 = 1 + 8 + 15 = 24 \end{aligned}$$

定義 **0.4.** 範 (norm):

一個右向量 $|\psi\rangle = (\psi_1, \psi_2, \dots, \psi_n)^T$ 的範 (norm) 定義為: $\| |\psi\rangle \| = \sqrt{\psi_1^*\psi_1 + \psi_1^*\psi_1 + \cdots + \psi_n^*\psi_n}$

範也稱為模 (modulus), 一般而言, 我們使用範來代表一個右向量的大小或強度 (magnitude)。實際上, 一個右向量的範的平方等於該右向量與自己的內積, 如下所示:

$$\| |\psi\rangle \|^2 = \langle\psi|\psi\rangle = |\psi\rangle^\dagger|\psi\rangle$$

$$\begin{aligned} &= \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}^\dagger \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} = (\psi_1, \psi_2, \dots, \psi_n)^* \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} = (\psi_1^*, \psi_2^*, \dots, \psi_n^*) \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \\ &= \psi_1^*\psi_1 + \psi_1^*\psi_1 + \cdots + \psi_n^*\psi_n \end{aligned}$$

換句話說, 一個右向量 $|\psi\rangle$ 的範可以所示為: $\| |\psi\rangle \| = \sqrt{\langle\psi|\psi\rangle}$

定義 **0.5.** 外積 (outer product): 給定兩個右向量 $|\psi\rangle = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}$ 與 $|\phi\rangle = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{pmatrix}$, 則此二向量的外積定義

為:

$$|\psi\rangle\langle\phi| = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{pmatrix}^\dagger = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} (\phi_1^*, \phi_2^*, \dots, \phi_n^*) = \begin{pmatrix} \psi_1\phi_1^* & \psi_1\phi_2^* & \cdots & \psi_1\phi_n^* \\ \psi_2\phi_1^* & \psi_2\phi_2^* & \cdots & \psi_2\phi_n^* \\ \vdots & \vdots & \ddots & \vdots \\ \psi_n\phi_1^* & \psi_n\phi_2^* & \cdots & \psi_n\phi_n^* \end{pmatrix}$$

依照上述的定義兩個右向量的外積為一個矩陣，舉例而言，若二個右向量 $|\psi\rangle$ 與 $|\phi\rangle$ 都是 n 維向量，則此二向量的外積 $|\psi\rangle\langle\phi|$ 是一個 $n \times n$ 矩陣。若右向量 $|\psi\rangle$ 是 m 維向量而 $|\phi\rangle$ 是 n 維向量，則此二向量的外積 $|\psi\rangle\langle\phi|$ 是一個 $m \times n$ 矩陣。

以下是一個外積的範例：令 $|\psi\rangle = \begin{pmatrix} 1+2i \\ 3-4i \\ 5+6i \end{pmatrix}$ ，且令 $|\phi\rangle = \begin{pmatrix} 7+8i \\ 9-i \end{pmatrix}$ ，則 $|\psi\rangle$ 與 $|\phi\rangle$ 的外積 $|\psi\rangle\langle\phi|$ 的計算如下：

$$|\psi\rangle\langle\phi| = |\psi\rangle|\phi\rangle^\dagger = \begin{pmatrix} 1+2i \\ 3-4i \\ 5+6i \end{pmatrix} (7-8i, 9+i) = \begin{pmatrix} 7-16i^2 & 9+2i \\ 21+36i^2 & 27-4i^2 \\ 35-48i^2 & 45+6i^2 \end{pmatrix} = \begin{pmatrix} 23 & 7 \\ -15 & 31 \\ 83 & 39 \end{pmatrix}$$

以下介紹張量積 (tensor product)，在介紹張量積之前，我們先介紹張量 (tensor)。所謂張量可以包含純量 (scalar)、向量 (vector)、及矩陣 (matrix)。具體地說，純量為零階張量，向量為一階張量，而矩陣為高階張量。以下以矩陣形式描述張量乘積的定義：

定義 0.6. 張量積 (tensor product):

令 $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$ 為 $m \times n$ 矩陣， $B = \begin{pmatrix} b_{11} & \cdots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \cdots & b_{pq} \end{pmatrix}$ 為 $p \times q$ 矩陣，則 A 與 B 的張量積 $A \otimes B$ 定義為：

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & \cdots & a_{11}b_{1q} & \cdots & a_{1n}b_{11} & \cdots & a_{1n}b_{1q} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & \cdots & a_{11}b_{pq} & \cdots & a_{1n}b_{p1} & \cdots & a_{1n}b_{pq} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} & \cdots & a_{m1}b_{1q} & \cdots & a_{mn}b_{11} & \cdots & a_{mn}b_{1q} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & \cdots & a_{m1}b_{pq} & \cdots & a_{mn}b_{p1} & \cdots & a_{mn}b_{pq} \end{pmatrix}$$

以下舉兩個張量積的範例：

$$|0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & 0 \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & 1 \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

0.3 量子位元表示法

量子位元有許多表示方法，包括單量子位元的布洛赫球表示法 (Bloch sphere) 以及以波的方式表示量子位元的方法。本節介紹使用狄拉克記號表示量子位元的方法，先介紹單位元的表示法，然後再介紹多位元的表示法。

單量子位元對應一個處於疊加狀態的量子，量子具有波粒二象性 (wave-particle duality)，在疊加狀態時量子處於波動現象，此時量子同時是'0' 狀態也是'1' 狀態。任何時刻，我們無法得知量子的狀態直到我們進行測量為止。但是當我們進行測量時，量子的波動現象受到干擾而呈現粒子現象，量子的狀態坍縮 (collapse) 為'0' 狀態或是'1' 狀態。量測時量子出現'0' 狀態或是'1' 狀態的機率可能相同或不同，針對在疊加狀態時的量子進行操作會影響量測時出現'0' 狀態或是'1' 狀態的機率，很顯然的，量測時量子出現'0' 狀態或是'1' 狀態的機率的總和必定為 100%。

一個單一的量子狀態可以使用狄拉克記號描述為：

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

在以上的式子中， $\alpha, \beta \in \mathbb{C}$ 是兩個複數係數，用來表現量子疊加狀態的週期波函數。在進行量子測量時，量測到 $|0\rangle$ 的機率為 $|\alpha|^2$ ，而量測到 $|1\rangle$ 的機率為 $|\beta|^2$ ，很顯然的， $|\alpha|^2 + |\beta|^2 = 1$

另外，上式中的 $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ 與 $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 是希爾伯特空間的向量，構成正交基底 (orthogonal basis) 或正交範基底 (orthonormal basis)。一組正交基底的複數係數線性組合可以表示所有可能的疊加狀態。

除了 $|0\rangle$ 與 $|1\rangle$ 這組標準的正交基底之外，常見的正交基底還有加 (plus) 與減 (minus) 基底，如下所示：

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

$$|-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

還有一組常用的正交基底為順時針 (clockwise) 與逆時針 (counter-clockwise) 基底，如下所示：

$$|\odot\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle + i|1\rangle$$

$$|\oslash\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle - i|1\rangle$$

以下針對 1 個量子位元、2 個量子位元以及 n 個量子位元的標準基底，以狄拉克記號表示：

□ 單量子位元：

狄拉克記號表示法 (二進位)： $|0\rangle$ 、 $|1\rangle$

狄拉克記號表示法 (十進位)： $|0\rangle$ 、 $|1\rangle$

□ 雙量子位元：

狄拉克記號表示法 (二進位)： $|00\rangle$ 、 $|01\rangle$ 、 $|10\rangle$ 、 $|11\rangle$

狄拉克記號表示法 (十進位)： $|0\rangle$ 、 $|1\rangle$ 、 $|2\rangle$ 、 $|3\rangle$

□ n 量子位元：

狄拉克記號表示法 (二進位)： $|00 \cdots 00\rangle$ 、 $|00 \cdots 01\rangle$ 、 $|00 \cdots 10\rangle \cdots |00 \cdots 11\rangle$

狄拉克記號表示法 (十進位)： $|0\rangle$ 、 $|1\rangle$ 、 $|2\rangle$ 、 $|3\rangle$ 、 \cdots 、 $|2^n - 1\rangle$

0.4 量子邏輯閘 (quantum logic gate)

現今有許多不同的量子計算模型，如量子電路 (quantum circuit) 計算模型及量子退火 (quantum annealing) 計算模型等，各有各的優缺點及適合的應用領域。本節則專注於介紹使用於量子電路計算模型的量子閘 (quantum gate) 或量子邏輯閘 (quantum logic gate) 的概念。而量子邏輯閘一般使用么正矩陣 (unitary matrix) 來表示，因此本節一開始就先介紹么正矩陣的概念。

么正矩陣又稱為酉矩陣，是一個複數矩陣，由實數空間的正交矩陣 (orthogonal matrix) 和正交範矩陣 (orthonormal matrix) 推廣而來的。以下是么正矩陣的定義：

定義 0.7. 么正矩陣 (unitary matrix):

一個複數矩陣 U 為么正矩陣，若且唯若

$$U^\dagger U = U U^\dagger = I$$

其中 U^\dagger 代表 U 的共軛轉置矩陣，而 I 為單位矩陣 (identity matrix)。

根據以上的定義，么正矩陣亦必定可逆，且其逆矩陣等於其共軛轉置。也就是說：

$$U^{-1} = U^\dagger$$

以下是一個么正矩陣的範例：

$$U = \begin{pmatrix} -\frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

驗證如下：

$$U^\dagger U = \begin{pmatrix} \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} -\frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$U U^\dagger = \begin{pmatrix} -\frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{i}{\sqrt{2}} & -\frac{i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

就像常見的傳統古典邏輯閘，如且閘 (and gate)、或閘 (or gate) 及反閘 (not gate)，是針對一個或兩個位元進行操作，常見的量子閘也是針對一個、兩個或少量幾個量子位元進行操作。大多數的古典邏輯閘不可逆，其輸入與輸出的位元數通常不同；而量子閘可逆，因此其輸入與輸出的量子位元數一定相同。如前所述，量子閘可以使用么正矩陣表示，么正矩陣為可逆的。操作 K 個量子位元的量子閘可以用 $2^k \times 2^k$ 的么正矩陣表示。一個量子閘輸入跟輸出的量子位元數量必須相等，而其運算操作可以透過代表量子閘的么正矩陣與代表量子位元狀態的向量的乘積來表示。

圖6顯示常見的量子閘、其對應的電路圖形表示以及對應的么正矩陣：

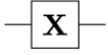




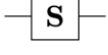
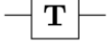
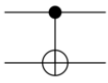
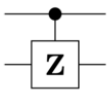



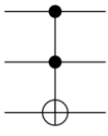
Operator	Gate(s)	Matrix
Pauli-X (X)	 	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)	 	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP	 	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

圖 6: 常見量子閘、其對應的電路圖形以及對應的么正矩陣

(資料來源: https://en.wikipedia.org/wiki/Quantum_logic_gate#/media/File:Quantum_Logic_Gates.png)

以下詳細介紹幾個圖6中提到的量子閘，包括包利-X 閘 (Pauli-X gate)、哈達馬閘 (Hadamard gate) 以及受控反閘 (controlled not gate)。

包利-X 閘簡稱 X 閘 (X gate)，相當於經典的邏輯反閘，有時也被稱為位元翻轉 (bit-flip) 閘。X 閘針對一個量子位元進行操作，映射 $|0\rangle$ 至 $|1\rangle$ 並且映射 $|1\rangle$ 至 $|0\rangle$ 。X 閘的么正矩陣如下所示：

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

一個處於疊加狀態的量子位元 $\alpha|0\rangle + \beta|1\rangle$ ，在經過 X 閘操作後變成 $\beta|0\rangle + \alpha|1\rangle$ ，驗證如下：

$$X \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

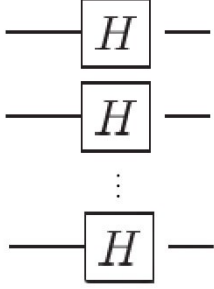
哈達馬閘 (Hadamard gate) 簡稱 H 閘 (H gate)。H 閘也針對一個量子位元進行操作，將基本基底 $|0\rangle$

轉變為 $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ ，並且將基本基底 $|1\rangle$ 轉變為 $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ 。另一方面，H 閘也可將 $|+\rangle$ 轉變回 $|0\rangle$ ，將 $|-\rangle$ 轉變回 $|1\rangle$ 。

H 閘的么正矩陣如下所示：

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

H 閘可以用來產生量子的疊加狀態，其作法如下：假設有 n 個量子位元，則只要在每個量子位元加上一個 H 閘門就可以產生量子的疊加狀態，如下圖所示：



可以透過矩陣張量積的形式來表示如下：

$$H \otimes \cdots \otimes H = \bigotimes_1^n H = H^{\otimes n}$$

以雙量子位元 $|00\rangle$ 為例，我們可得：

$$\begin{aligned} H \otimes H |00\rangle &= H^{\otimes 2} |00\rangle = \left(\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \right) \otimes \left(\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \right) \\ &= \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) \end{aligned}$$

受控反閘 (controlled not gate) 簡稱 CNOT 閘 (CNOT gate)。CNOT 閘有二個輸入與二個輸出位元，其中一個位元為控制 (control) 位元，另一個為目標 (target) 位元。當控制位元為 0 時，不對目標位元進行任何操作；而當控制位元為 1 時，則針對目標位元進行反轉操作。稱 CNOT 閘的么正矩陣如下所列：

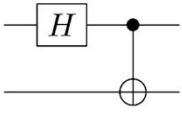
$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

CNOT 閘維持 $|00\rangle$ 與 $|01\rangle$ 、 $|10\rangle$ 轉變為 $|11\rangle$ ，將 $|11\rangle$ 轉變為 $|10\rangle$ ，類似於古典的互斥或 (exclusive OR, XOR) 閘維持 00、01，但是將 10 變 11、11 變 10 的行為。

以下為 CNOT 閘運算的例子：

$$\text{CNOT}|10\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle$$

H 閘搭配 CNOT 閘共同使用可以產生量子糾纏態，其作法如下圖所示：



其做法為先取得控制位元 (假設為 $|0\rangle$) 的疊加態，我們可得：

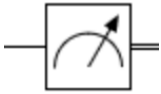
$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \left(\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}} \right)$$

接著套用 CNOT 關於目標位元上 (假設為 $|1\rangle$)，計算結果如下：

$$\begin{aligned} \text{CNOT}(H|0\rangle \otimes |1\rangle) &= \text{CNOT} \left(\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \end{aligned}$$

$\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$ 是典型的雙量子位元糾纏態，表示輸出的兩個位元必定相反。假設相隔很遠的 Alice 與 Bob 分別獨立地測量二個位元中的一個，若 Alice 先測量，有 50% 的機率測得 $|0\rangle$ ，也有 50% 的機率測得 $|1\rangle$ 。但是當 Alice 測得 $|0\rangle$ 之後，則 Bob 以 100% 的機率測得 $|1\rangle$ ；反之，當 Alice 測得 $|1\rangle$ 之後，則 Bob 以 100% 的機率測得 $|0\rangle$ 。

除了量子邏輯閘之外，量子電路也使用到量子測量 (quantum measurement)，在量子電路最後階段用以產生輸出。其電路表示圖形如下所示：



量子測量電路中左方的單線代表量子位元，而右方的雙線則古典位元。

0.5 Deutsch 演算法

Deutsch 演算法由英國牛津大學 David Deutsch 教授在 1985 年提出 [5]，顯示量子演算法可以利用量子疊加態的特性，相對於古典演算法具有類似平行計算的能力，可以說是開啓量子演算法的研究始祖。

以下為 Deutsch 演算法的問題描述：

定義 0.8. Deutsch 演算法問題：

給定黑箱函數 (blackbox function) f ：

$$f : \{0, 1\} \rightarrow \{0, 1\}$$

求 $f(0) \oplus f(1)$ ，其中 \oplus 表示”互斥或”操作。

我們首先說明，黑箱函數是未知函數 (unknown function)，也稱為神諭 (oracle)。我們不知道黑箱函數究竟進行什麼計算，也無從得知它如何計算，但是我們可以詢問 (query) 神諭，也就是向它提供輸入並接收它計算之後的輸出。當然，在古典計算模式下，這個黑箱函數也使用古典計算模式；反之，在量子計算模式下，這個黑箱函數也使用量子計算模式。

在古典計算模式下，需要詢問 f 函數兩次，一次求 $f(0)$ ，一次求 $f(1)$ ，再執行互斥或操作就可得到答案。Deutsch 在量子計算模式下，設計一個如圖7所示的量子電路 (quantum circuit)，只需要詢問 f 函數一次就可以得到答案，這個電路就是 Deutsch 演算法。

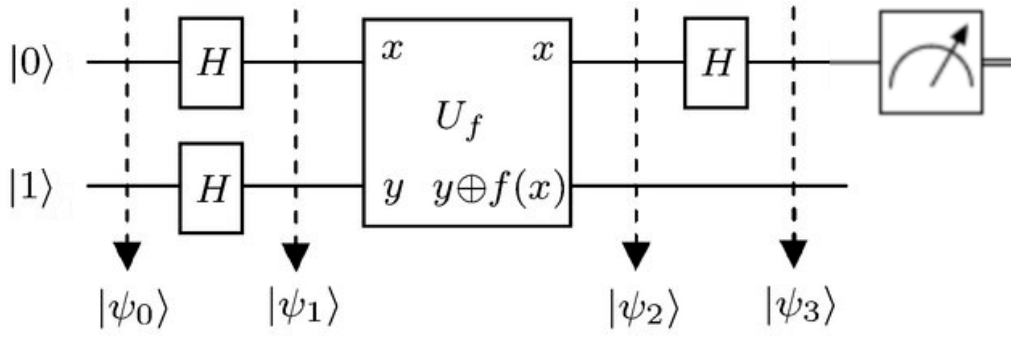


圖 7: Deutsch 演算法量子電路

以下詳細說明 Deutsch 演算法，也就是圖7中的量子電路。在量子電路， U_f 是量子計算模式的黑箱函數 f ，輸入 x 和 y ，輸出 x 和 $y \oplus f(x)$ 。整個電路使用 3 個 H 閘 (Hadamard 閘)，如前所述， H 閘作用在量子位元 $|0\rangle$ 與 $|1\rangle$ 之上，可以使量子位元具有疊加性，我們有以下的結果：

$$H|0\rangle = |+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$$

$$H|1\rangle = |-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$$

$$H|+\rangle = |0\rangle$$

$$H|-\rangle = |1\rangle$$

電路中的 $|\psi_0\rangle$ 、 $|\psi_1\rangle$ 、 $|\psi_2\rangle$ 、 $|\psi_3\rangle$ 代表電路中不同階段的中間結果值，我們有 $|\psi_0\rangle = |0\rangle|1\rangle$

以下，我們計算 $|\psi_1\rangle$ 、 $|\psi_2\rangle$ 、 $|\psi_3\rangle$ ，然後再討論測量結果和 f 的關係。

如前所述， $H|0\rangle = |+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ ， $H|1\rangle = |-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ ，所以

$$|\psi_1\rangle = (H|0\rangle)(H|1\rangle) = (|0\rangle + |1\rangle)(|0\rangle - |1\rangle)/2.$$

以下計算 $|\psi_2\rangle$ ，在計算之前我們先說明 U_f 。 U_f 是量子計算模式的黑箱函數 f ， $U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$ 。假設它第一個輸入值是 $|x\rangle$ ，第二個輸入值 $|y\rangle$ 直接代入 $(|0\rangle - |1\rangle)/\sqrt{2}$ 。則我們可得：

$$\begin{aligned} |\psi_1\rangle &= U_f|x\rangle(|0\rangle - |1\rangle)/\sqrt{2} = |x\rangle(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle)/\sqrt{2} \\ &= |x\rangle(|f(x)\rangle - |1 \oplus f(x)\rangle)/\sqrt{2} \quad \text{註: } 0 \oplus f(x) = f(x) \\ &= \begin{cases} |x\rangle(|0\rangle - |1\rangle)/\sqrt{2}, & \text{if } f(x) = 0, \\ |x\rangle(|1\rangle - |0\rangle)/\sqrt{2}, & \text{if } f(x) = 1 \end{cases} \\ &= (-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle)/\sqrt{2}. \end{aligned}$$

之後代入 $x = (|0\rangle + |1\rangle)/\sqrt{2}$ ，我們可得：

$$\begin{aligned} |\psi_2\rangle &= U_f(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)/2 \\ &= [(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle] (|0\rangle - |1\rangle)/2 \\ &= (-1)^{f(0)} [|0\rangle + (-1)^{f(0) \oplus f(1)}|1\rangle] (|0\rangle - |1\rangle)/2. \end{aligned}$$

換個形式表示可得：

$$|\psi_2\rangle = \begin{cases} (-1)^{f(0)}|+\rangle|-\rangle, & \text{if } f(0) \oplus f(1) = 0, \\ (-1)^{f(0)}|-\rangle|-\rangle, & \text{if } f(0) \oplus f(1) = 1. \end{cases}$$

最後我們計算 $|\psi_3\rangle$ ，因為 Deutsch 量子電路中只量測低位元 (x 位元)，因此以下 $|\psi_3\rangle$ 指的是 $|\psi_2\rangle$ 的

低位元，也就是 $|\psi_3\rangle = H|\psi_2\rangle$ 。

如前所述，H 閘可以將 $|0\rangle$ 及 $|1\rangle$ 轉變為 $|+\rangle$ 及 $|-\rangle$ ，也可以將 $|+\rangle$ 及 $|-\rangle$ 轉變回 $|0\rangle$ 及 $|1\rangle$ ，我們可得以下結果：

$$|\psi_3\rangle = \begin{cases} (-1)^{f(0)}|0\rangle, & \text{if } f(0) \oplus f(1) = 0, \\ (-1)^{f(0)}|1\rangle, & \text{if } f(0) \oplus f(1) = 1. \end{cases}$$

最後，我們只需要針對低位元測量 $|0\rangle$ ，若量測出 $|0\rangle$ 的機率是 1，就可推導出 $f(0) \oplus f(1) = 0$ ，反之就可推導出 $f(0) \oplus f(1) = 1$ 。

上述的 Deutsch 量子演算法 (量子電路) 只需要詢問 U_f 一次就可以得知 $f(0) \oplus f(1)$ 的結果，這是由於 Deutsch 量子演算法利用量子疊加性質，以類似平行計算的方式得到 $f(0)$ 與 $f(1)$ 的結果。雖然 Deutsch 量子演算法可以顯示量子演算法相對於古典演算法具有類似平行計算的優越性，但是因為 Deutsch 量子演算法的輸入只有一個位元，因此較難看出實際計算上的加速。下節中將介紹 Deutsch-Jozsa 量子演算法，這個演算法是 Deutsch 量子演算法的延伸版本，具有 n 個輸入位元，可以很容易讓讀者看出量子演算法相較於古典演算法具有指數量級的加速優越性質。

0.6 Deutsch—Jozsa 演算法

Deutsch—Jozsa 演算法 (以下簡稱 DJ 演算法) 由英國牛津大學 David Deutsch 教授與劍橋大學 Richard Jozsa 教授在 1992 年提出 [6]，可以用以顯示量子演算法相對於古典演算法具有指數級別的加速能力。以下為 DJ 演算法的問題描述：

定義 0.9. Deutsch—Jozsa 演算法問題：
給定黑箱函數 (blackbox function) f ：

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

判斷函數 f 是常數 (constant) 函數或是平衡 (balanced) 函數。

f 函數有以下兩種可能的類型：

f 是常數的 (constant)，也就是針對輸入的 $x, x \in \{0, 1\}^n$ ，都得到 $f(x) = 0$ 或 $f(x) = 1$ 。

f 是平衡的 (balanced)，也就是針對輸入的 $x, x \in \{0, 1\}^n$ ， $f(x)$ 輸出 0 和 1 的次數相同。

DJ 演算法的目標為判斷函數 f 是常數的還是平衡的。

以下先考慮使用古典計算模式的情況：

因為 f 函數的輸入有 n 個位元，因此總共有 2^n 種可能的輸入。在古典計算模式下，演算法必須一一測試每一個可能的輸入來判斷 f 函數屬於什麼類型。在最佳狀況下，最少需要測試 2 次輸入才能判斷 f 函數屬於什麼類型，這發生在第 1 個輸入與第 2 個輸入讓 f 函數產生不同輸出的時候，此時可以立即判斷 f 函數是平衡的。但是在最差狀況下，則需要測試 $2^{n-1} + 1$ 次輸入才能確認 f 函數屬於什麼類型，這發生在做完一半 (也就是 2^{n-1} 次) 可能輸入的測試時，若這一半的輸入都使 f 函數產生相同的輸出，則此時需要再測試額外 1 個輸入才能判斷 f 函數屬於什麼類型。綜合而言，古典計算模式 DJ 演算法的最佳狀況時間複雜度為 $O(1)$ ，而其最差狀況時間複雜度為 $O(2^n)$ 。

使用量子計算模式的情況：

通過建構如圖 8 的 DJ 演算法量子電路 (circuit)，則演算法僅需運行一次就能確定函數的類型，再加上讓 n 個位元產生疊加狀態的運算，量子計算模式 DJ 演算法在所有狀況下的時間複雜度為 $O(n)$ ，這相對於古典計算模式 DJ 演算法具有指數量級的加速。

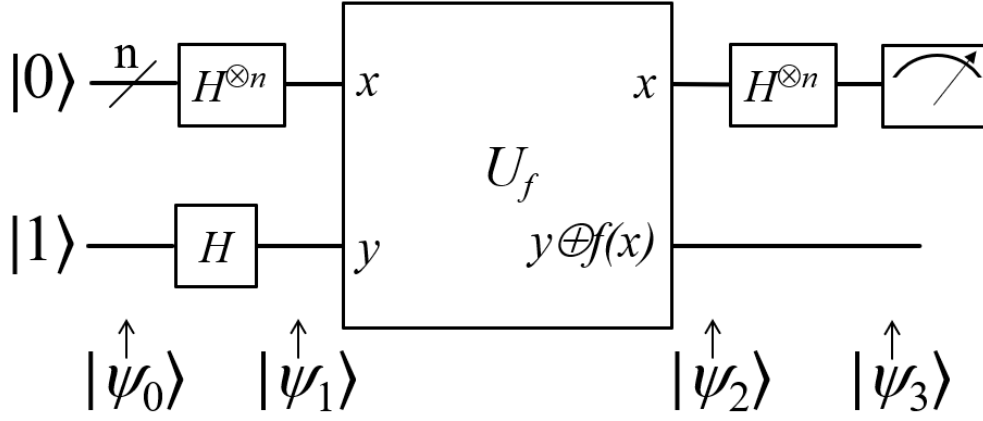


圖 8: Deutsch-Jozsa 演算法量子電路 (quantum circuit)

(資料來源: By Peplm: <https://commons.wikimedia.org/w/index.php?curid=75740173>)

以下說明建構 DJ 演算法量子電路的方式，因為量子電路是逆的，因此對應一個么正矩陣，而且必須具有相同數量的輸入於輸出位元。所以，除了原先的輸入位元之外，還要加上一個額外的位元才可以滿足這個要求。具體的說，DJ 演算法量子電路具有 n 個工作 (working) 位元與一個輔助 (ancillary) 位元。

以下為 DJ 演算法量子電路的建構步驟：

步驟 1: 準備 n 個工作 (working) 位元為 $|0\rangle$ 狀態 (也就是 $|0\rangle^{\otimes n}$ ，與一個輔助 (ancillary) 位元為 $|1\rangle$ 狀態。

步驟 2: 讓所有位元都經過 Hadamard 閘，使位元處於疊加狀態，如下式所示。

$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H^{\otimes n+1}} \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle(|0\rangle - |1\rangle)$$

步驟 3: 系統通過 Oracle 的么正變換，將輸入 $|x\rangle|y\rangle$ 轉變為 $|x\rangle|y \oplus f(x)\rangle$ ，其中 \oplus 為模 2 加法 (addition modulo 2)，也就是：

$$U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$$

U_f 的輸出為：

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \left[\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle \right] = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \left[\sum_{x=0}^{2^n-1} (-1)^{f(x)} (-1)^{x \cdot y} \right] |y\rangle$$

其中， $x \cdot y = x_0 y_0 \oplus x_1 y_1 \oplus \cdots \oplus x_{n-1} y_{n-1}$ 。

步驟 4: 去除輔助位元，執行量子測量。如果輸出全部為 0 的機率為 1，則 f 是常數函數；反之， f 是平衡函數。具體的說，也就是量測得 $|0\rangle^{\otimes n}$ 的機率為 $\left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2$ ，若量測出的機率為 1，則 f 是常數函數；反之， f 是平衡函數。

綜合而言，Deutsch-Jozsa 問題的古典演算法在最壞情況下需要驗證的次數為 $O(2^n)$ ，但是其量子演算法，加上量子位元疊加狀態的準備和測量的時間，在所有的情況下需要的操作步驟為 $O(n)$ ，這說明量子演算法相對於古典演算法具有指數級別的加速。

Bibliography

- [1] Russ Juskalian. Practical quantum computers. *TECHNOLOGY REVIEW*, 120(2):77–81, 2017.
- [2] Erwin Schrödinger. Discussion of probability relations between separated systems. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 555–563. Cambridge University Press, 1935.
- [3] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.
- [4] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.
- [5] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.
- [6] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992.