



# 我與量子電腦 的距離

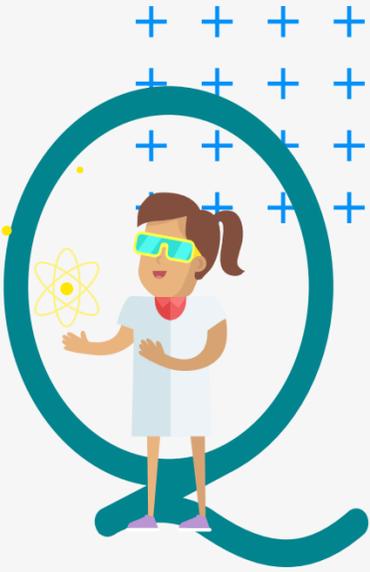
《專業對話版》

曾慶良（阿亮老師）





# 我與量子電腦的距離

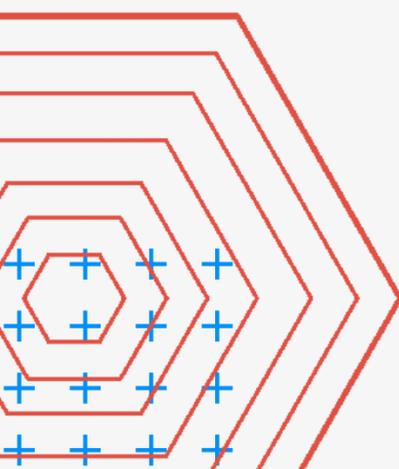


DON'T LOSE  
**FOCUS**

大量子電腦是什麼  
綱

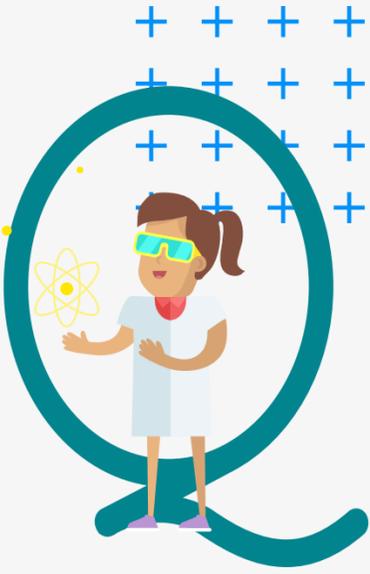
不同資料的處理型態  
量子力學的物理特性  
電路計算模型  
量子位元的測量  
實例說明

曾慶良 (阿亮老師)





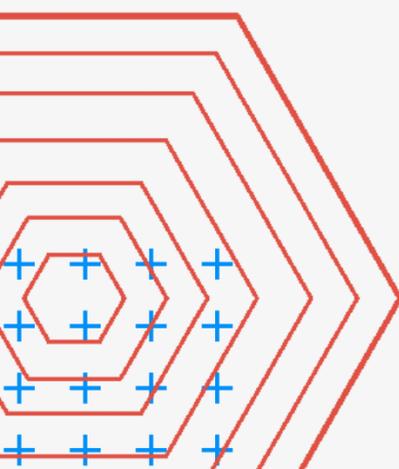
# 我與量子電腦的距離



DON'T LOSE  
**FOCUS**

大 中學生學 AI 與  
網 量子電腦

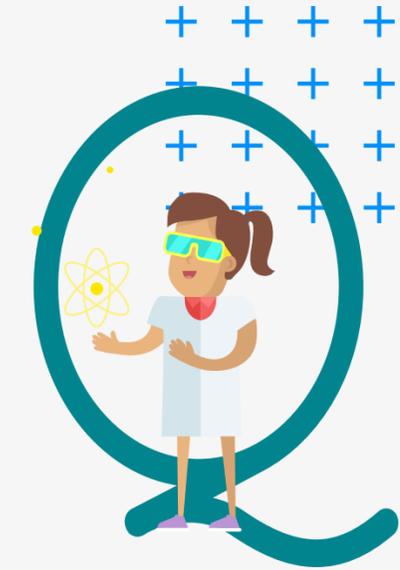
教學的利基  
學生有時間嗎  
學生有能力嗎  
學生有需求嗎  
學生有發展嗎



曾慶良 (阿亮老師)



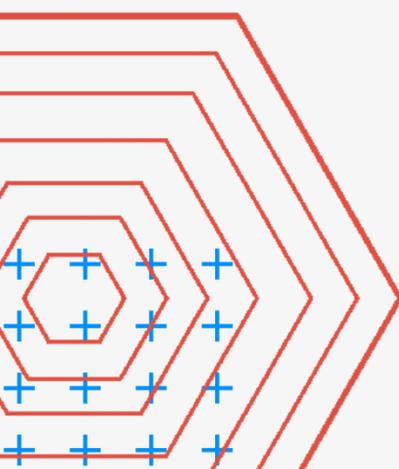
# 我與量子電腦的距離



DON'T LOSE  
**FOCUS**

大 **中學生學 AI 與**  
網 **量子電腦**

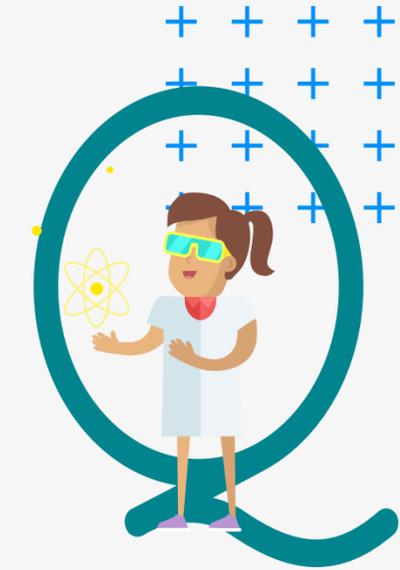
教學的利基  
學生有時間嗎  
學生有能力嗎  
學生有需求嗎  
學生有發展嗎



曾慶良 (阿亮老師)



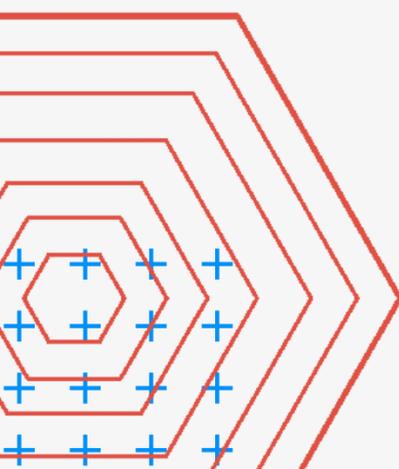
# 我與量子電腦的距離



DON'T LOSE  
**FOCUS**

大量子位元的  
綱表示法

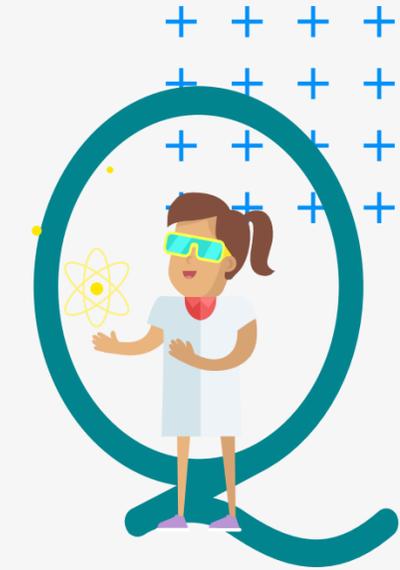
迪拉克符號、布赫洛球、  
波的类型態  
多個量子位元疊加態表示法  
邏輯閘與量子閘  
雙量子位元 CNOT (受控反閘)



曾慶良 (阿亮老師)



# 我與量子電腦的距離



DON'T LOSE  
**FOCUS**

大  
綱

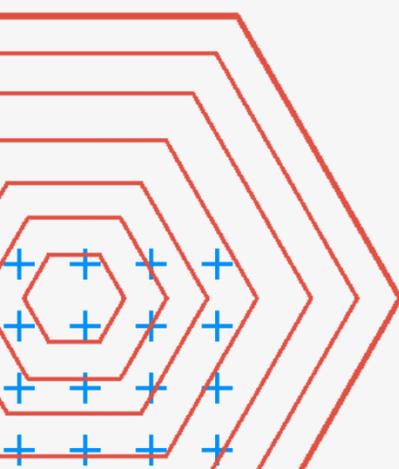
量子位元的表示  
方式及觀測機率

量子演算法

GROVER演算法

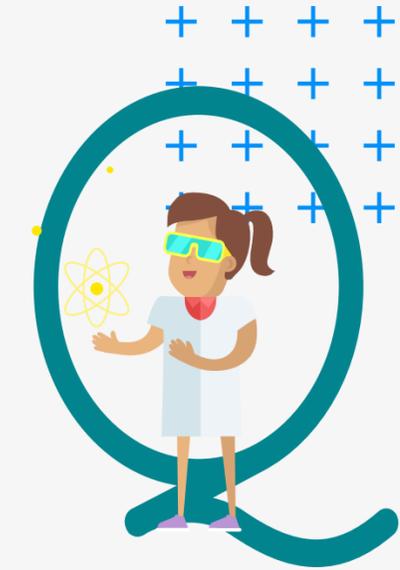
SHOR演算法

曾慶良 (阿亮老師)





# 我與量子電腦的距離



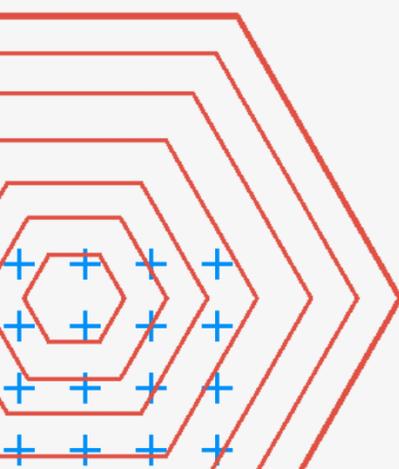
DON'T LOSE  
**FOCUS**

## 大 量子電腦的未來

量子電腦在8種行業應用案例

## 綱 3A中心自製動畫

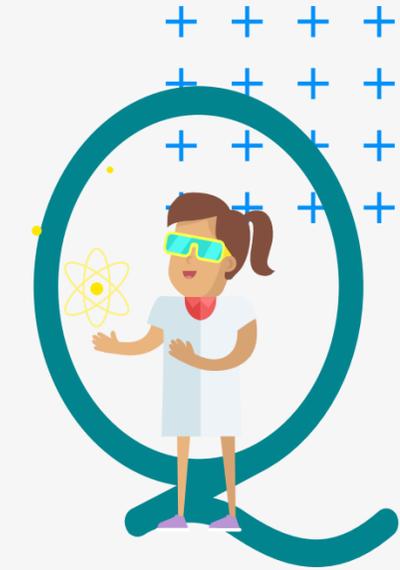
人工智慧與量子電腦動畫



曾慶良 (阿亮老師)



# 我與量子電腦的距離

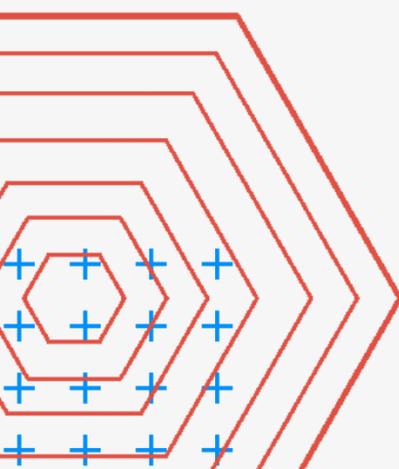


DON'T LOSE  
**FOCUS**

大 量子計算

綱

IBM QUANTUM EXPERIENCE



曾慶良 (阿亮老師)



# 量子電腦是什麼



關於計算這件事...

紙筆  
計算

速度



傳統  
電腦  
計算

速度



量子  
電腦  
計算

不同的資料處理形式





# 量子電腦是什麼



關於計算這件事...

傳統  
電腦  
計算

傳統電腦以位元  
(**BIT**) 的形式  
處理資料，每一  
個位元會在兩種  
狀態中切換，  
這兩種狀態被標  
為 **0** 和 **1**

量子電腦是用量  
子位元  
(**QUBIT**) 來  
做，它可以  
**0**、**1** 的線性組  
合的疊加態。

量子  
電腦  
計算

不同的資料處理形式



# 量子電腦是什麼

關於計算這件事...

傳統  
電腦  
計算



量子  
電腦  
計算

不同的資料處理形式

圖片來源：泛科學網站

# 量子電腦是什麼

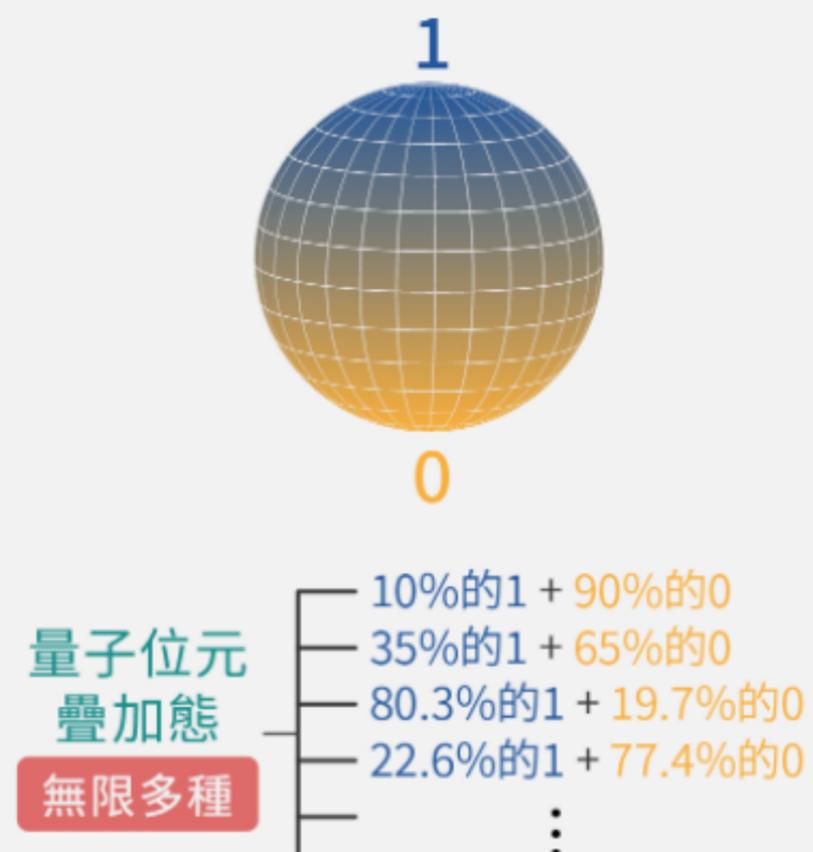
關於計算這件事...



量子位元疊加態

(**SUPERPOSITION**) 把位元的位置以球體標示，南、北極位置分別代表**0**和**1**，傳統電腦的位元只能在兩極之間切換，但若是量子位元疊加時，它能在二維球面上任何位置，不限於南北極。

以量子位元 (qubit) 的形式處理資料



# 量子電腦計算

不同的資料處理形式

圖片來源：泛科學網站





# 量子電腦是什麼

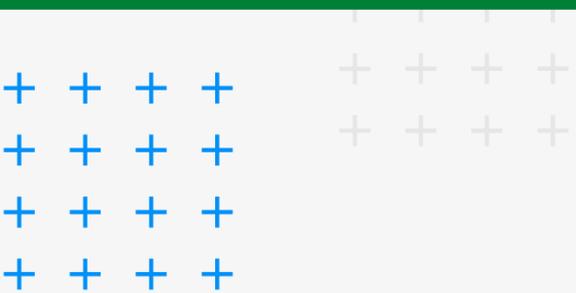
關於計算這件事...



量子電腦：  
積極使用量子力學特有的  
物理狀態來實現  
高速計算的電腦

量子  
電腦  
計算

不同的資料處理形式





# 量子電腦是什麼

關於計算這件事...



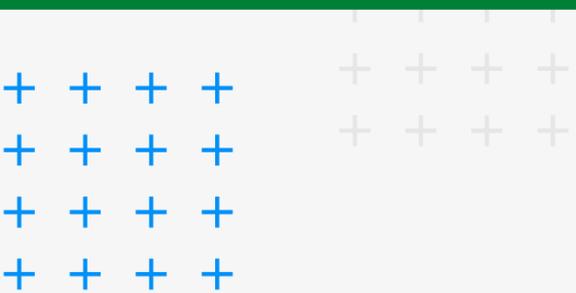
量子力學特有的物理狀態

量子疊加態  
(SUPERPOSITION)

量子糾纏態  
(QUANTUM ENTANGLEMENT)

量子  
電腦  
計算

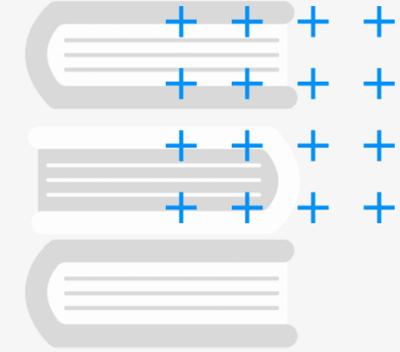
不同的資料處理形式





# 量子電腦是什麼

## 量子疊加態與量子糾纏態



「量子電腦」可在短時間內、處理極大量的資訊，若將**300**位元數字組合作質因數分解，「傳統電腦」需要運算**15**萬年，但「量子電腦」**1**秒鐘內就可算出，「量子電腦」能如此強大是因為「量子電腦」最基本的運算單元「量子位元」(QUBIT)有「量子疊加」和「量子糾纏」同時處於這兩種物理狀態，且兩個量子間需形成聯結，使得兩個量子就算不在於同一個空間，卻可以「即時」互相影響，才能做為量子運算基本單元，也因此才能如此快速處理相當龐大的資訊。同步運算，每個位元可有不同特質，所以才能在短時間內，運算處理龐大資訊。

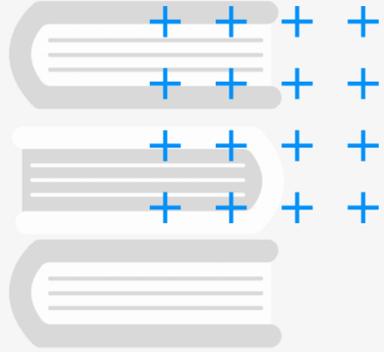
量子可以是電子、離子或光子，只要能夠達到「疊加」和「糾纏」狀態就可以做為「量子位元」。





# 量子電腦是什麼

## 量子疊加態與量子糾纏態

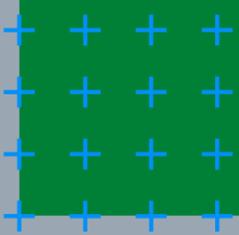
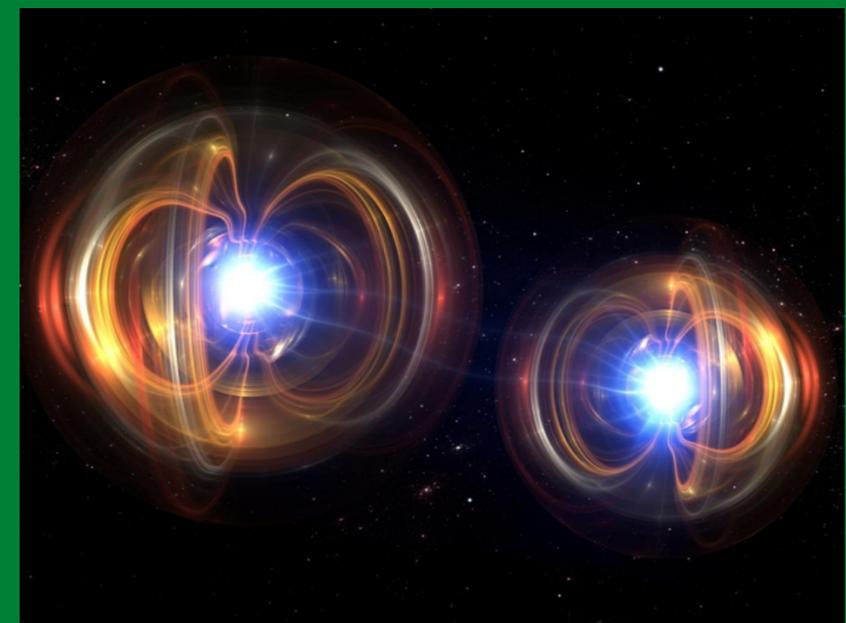


量子疊加：

同一個位置，原本只能儲存**0**和**1**，因為現在可以疊加，疊加上去的東西是可以被分離的，同一個位置可同時儲存好幾個不同位元，運算時不同位元可同步運算，每個位元可有不同特質，所以才能在短時間內，運算處理龐大資訊。

量子糾纏：

當兩個粒子糾纏在一起，兩粒子往外不同方向擴張時，不管距離多遠，去量測其中之一粒子，量測的效益在另外一個粒子也同時感受到。





# 量子電腦是什麼

關於計算這件事...



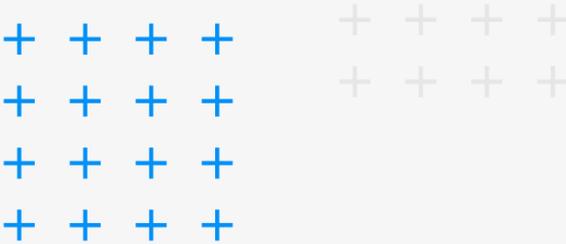
## 量子力學特有的物理狀態

### 測不準原理 (海森堡於1927年提出)

測不準原理並不是說因為工具和手段的限制而造成了結果的觀測有誤。這個理論是從量子物理的角度出發，意思是說，微觀界的粒子行為與宏觀物質很不一樣：你不可能同時知道一個粒子的位置和它的速度。

# 量子 電腦 計算

## 不同的資料處理形式





# 量子電腦是什麼

關於計算這件事...



## 量子力學特有的物理狀態

### 薛丁格的貓 (薛丁格於1935年提出)



# 量子 電腦 計算

著名思想實驗的，描述量子力學的真相：粒子的某些特性無法確定，直到測量外力迫使它們選擇。整個實驗是這樣進行的：在一個盒子裡有一隻貓，以及少量放射性物質。在一小時內，大約有**50%**的概率放射性物質將會衰變並釋放出毒氣殺死這隻貓，剩下**50%**的概率是放射性物質不會衰變而貓將活下來。

# 量子電腦是什麼

電路計算模型...

傳統  
電腦  
計算

使在古典電腦裡的「電路」與「邏輯閘」來進行計算的模型

使用「量子電路」與「量子」來進行計算的模型

量子  
電腦  
計算

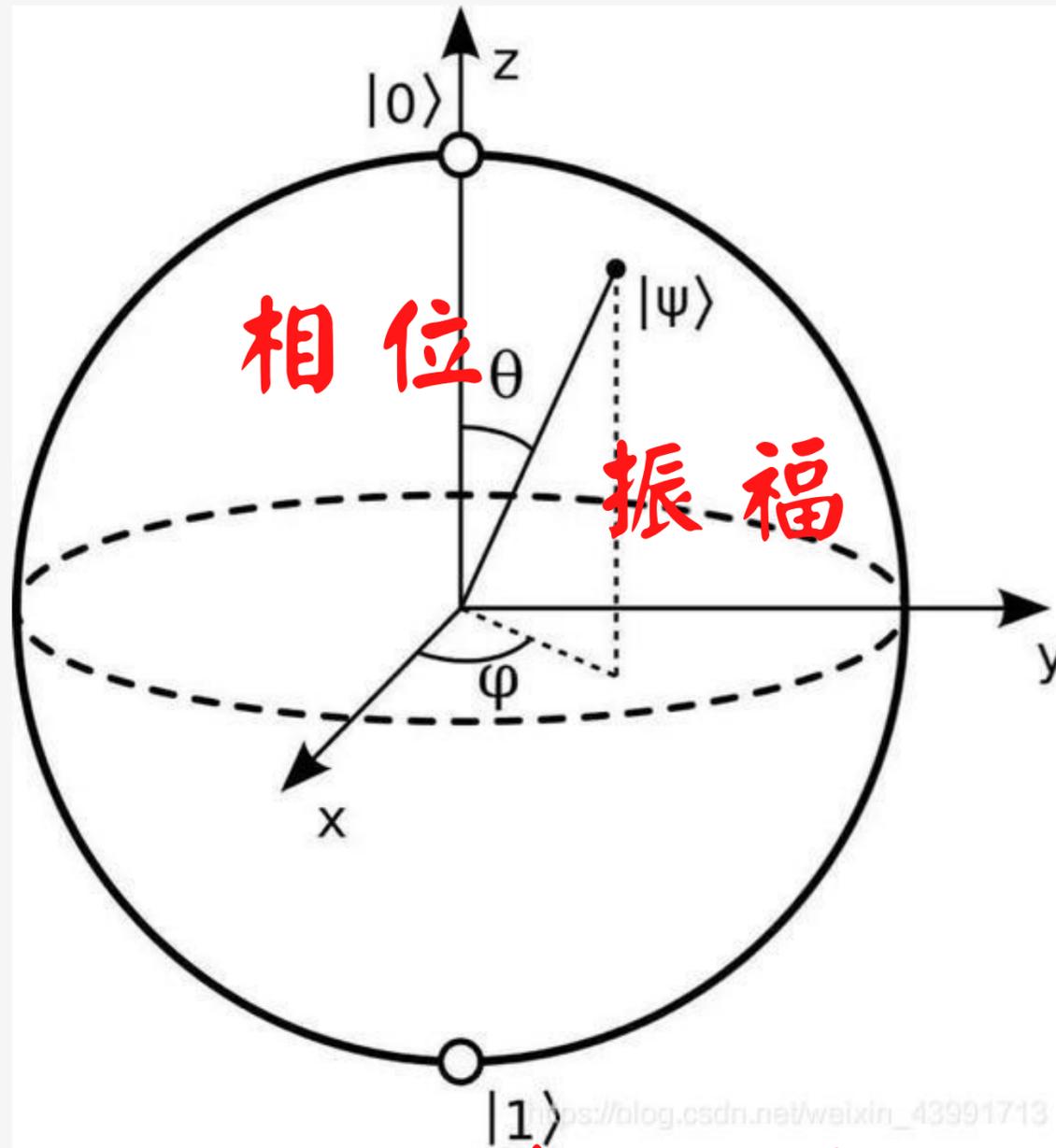
不同的資料處理形式



# 量子電腦是什麼



## 量子位元的測量



### 布洛赫球

圖片來源：維基百科

1. 測量前，於“0”與“1”的疊加態，指向布洛赫球表面的箭頭來表示（以振幅與相位表示）。
2. 當「測量」該量子位元，會以機率來決定為“0”狀態或“1”狀態。
3. 測量量子位元時出現“0”或“1”的機率，是由將測量前指向的箭頭投影至過0與1的軸來決定，如果投影而得的箭頭較接近“0”則出現“0”的機率較高，較接近“1”則出現“1”的機率較高。
4. 藉由測量可讀出0或1的古典位元資訊，而測量後的量子位元狀態，變化為與測量結果相同的“0”狀態或“1”狀態。





# 量子電腦是什麼



## 量子位元的測量

向量  
特性



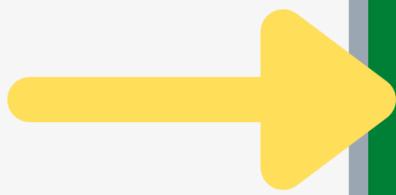
1. 測量前,於“0”與“1”的疊加態,指向布洛赫球表面的箭頭來表示(以振幅與相位表示)。

機率  
特性



2. 當「測量」該量子位元,會以機率來決定為“0”狀態或“1”狀態。

矩陣  
特性



3. 測量量子位元時出現“0”或“1”的機率,是由將測量前指向的箭頭投影至過0與1的軸來決定,如果投影而得的箭頭較接近“0”則出現“0”的機率較高,較接近“1”則出現“1”的機率較高。

4. 藉由測量可讀出0或1的古典位元資訊,而測量後的量子位元狀態,變化為與測量結果相同的“0”狀態或“1”狀態

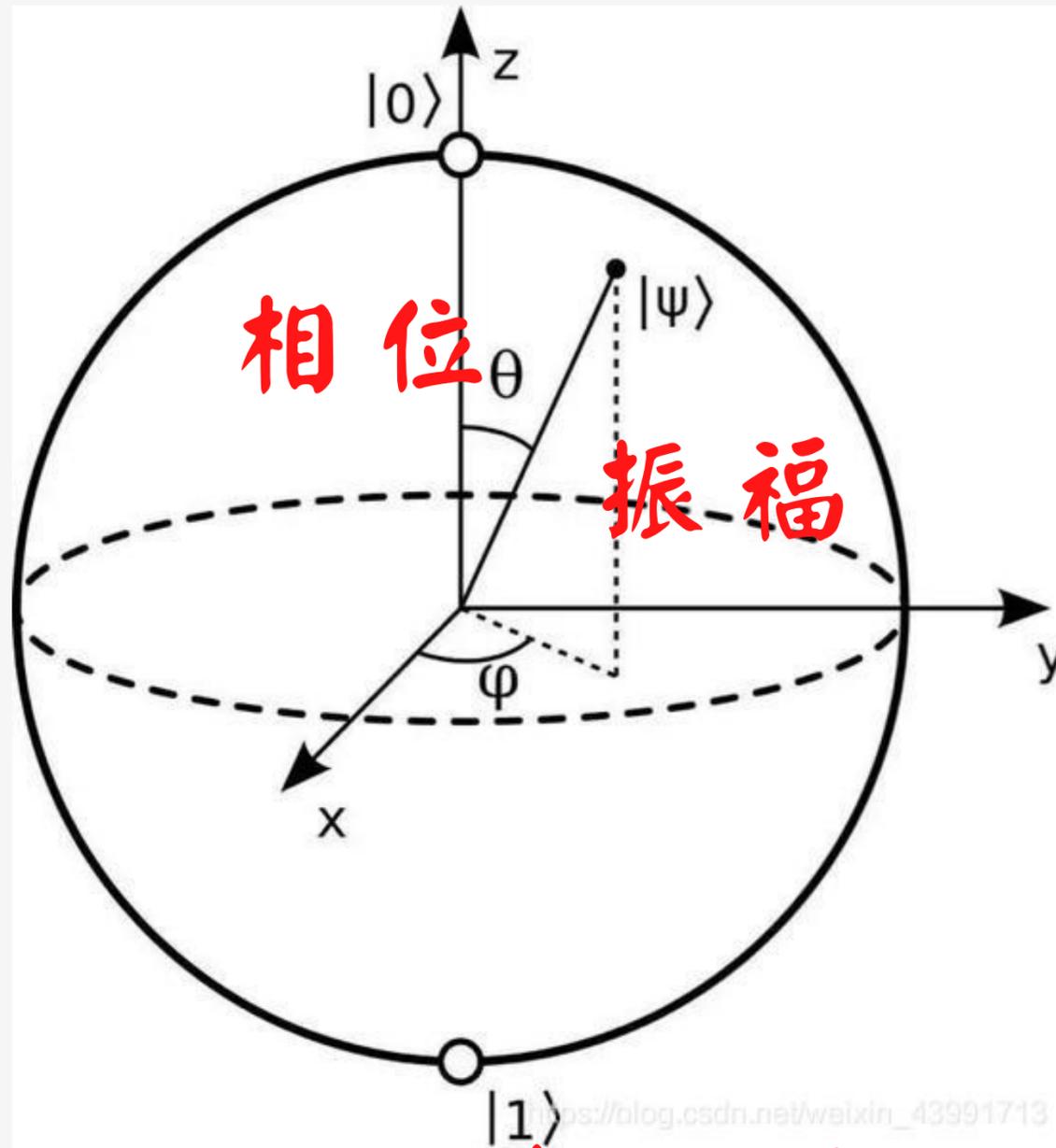




# 量子電腦是什麼



## 量子位元的測量



$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

其中  $|\alpha|^2 + |\beta|^2 = 1, \forall \alpha, \beta \in C$

布洛赫球

圖片來源：維基百科





# 量子電腦是什麼



## 以計程車分配問題為例

### 問題

3位人士坐2輛計程車，現在必須規劃如何分配座位。爲了確保旅程氣氛和樂，希望透過座位分配達成兩個目標：

最大化同輛車的朋友數量，同時最小化同輛車的敵人數量。

假設已知：

- 阿亮 和 水水 是朋友
- 阿亮 和 彤彤 是敵人
- 水水 和 彤彤 是敵人

爲了方便起見，設阿亮爲 **A**、水水爲 **B**、彤彤爲 **C**，  
計程車編號爲 **0** 與 **1**





# 量子電腦是什麼



## 以計程車分配問題為例

若不考慮計程車搭乘人數的上限和下限，每個人都有 2 個選擇，因此共有  $2^3 = 8$  種方法將這組人分成兩輛車。

傳統  
電腦  
解決  
方式

A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1





# 量子電腦是什麼



## 以計程車分配問題為例

定義每項組合的「分數」比較每個方案：

- (朋友共享同一輛車) - (敵人共用同一輛車) = 得分

A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1
分數	-1	1	-1	-1	-1	-1	1	-1

得分為1的

001和110 是兩個最佳解答

傳統  
電腦  
解決  
方式





# 量子電腦是什麼



## 以計程車分配問題為例

定義每項組合的「分數」比較每個方案：

- (朋友共享同一輛車) - (敵人共用同一輛車) = 得分

A	0	0	0	0	1	1	1	1
B	0	0	1	1	0	0	1	1
C	0	1	0	1	0	1	0	1
分數	-1	1	-1	-1	-1	-1	1	-1

但若改為100人，以此種方式要在極短時間計算 $2^{100}$ 對應的值，傳統電腦無法做到

傳統  
電腦  
解決  
方式





# 量子電腦是什麼



## 以計程車分配問題為例

### 量子 電腦 解決 方式

運用量子位元，每個量子位元可以有**0**與**1**兩種選擇（平行世界）  
因此只需要運用**3**個量子位元，即可以有  
**000**、**001**、**010**、**011**、**100**、**101**、**110** 和 **111**  
八個平行世界  
只需要一次操作，即可以得到答案  
只要量子位元數夠多，就可以用最短的時間獲取答案

但需要注意兩件事：

1. 要將量子位元對應到解決方案
2. 需要給予解決方案的函數





# 量子電腦是什麼



## 量子電腦的工作流程

**定義問題** 根據專業領域訂出演算規則並將問題系統化。

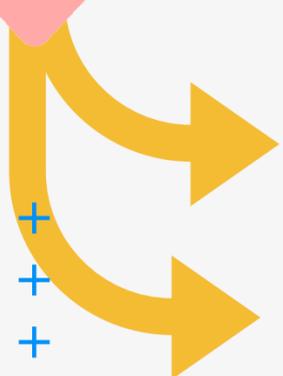
**量子演算法** 決定運用何種量子演算法與模型。

**量子電路** 產生基於量子邏輯閘的量子電路以解決複雜的計算問題。

**量子編譯器** 將高階程式語言進行編譯，進而產生量子組合語言。

**模擬器**

**量子電路**





# 中學生學 AI 與量子電腦

利基何在

疑問 1：學生有能力嗎？

疑問 2：學生有時間嗎？

疑問 3：學生有需求嗎？

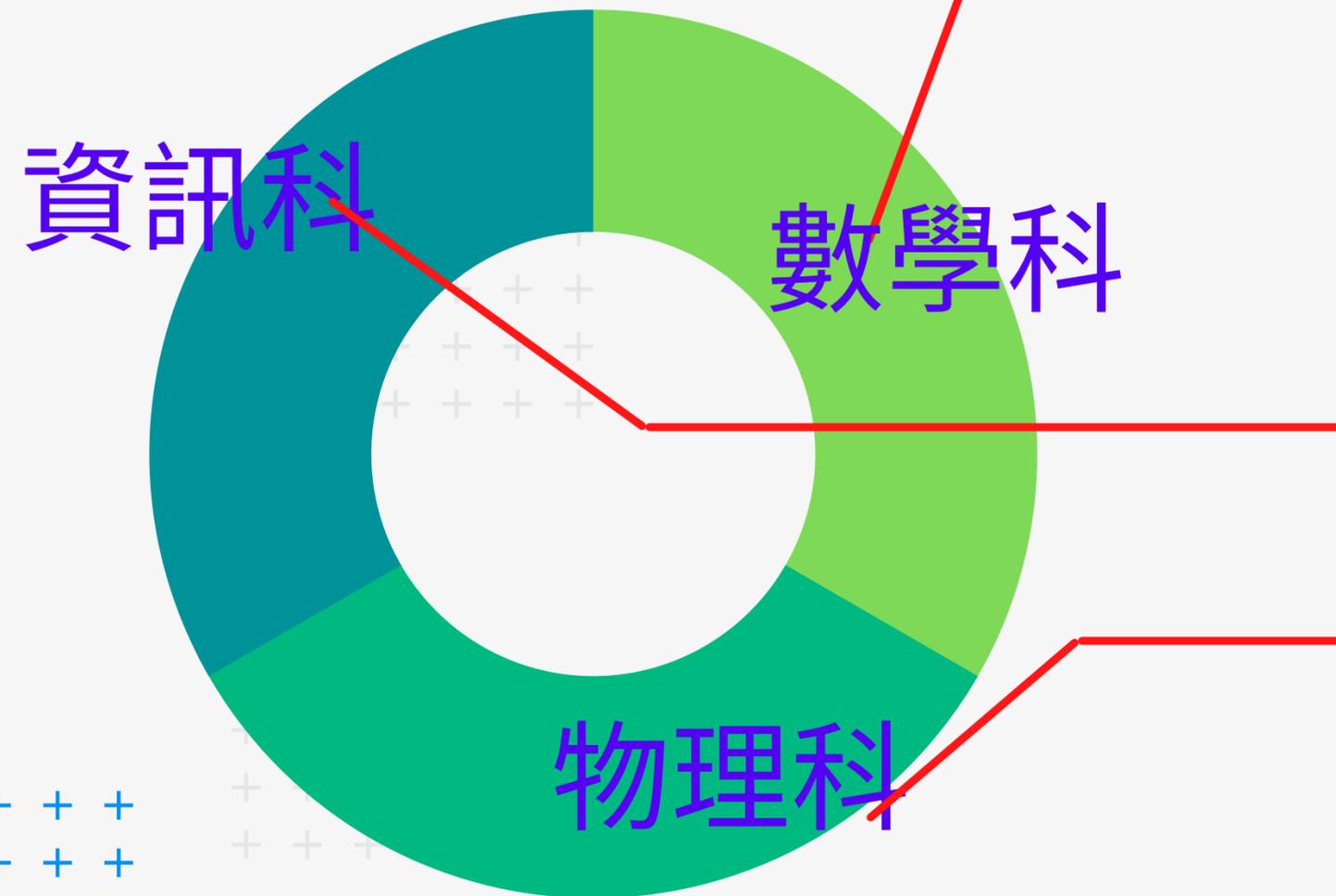
疑問 4：學生有發展嗎？





# 中學生學 AI 與量子電腦

## 疑問1：學生有能力嗎？



函數、向量、矩陣  
三角函數、複數

布林代數、卡諾圖  
真值表

(AND NAND OR NOR XOR XNOR)

電磁波、  
量子現象概念





# 中學生學AI與量子電腦



## 疑問2：學生有時間嗎？





# 中學生學AI與量子電腦



疑問3：學生有需求嗎？

## 學習歷程檔案

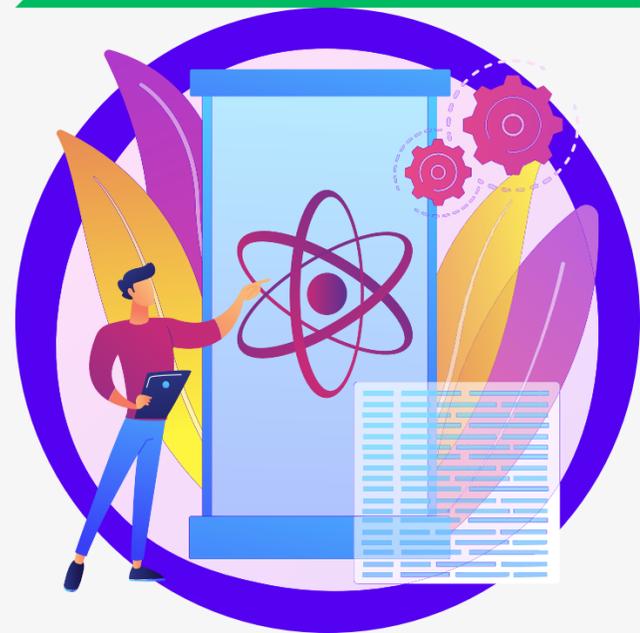
部編課程深入專題報告

多元選修課程成果

自主學習課程成果

社團課程成果與競賽

校外競賽歷程





# 中學生學 AI 與量子電腦



疑問4：學生有發展嗎？

## 升學與職涯發展

提交學習歷程檔案

參與大學升學面試

了解未來職業內容

與個人職涯規劃

校外競賽歷程





# 中學生學 AI 與量子電腦



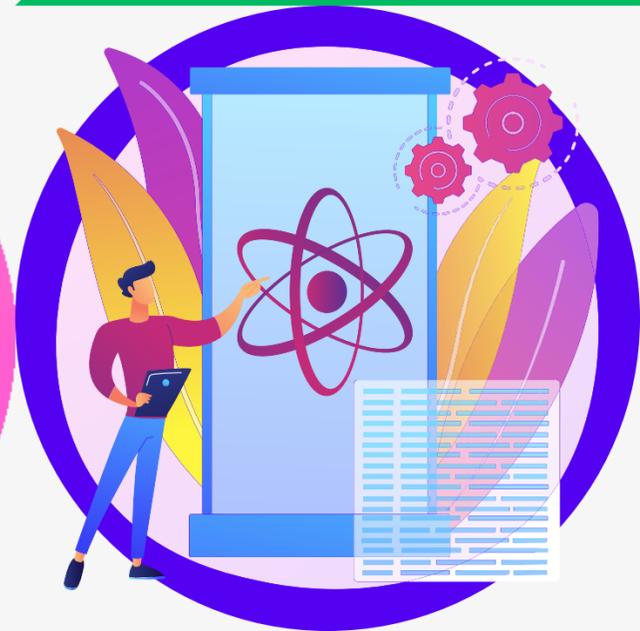
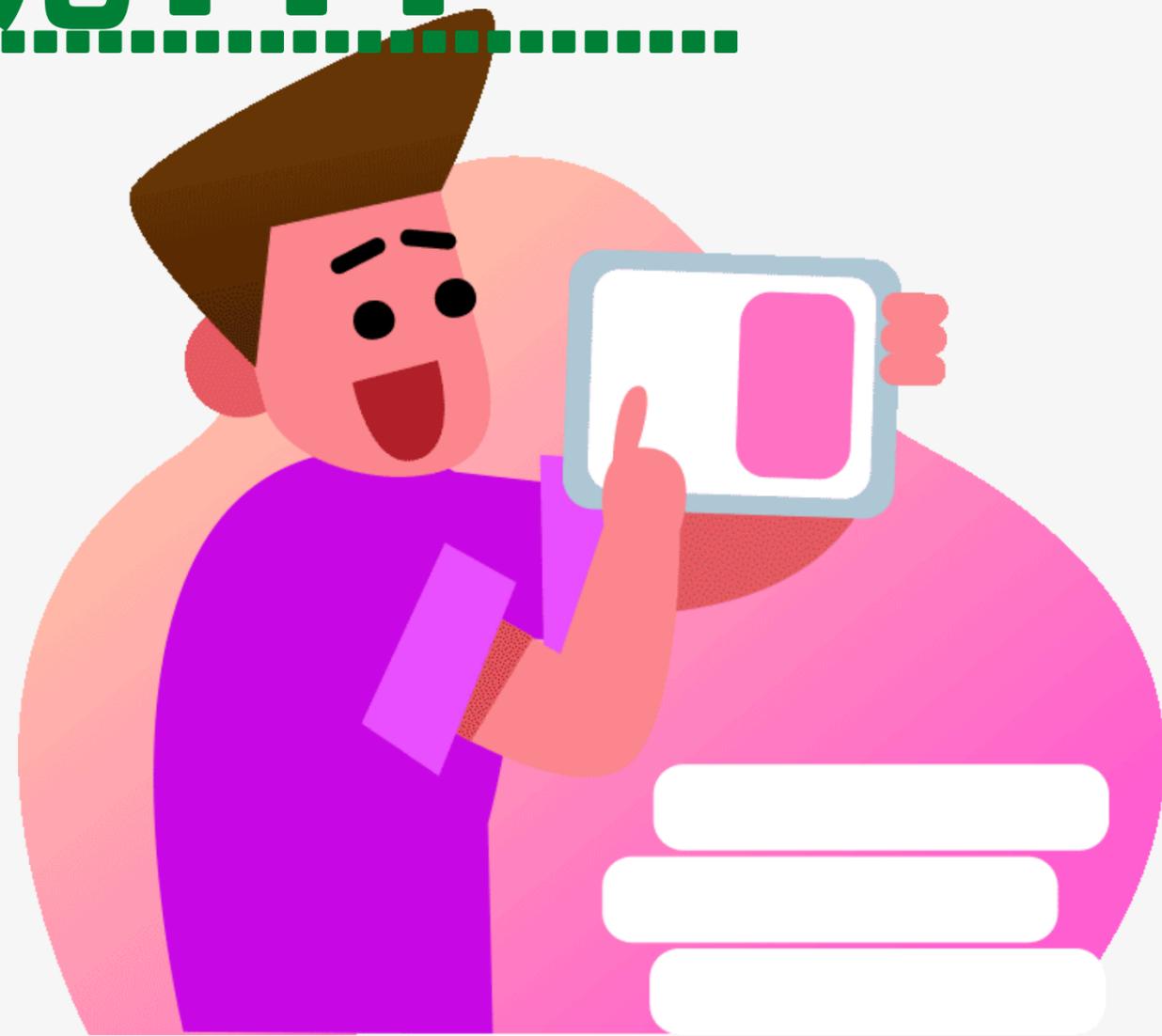
疑問：那...老師呢???

有能力教?

開什麼課?

哪來教材?

課程成效?





# 量子電腦教學建議



國小  
老師

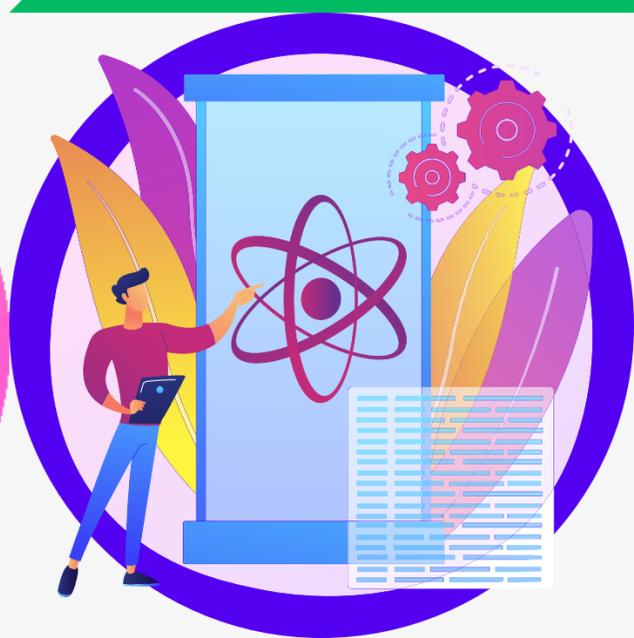
：量子基本概念認知、名詞概念認知

國中  
老師

：邏輯閘基礎運算、量子遊戲APP

高中  
老師

：邏輯閘運算、數學物理資訊知能、Python程式

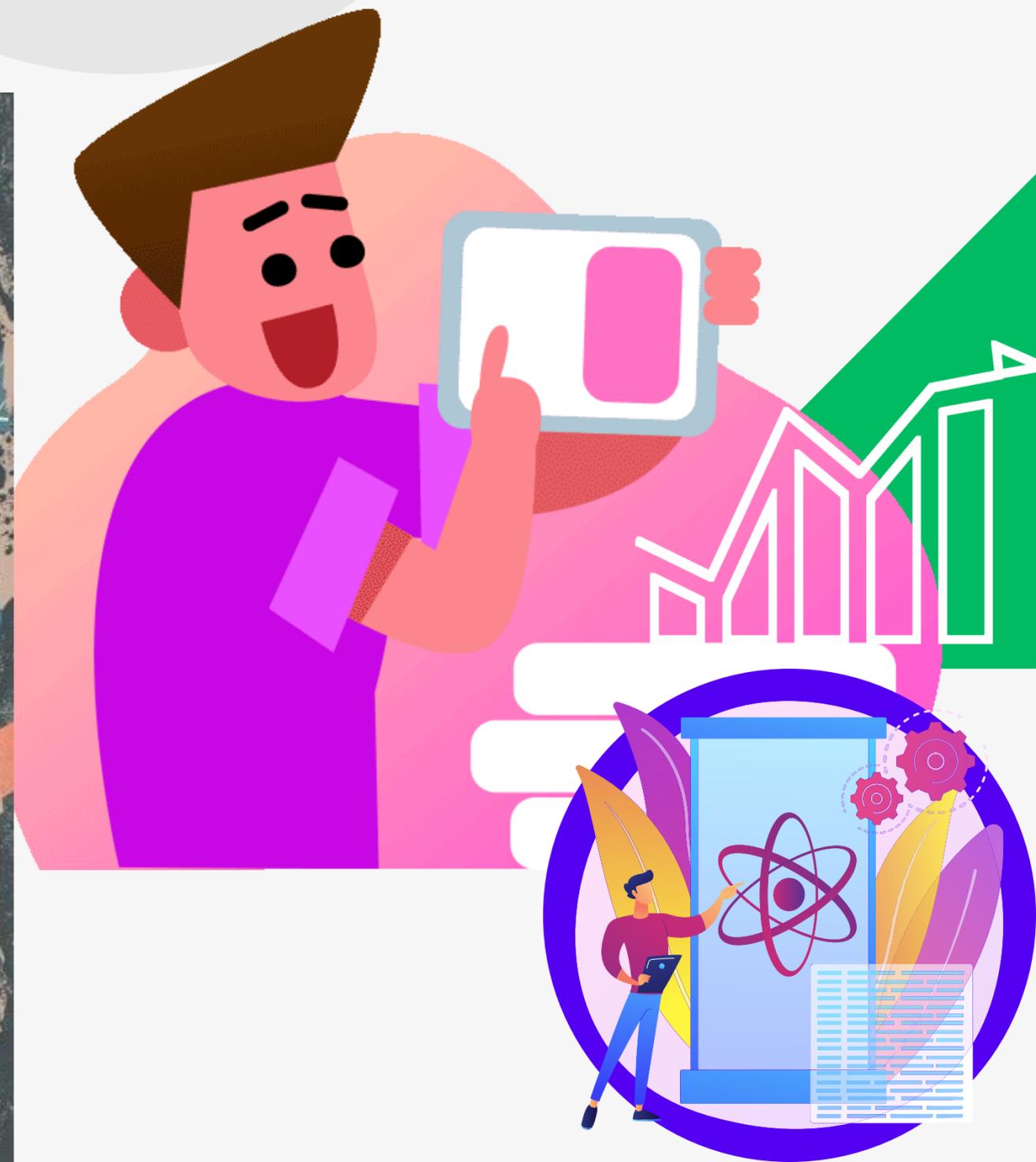
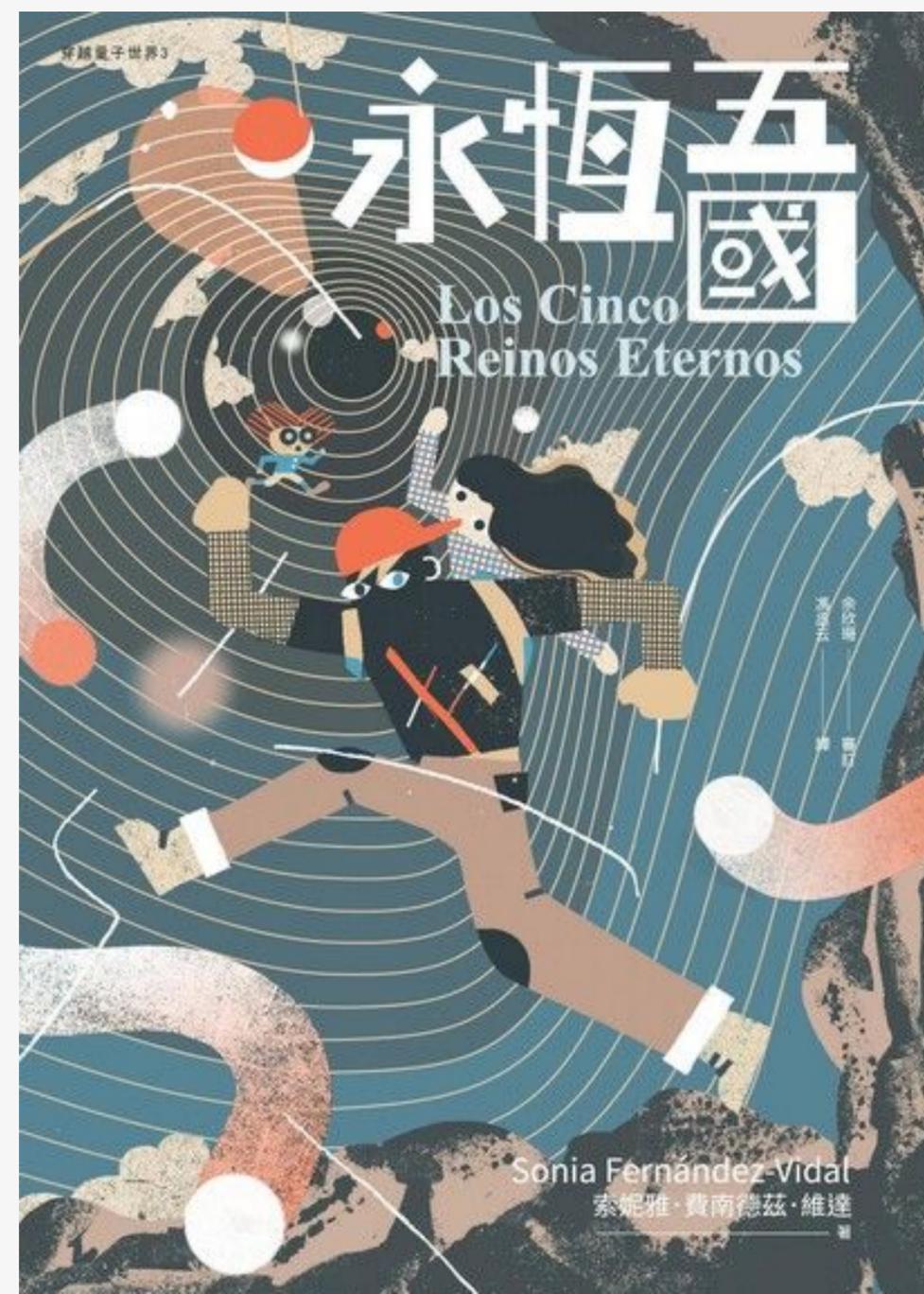




# 量子電腦推薦用書



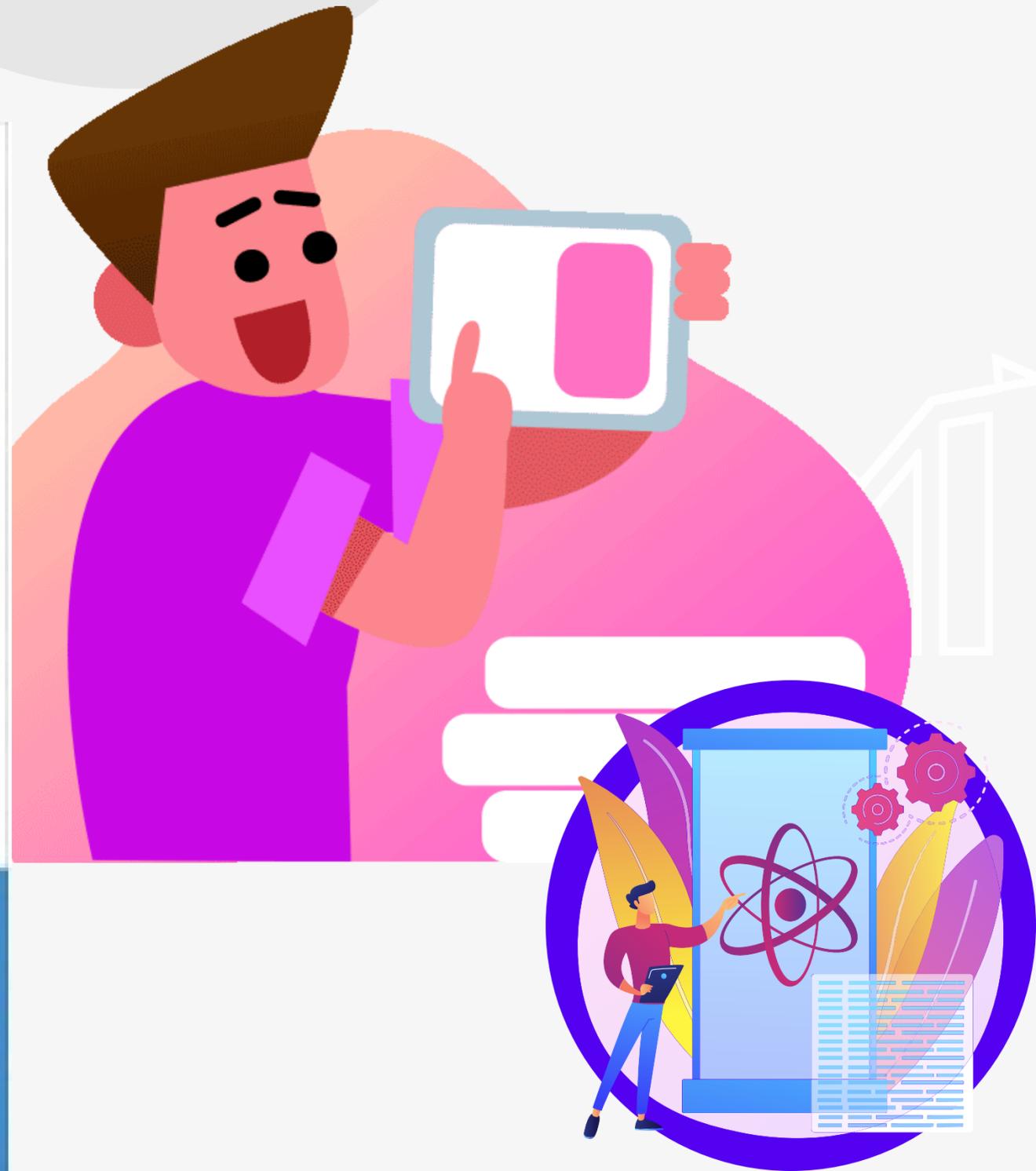
## 國小用書





# 量子電腦推薦用書

## 國中用書

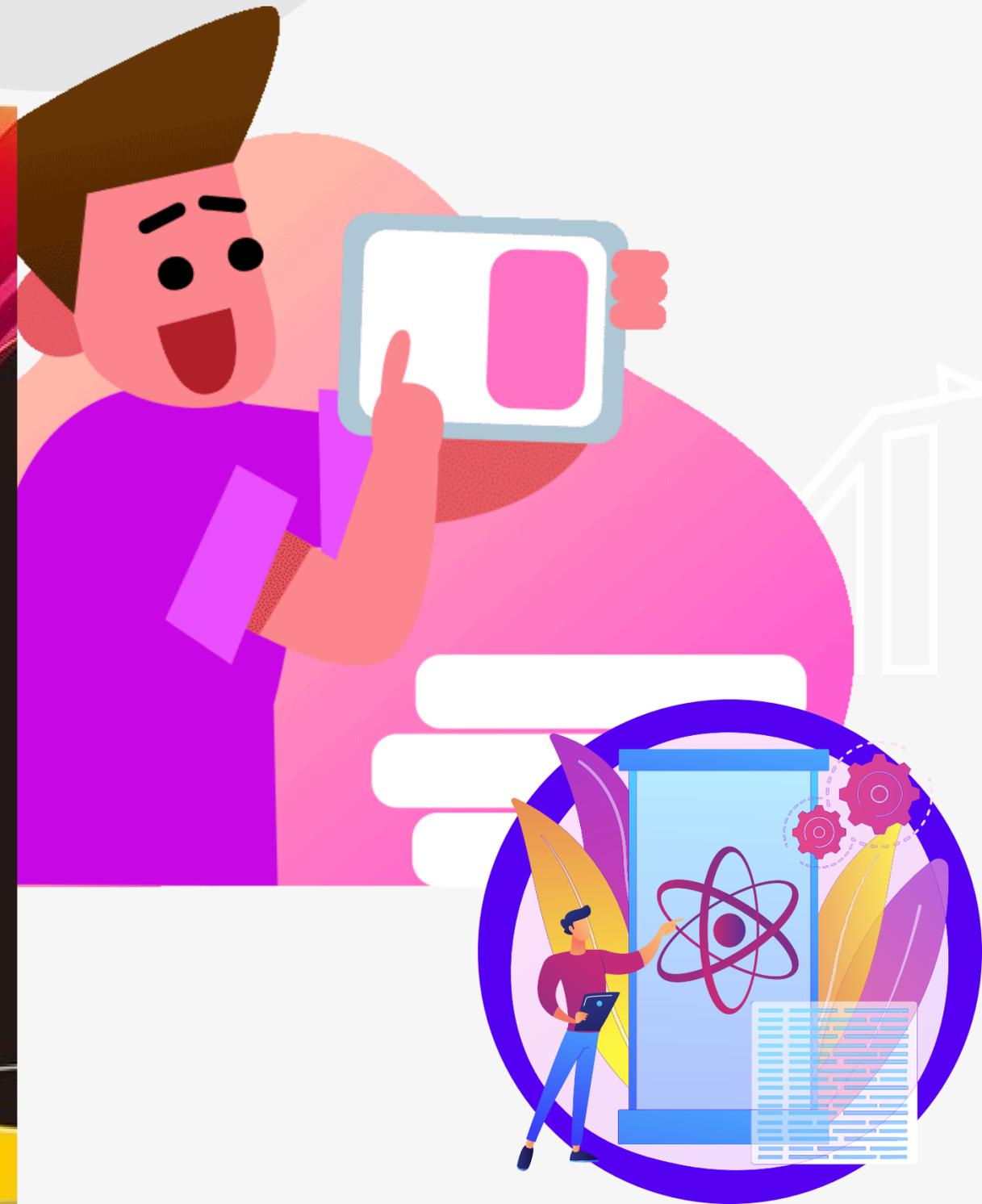
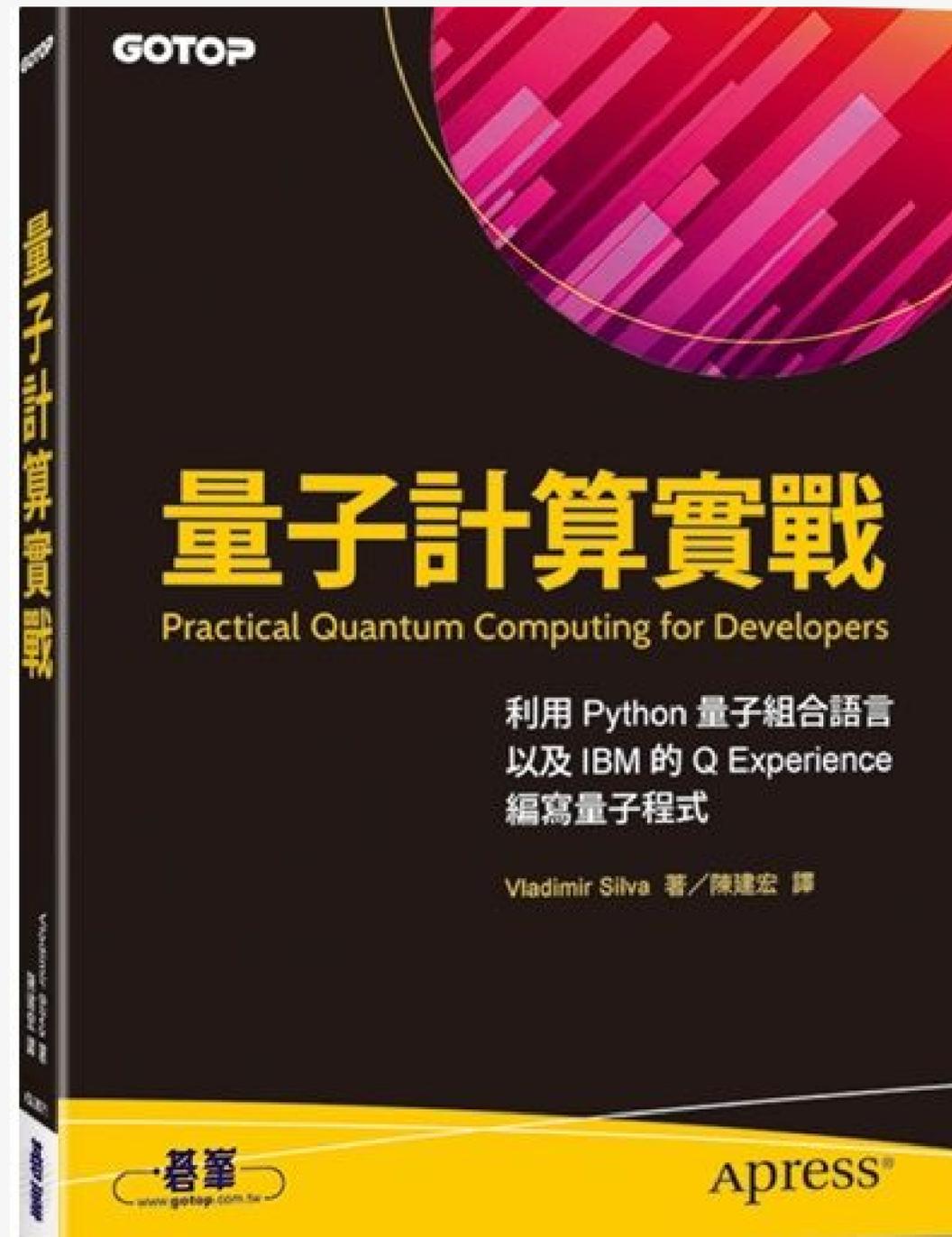




# 量子電腦推薦用書



## 高中用書





# 量子位元的表示法

## 迪拉克符號

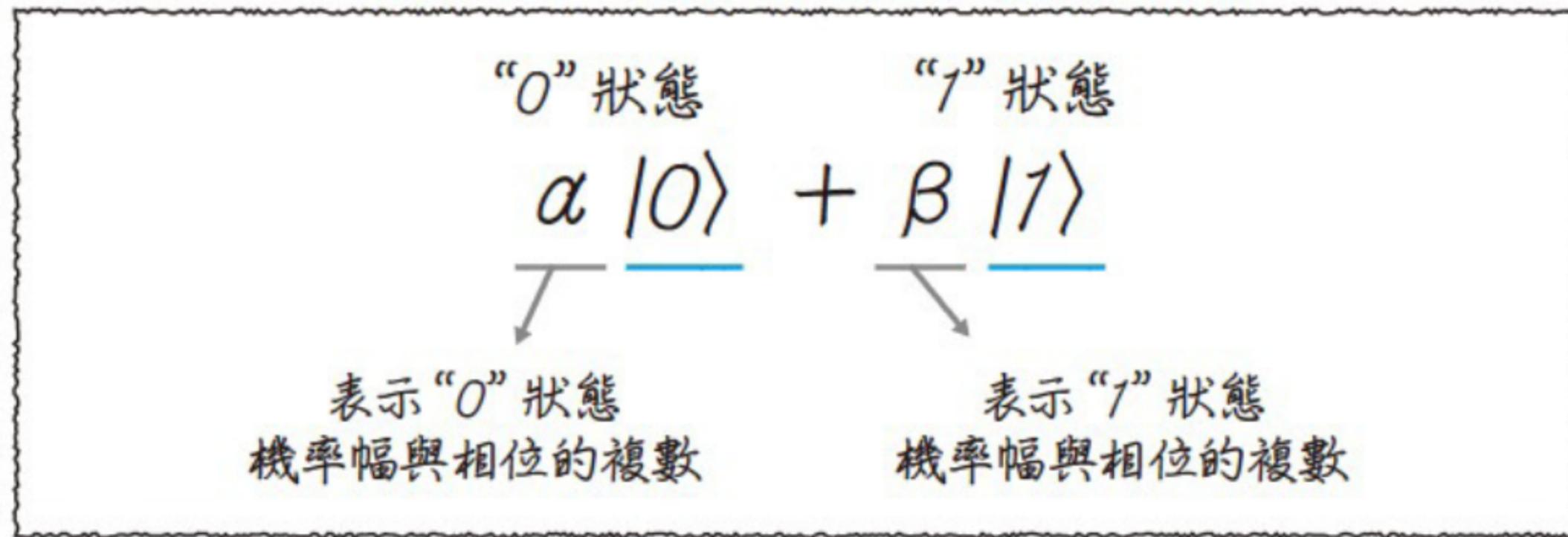


$|0\rangle$  表示 0 的狀態  
 $|1\rangle$  表示 1 的狀態

$|\alpha|^2$  表示測量出現  $|0\rangle$  的機率

$|\beta|^2$  表示測量後出現  $|1\rangle$  的機率

使用狄拉克符號的疊加態表示法



$$|\alpha|^2 + |\beta|^2 = 1$$

圖片來源：圖解量子電腦入門

$\alpha$  與  $\beta$  是分別表示  $|0\rangle$  與  $|1\rangle$  佔了多少比例來疊加的「複數」

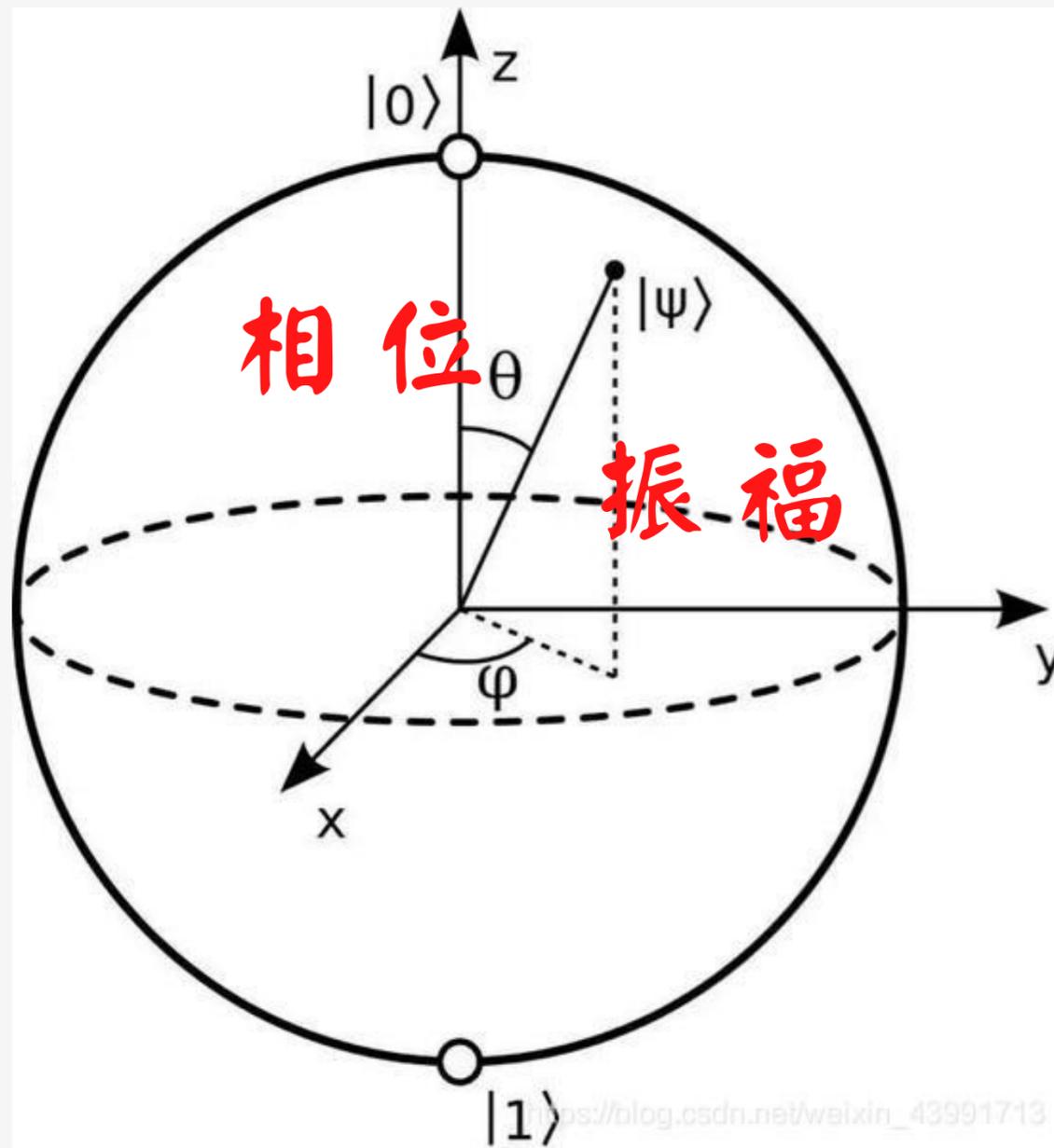




# 量子位元的表示法

## 布赫洛球

## 波的类型態



複數 "a" 表現波

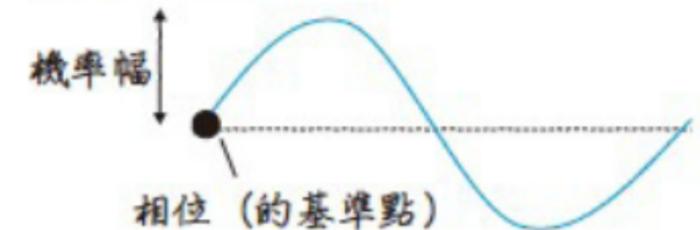
$$a (\text{複數}) = A (\cos \phi + i \sin \phi)$$

↑  
機率幅  
(實數)

↑  
相位  
(實數)



$$a (\text{複數}) =$$



圖片來源：圖解量子電腦入門

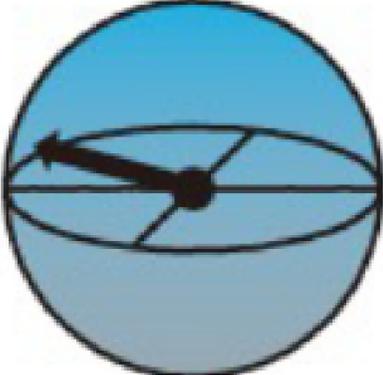
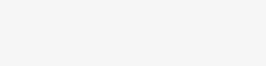


圖片來源：維基百科



# 量子位元的表示法

## 多個量子位元疊加態表示法

	狄拉克符號	布洛赫球	以波來表現
一個量子位元的疊加態	$\alpha  0\rangle + \beta  1\rangle$ ( $ \alpha ^2 +  \beta ^2 = 1$ )		$ 0\rangle$  $ 1\rangle$ 
多個狀態的疊加態	$\alpha  000\rangle + \beta  001\rangle + \gamma  010\rangle + \dots + \eta  111\rangle$	—	$ 000\rangle$  $ 001\rangle$  $ 010\rangle$  $ 011\rangle$  $ 100\rangle$  $ 101\rangle$  $ 110\rangle$  $ 111\rangle$ 

圖片來源：圖解量子電腦入門





# 量子位元的表示法



電路計算模型...

傳統  
電腦  
計算

使在古典電腦裡的「電路」與「邏輯閘」來進行計算的模型

使用「量子電路」與「量子」來進行計算的模型

量子  
電腦  
計算

不同的資料處理形式





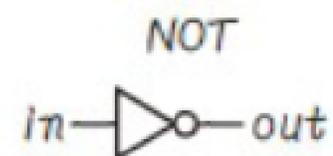
# 量子位元的表示法



## 邏輯閘

# 傳統 電腦 計算

NOT閘 (反閘)



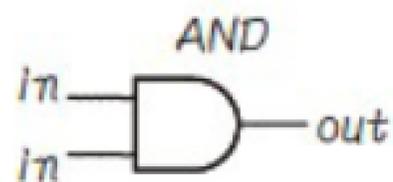
in		out
0		1
1		0

XOR閘 (互斥或閘)



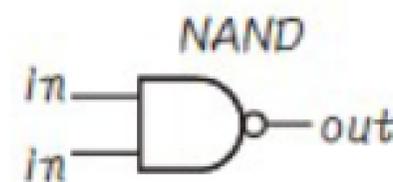
in		out
0	0	0
0	1	1
1	0	1
1	1	0

AND閘 (及閘)



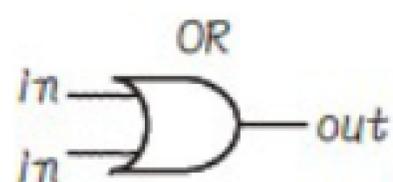
in		out
0	0	0
0	1	0
1	0	0
1	1	1

NAND閘 (反及閘)



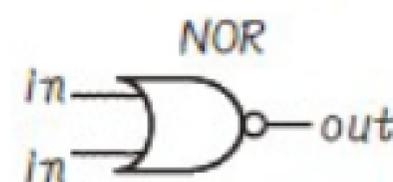
in		out
0	0	1
0	1	1
1	0	1
1	1	0

OR閘 (或閘)



in		out
0	0	0
0	1	1
1	0	1
1	1	1

NOR閘 (反或閘)



in		out
0	0	1
0	1	0
1	0	0
1	1	0





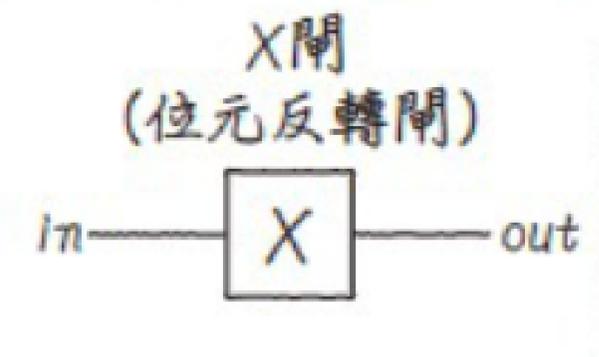
# 量子位元的表示法



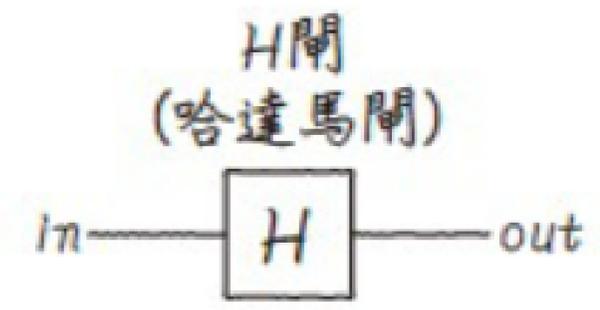
# 量子電腦計算



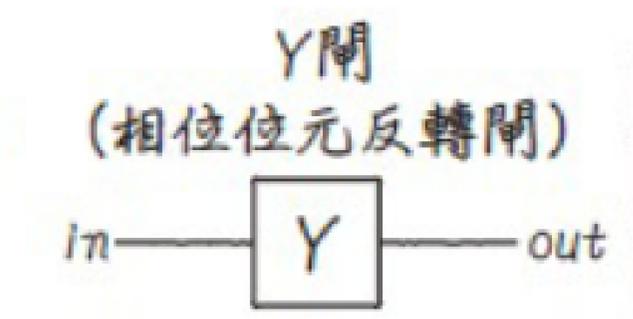
## 量子閘 (單一量子位元)



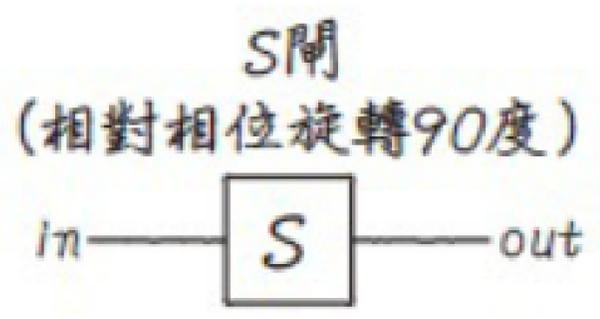
in	out
$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$



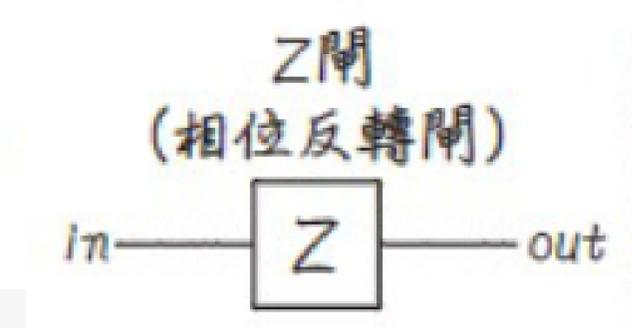
in	out
$ 0\rangle$	$\frac{ 0\rangle +  1\rangle}{\sqrt{2}}$
$ 1\rangle$	$\frac{ 0\rangle -  1\rangle}{\sqrt{2}}$



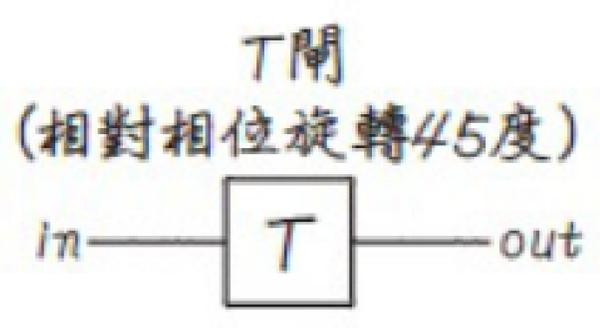
in	out
$ 0\rangle$	$i 1\rangle$
$ 1\rangle$	$-i 0\rangle$



in	out
$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$i 1\rangle$



in	out
$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$- 1\rangle$



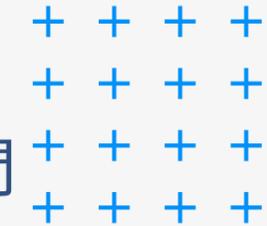
in	out
$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$e^{i\pi/4} 1\rangle$



# 量子位元的表示法



# 量子電腦計算

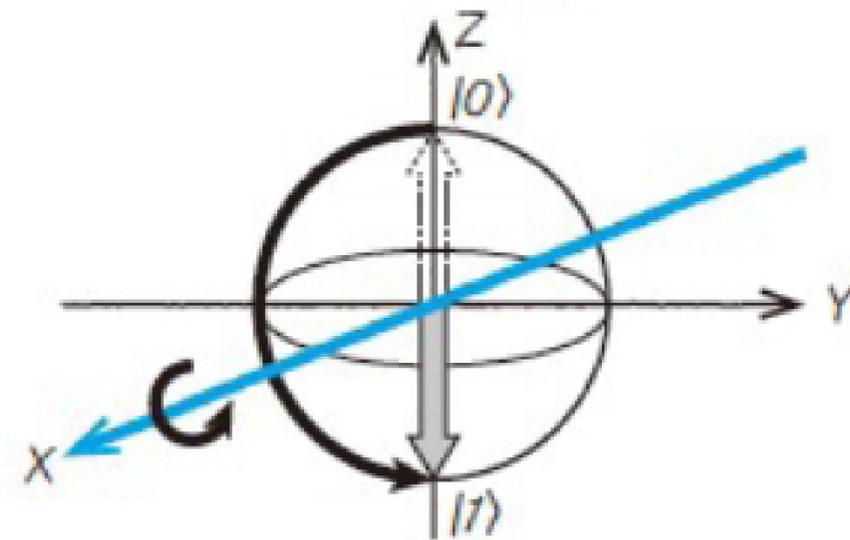


## 量子閘 (單一量子位元)

X閘  
(位元反轉閘)



in	out
$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$



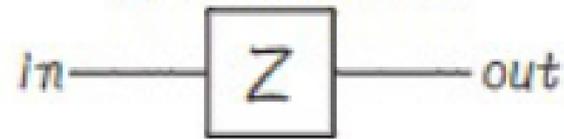
以X軸為中心進行180度旋轉操作

Y閘  
(相位位元反轉閘)

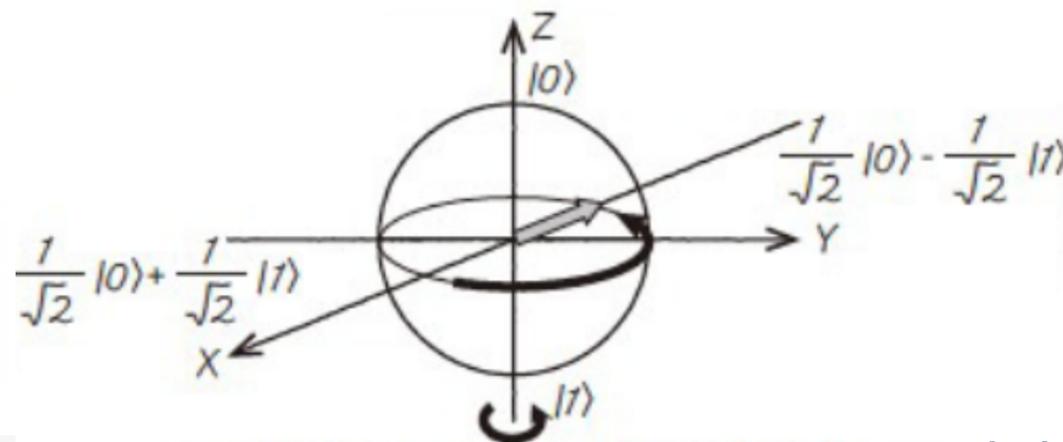


in	out
$ 0\rangle$	$i 1\rangle$
$ 1\rangle$	$-i 0\rangle$

Z閘  
(相位反轉閘)



in	out
$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$- 1\rangle$



以Z軸為中心進行180度旋轉操作 圖片來源：圖解量子電腦入門



# 量子位元的表示法

## 雙量子位元 CNOT (受控反閘)



兩個量子位元的量子閘 (雙量子位元閘)，若操作三個以上量子位元的量子閘，則藉由組合單一量子位元與雙量子位元來實現。

**CNOT** 為控制 (**CONTROLLED**) **NOT** 的閘 (受控反閘)，有兩個輸入與兩個輸出，在兩個輸入當中，一邊稱為控制 (**CONTROL**) 位元，另一邊稱為目標 (**TRGET**) 位元。

**CNOT** 動作：

當控制位元裡輸入  $|0\rangle$ ，不對目標元做任何操作；

當控制位元裡輸入  $|1\rangle$ ，對目標位元實施 **X** 閘 (**NOT**, 位元轉) 操作。

特徵是隨著控制位元的狀態來變化對目標位元的動作，控制位元扮演著是否反轉目標位元的開關角色。





# 量子位元的表示法

## 雙量子位元 CNOT (受控反閘)



### 受控反閘

輸入		輸出	
控制位元	目標位元	控制位元	目標位元
$ 0\rangle$	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$
$ 0\rangle$	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$
$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$

CNOT動作：

當控制位元裡輸入 $|0\rangle$ ，  
不對目標元做任何操作；

當控制位元裡輸入 $|1\rangle$ ，  
對目標位元實施X閘(NOT,位元轉)  
操作。

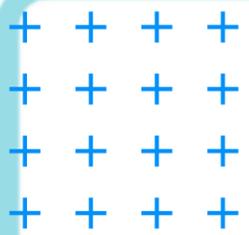
圖片來源：善科教育基金會

是否對目標位元進行X閘操作取決於控制位元的輸入態

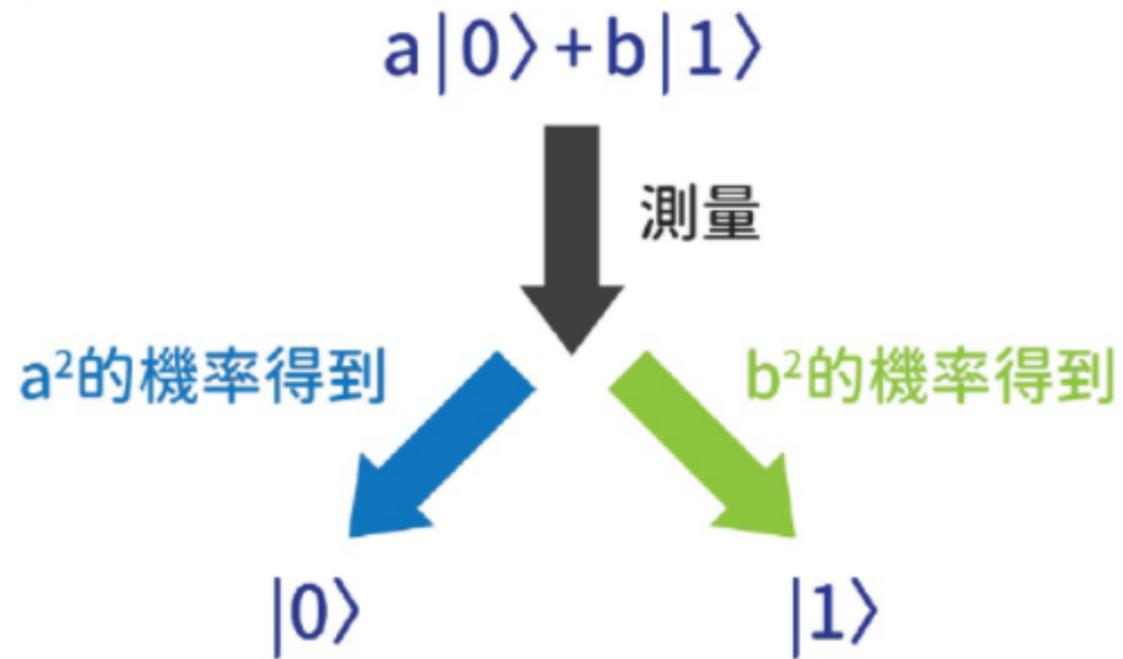




# 量子位元的表示方式及觀測機率

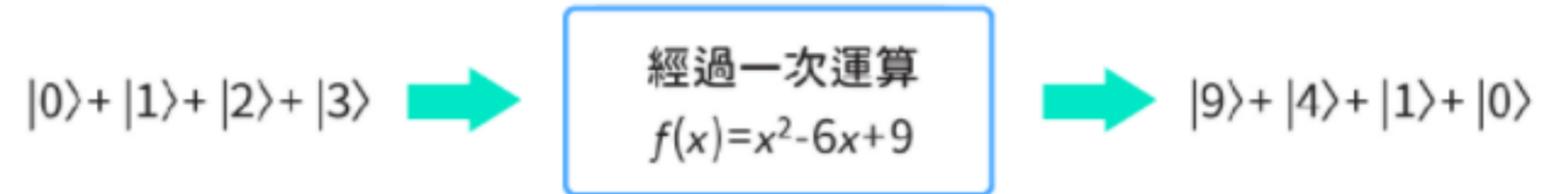


## 量子位元的表示方式及觀測機率



物理學家習慣用  $|0\rangle$  來代表量子位元0，用  $|1\rangle$  來代表量子位元1。而圖中的  $a|0\rangle + b|1\rangle$  則代表  $|0\rangle$  和  $|1\rangle$  的疊加態，其中  $a$  和  $b$  代表兩者的比例。不過，在量子力學中， $a$  和  $b$  並不代表觀測疊加態時得到對應結果的機率，而是  $a$  和  $b$  的平方。也就是說，對於  $a|0\rangle + b|1\rangle$  這個疊加態，觀測到  $|0\rangle$  的機率為  $a^2$ ，而觀測到  $|1\rangle$  的機率則為  $b^2$ 。

## 量子位元的表示方式及觀測機率



圖中的  $|0\rangle + |1\rangle + |2\rangle + |3\rangle$  代表0、1、2、3四種狀態的疊加態，只需要兩個量子位元就可以用二進位表示。此時，若是針對這兩個量子位元進行一次函數計算，就能得到9、4、1、0四種狀態的疊加態，分別對應到四個輸入的函數值。

圖片來源：善科教育基金會

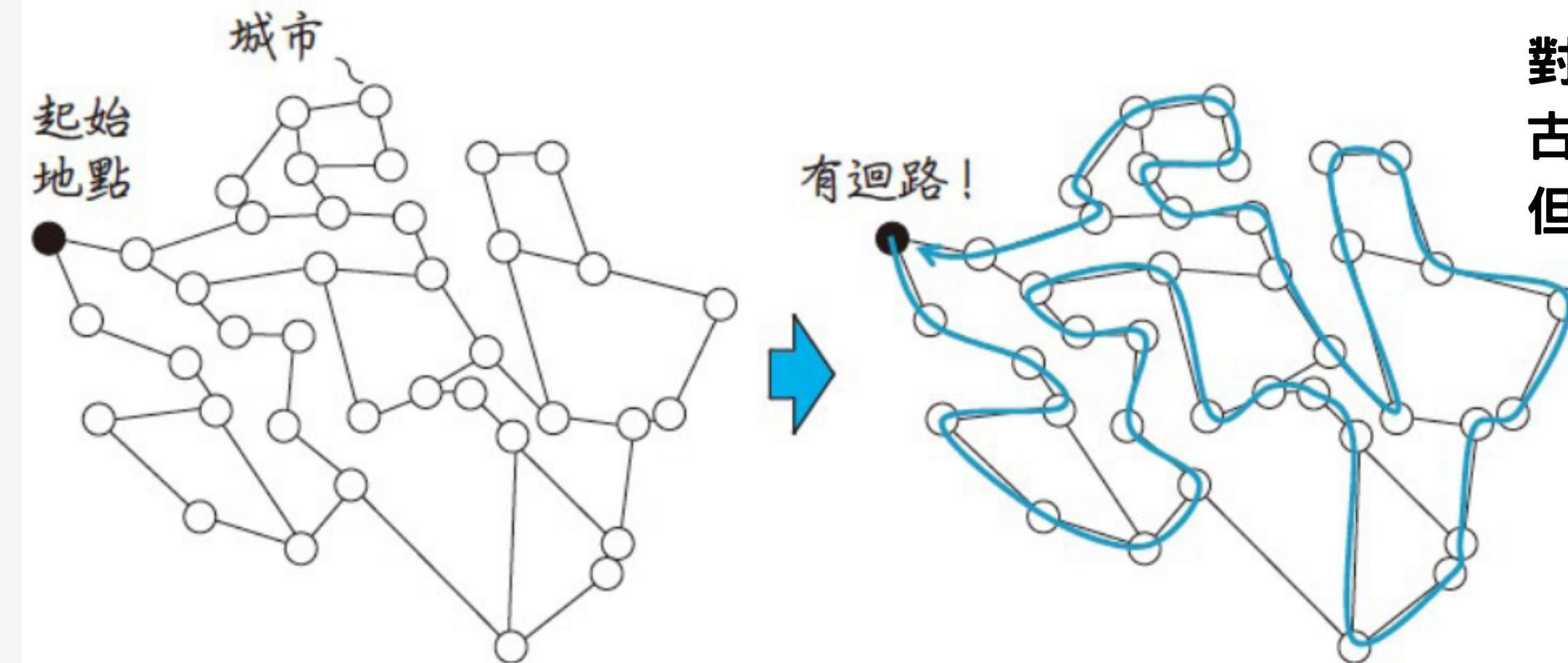
再利用"適當的"量子演算法（干涉）  
調控而得到疊加態中每一項的機率，以機率大的項作為我們的最終答案！

# 量子演算法

## GROVER演算法(格魯弗演算法)

特性：高速解決搜尋問題，例如：哈密盾循環問題

→ 研究「對於多個城市，是否存在從某地出發逐一經過各地一次又回到原地的循環(迴路)」的問題。



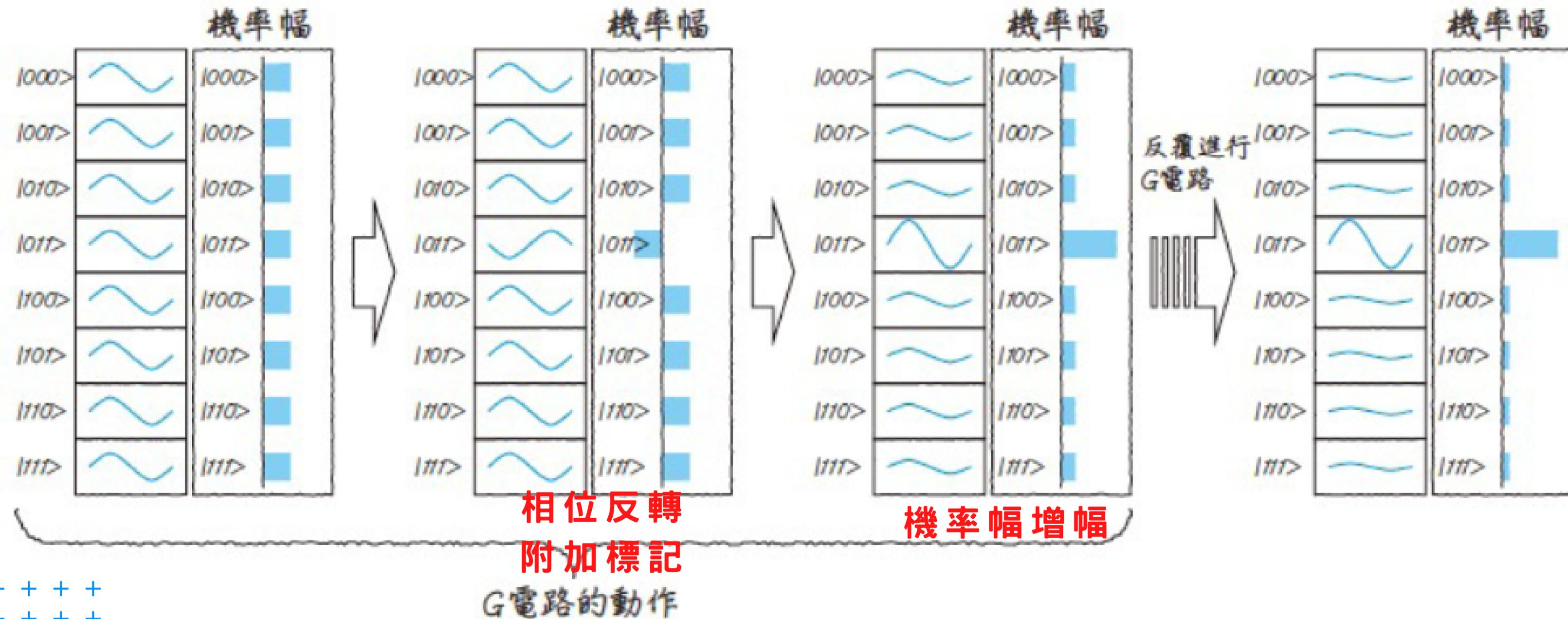
對於N種路徑，  
古典電腦需要N次驗證  
但是以GROVER演算法只需約  
 $\sqrt{N}$  次



# 量子演算法



## GROVER演算法(格魯弗演算法)



# 量子演算法

## SHOR演算法(秀爾演算法)

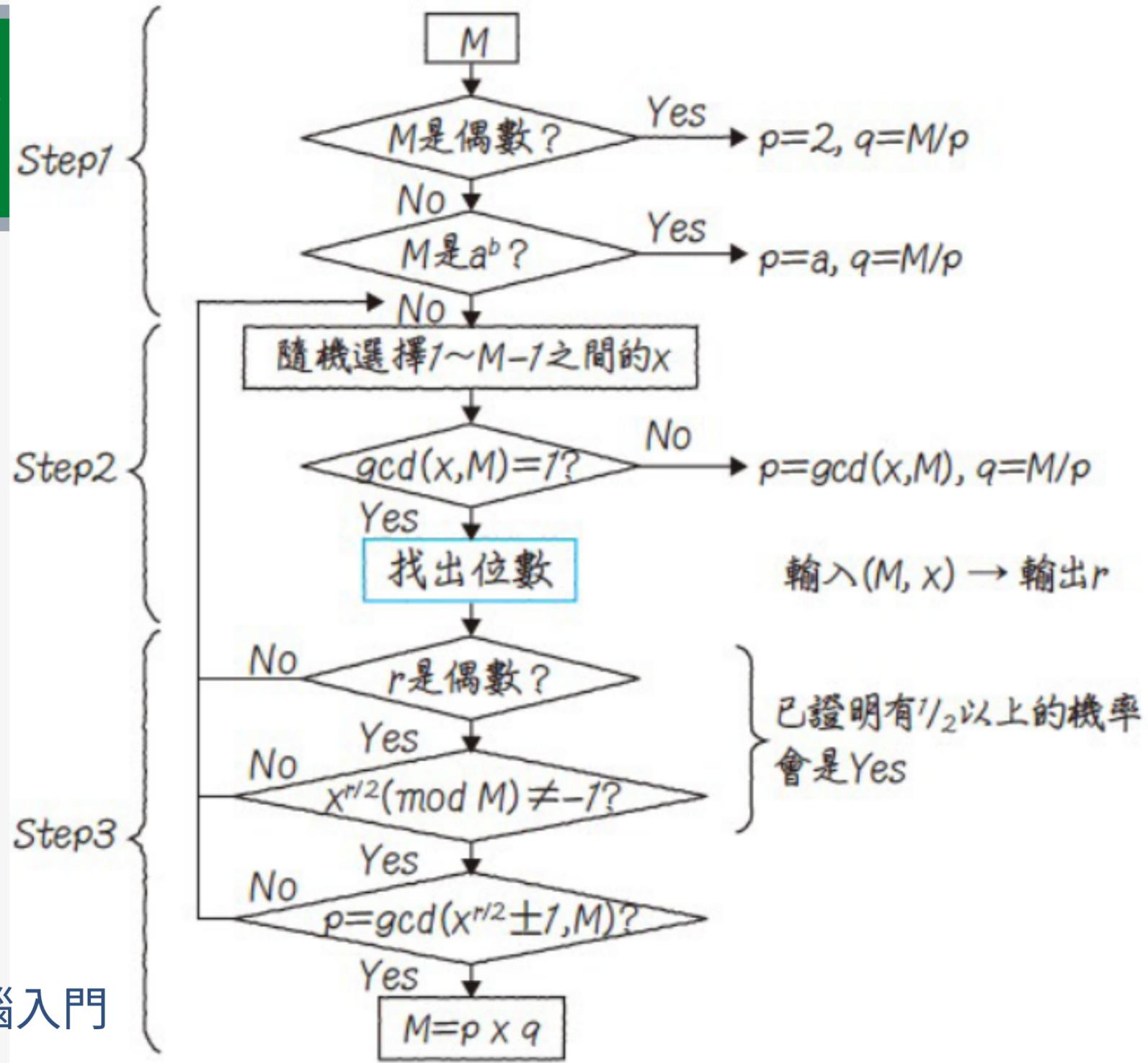
特性：加速質因數分解，撼動現代密碼系統  
→ 研究「高素質因數分解」的問題。

問題：給定一個合成數  $N$ ，  
找到整數  $1$  和  $P$  之間且不包含  $1$  和  $N$ ，  
並且  $N$  整除於  $P$ 。

第一步：以傳統電腦運作的簡化演算法，  
將因數分解簡化成搜尋階問題。

第二步：量子演算法，解決搜尋階問題。

目的：假設  $M = p \times q$  來求解  $p, q$





# 量子電腦的未來

## 八種行業的應用案例



### 發現新藥

對粒子在分子內相互作用的方式進行建模，創建一種能夠抵抗特定疾病的配方。

對一個只有**70**個原子的分子進行建模，一台傳統電腦需要計算長達**130**億年才能夠達成。

然而，量子電腦有可能在幾分鐘內即可解決分子模擬問題。因此，未來生物科技結合量子將成爲兵家必爭之地。





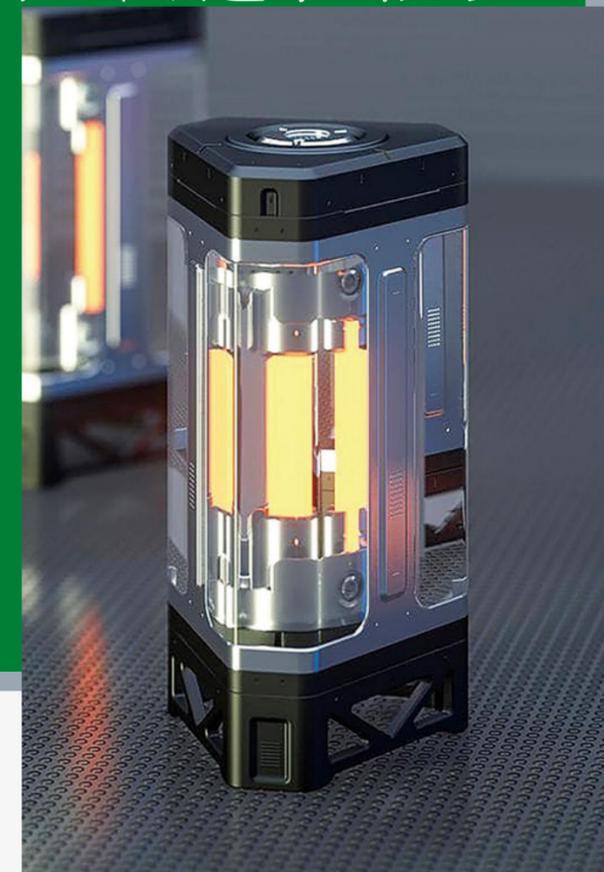
# 量子電腦的未來

## 八種行業的應用案例



### 製造高效能鋰硫電池

尋找具有更好性能的新材料來製造電池，已經成為電動車趨勢下的顯學。電池設計是一項數據非常繁重的工作，德國汽車製造商戴姆勒與**IBM**，正在合作評估量子電腦如何幫助其模擬硫分子在不同環境中的行為，最終目標是製造性能更好、壽命更長、更耐用的鋰硫電池。





# 量子電腦的未來

## 八種行業的應用案例



### 預測天氣

量子電腦能夠同時分析所有相關數據，會比現今天氣預報準確得多。這不僅有利於規劃人們的下一次戶外活動，亦可幫助政府更好地為自然災害做準備，並支持氣候變化研究。





# 量子電腦的未來

## 八種行業的應用案例

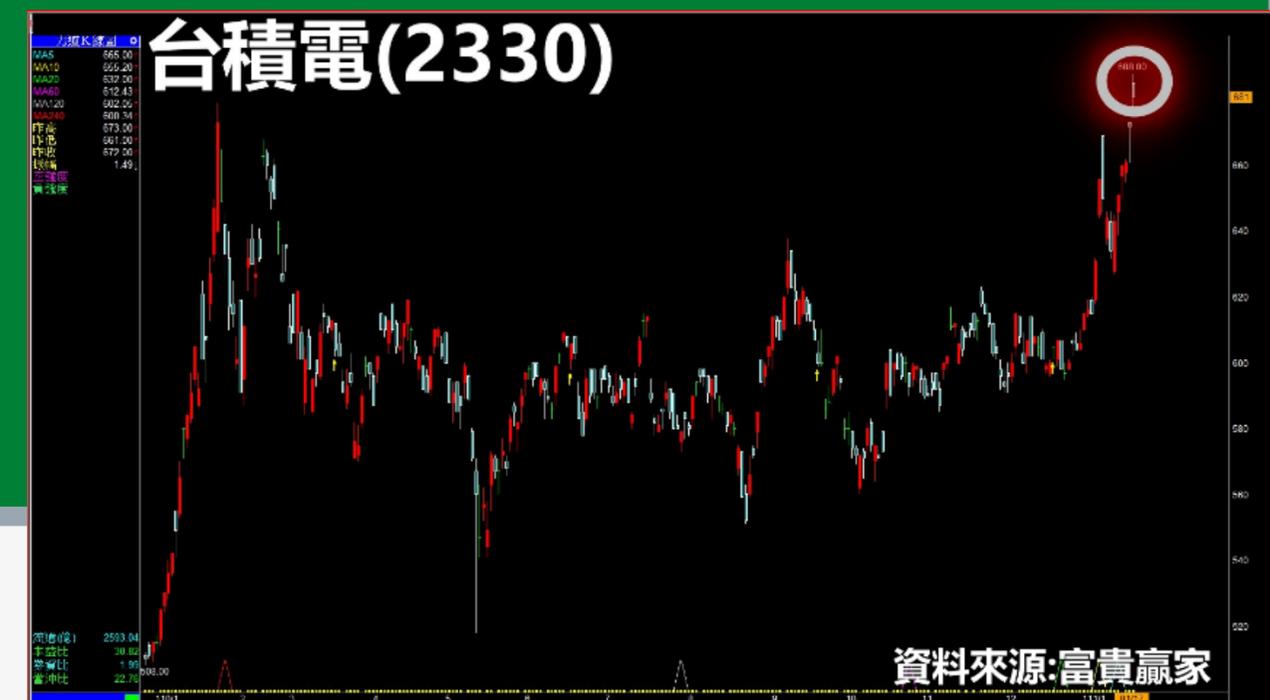


### 挑選股票

摩根大通、高盛和富國銀行積極研究量子電腦提高銀行營運效率，強化投資報酬。

將量子運算應用於稱爲 **MONTE CARLO** 模擬的程式。

根據高盛與量子運算公司 **QC WARE** 進行的研究，量子電腦前所未有的運算能力可以將 **MONTE CARLO** 運算速度提高 **1,000** 倍。





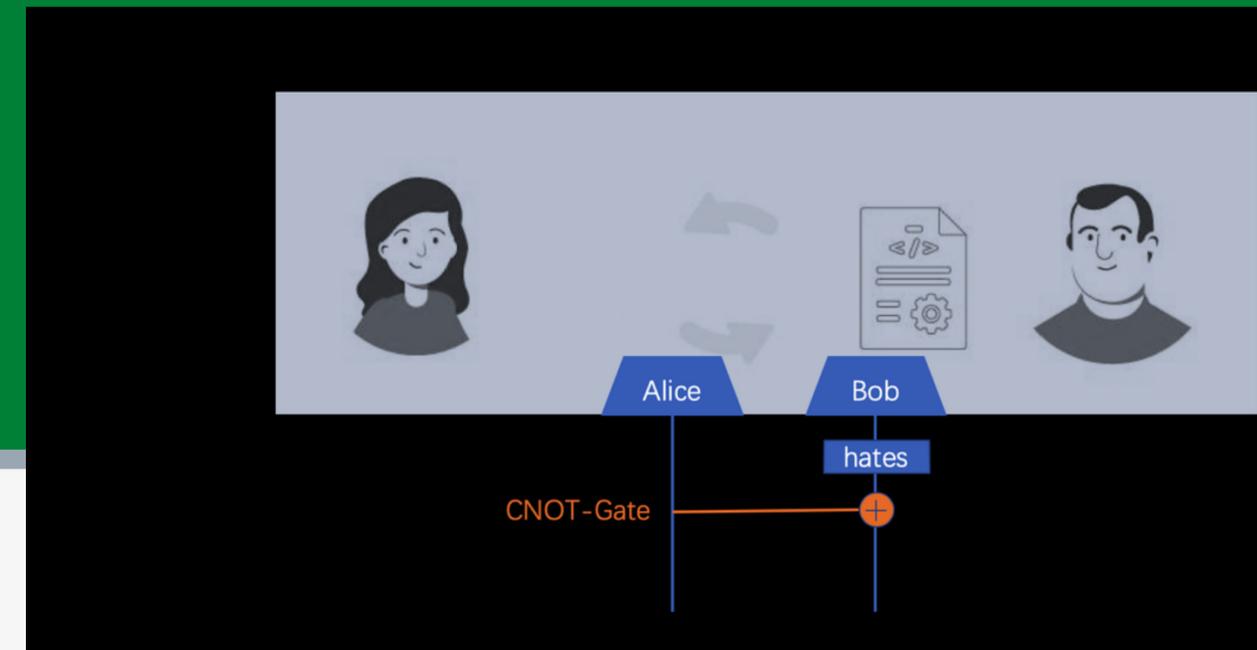
# 量子電腦的未來

## 八種行業的應用案例



### 量子自然語言處理

量子自然語言處理 (QNLP)，是劍橋量子運算的重點工作。該公司已經通過實驗證實，句子可以在量子電路上進行參數化，根據句子的語法結構嵌入詞義，強化電腦理解人們自然語言。





# 量子電腦的未來

## 八種行業的應用案例



### 協助解決運送問題

能源巨頭**EXXONMOBIL**一直在努力優化穿越海洋的商船之日常航線。畢竟這一運送價值高達**14**兆美元，如果能夠透過量子電腦尋找出可能的探索路線，將有助於降低航運成本，縮減運送時間。因此，**EXXONMOBIL**找上了**IBM**合作，研究證明，量子硬體一旦得到改善，量子演算法將能更提供更好的解決方案。





# 量子電腦的未來

## 八種行業的應用案例



### 減少交通壅塞

優化城市交通訊號的時間，幫助車輛流動順暢和避免繁忙路口的擁堵。  
微軟一直在與**TOYOTO TSUSHO**和量子運算新創公司**J1J**一起研究這個用例，期望將交通等待時間減少多達**20%**。





# 量子電腦的未來

## 八種行業的應用案例



### 保護敏感性數據

隨機性是量子行為的基本組成部分：例如，構成量子處理器的粒子之行為方式完全不可預測。因此，這種行為可用於確定即使使用最強大的超級電腦也無法進行逆向工程的加密密鑰。英國新創公司 **NU QUANTUM** 正在研究一系統可以測量量子粒子的行為以生成隨機流，然後可以使用這些隨機流來建構更強大的加密密鑰，保護敏感數據。





# 3A 中心動畫開發



## 人工智慧與量子電腦

- 一、 AI人工智慧介紹
- 二、 AI研究課題與機器學習
- 三、 監督式學習
- 四、 非監督式學習
- 五、 增強式學習
- 六、 深度學習
- 七、 AI發展現況、應用與挑戰
- 八、 量子電腦介紹
- 九、 量子物理 V.S. 古典物理
- 十、 量子資訊與傳輸
- 十一、 量子糾纏態與疊加態
- 十二、 量子演算法
- 十三、 量子技術實現與挑戰
- 十四、 量子電腦發展與應用



陳瑋彤





# 3A 中心動畫開發





# 量子計算



## 量子電腦的工作流程

### 定義問題

根據專業領域訂出演算規則並將問題系統化。

1

### 量子演算法

決定運用何種量子演算法與模型。

2

### 量子電路

產生基於量子邏輯閘的量子電路以解決複雜的計算問題。

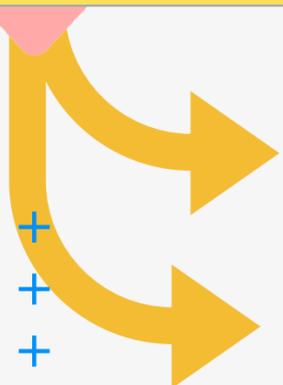
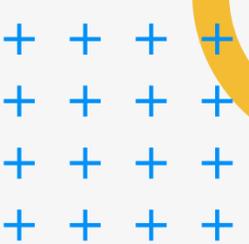
3

### 量子編譯器

將高階程式語言進行編譯，進而產生量子組合語言。

模擬器

量子電路



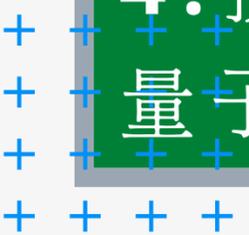


# 量子計算

## 量子模擬器



1. 由於建置一台量子電腦的成本極高，大部分系統都位於大學和研究實驗室。再加上真正的成熟量子電腦硬體大約**3-5**年的時間推進，因此，量子模擬器，就是一種可以在**傳統電腦上模擬量子電腦運算方式的程式**。
2. 這種模擬器的功能，有點類似**APP**的開發人員，在電腦上以**IOS** 模擬器、或 **ANDROID** 模擬器開發**APP**的方式是一樣的。
3. 量子模擬器是可在傳統電腦上執行的軟體程式，可以讓程式設計人員熟悉、預測量子位元將如何回應不同運算的任務，測試量子程式。
4. 預估到**2023**年，研究量子計算策略的組織中有**95%** 透過雲端運算平台使用量子電腦、使用雲端運算平台使用的程式語言，也是模擬器的概念。



# 量子計算

## HELLO QUANTUM

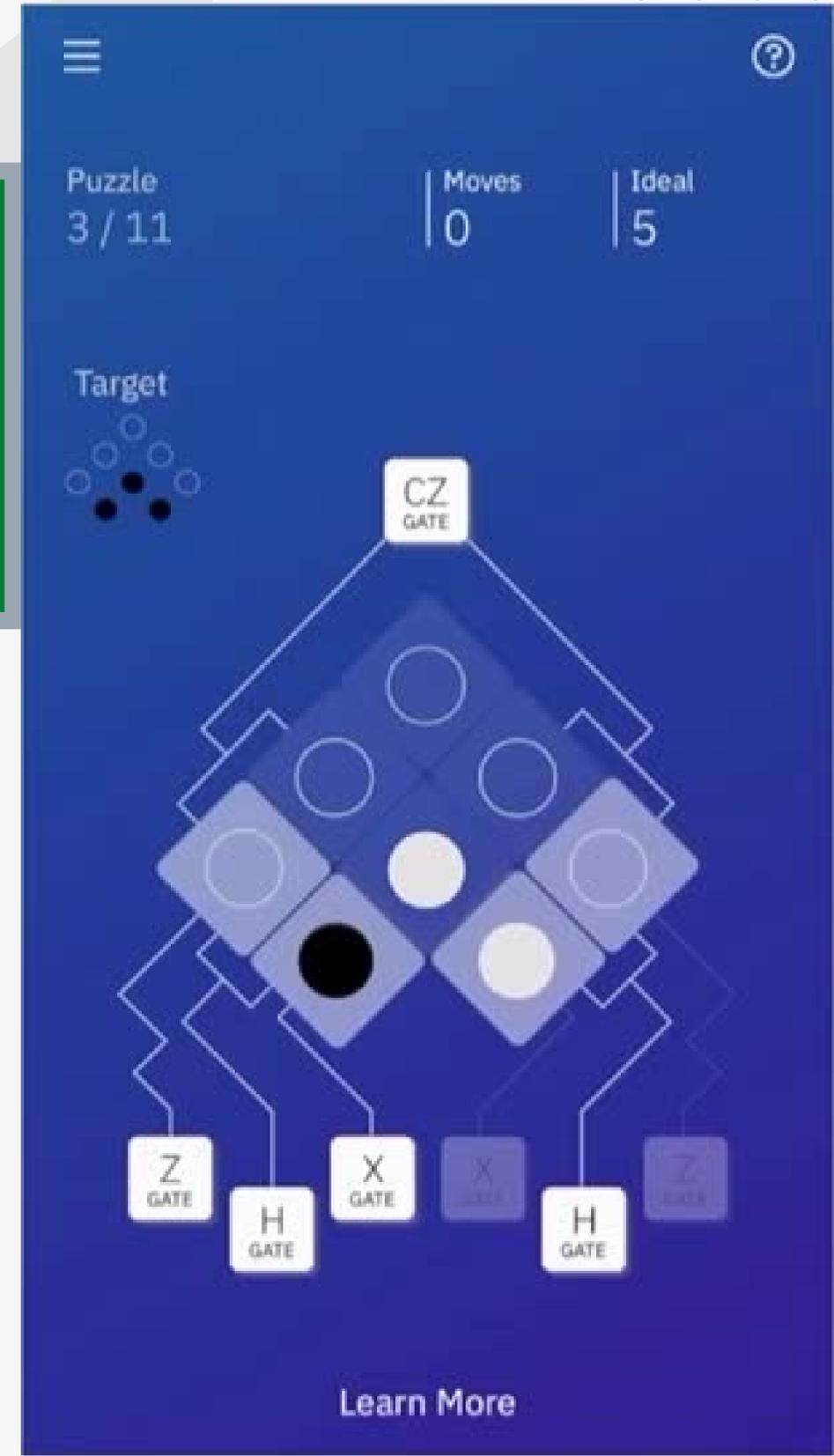
IBM與巴塞爾大學 (UNIVERSITY OF BASEL) 的研究團隊共同開發出了一款「HELLO QUANTUM」的量子解謎遊戲，能以遊戲的方式去體驗量子電腦的操作，也能從中認識量子計算的重要原理！



APPLE



ANDROID





# HELLO QUANTUM

開啟遊戲需要先「AGREE」同意相關的條款，接著就可以進入遊戲的主畫面。點擊「PLAY」進入關卡，當然也可以先到「HOW TO PLAY」裡了解一下。



## Terms and Conditions

### TERMS AND CONDITIONS

This License is solely between you and IBM. IBM is solely responsible for the App and its content.

APPS RECEIVED FROM IBM UNDER THESE TERMS AND CONDITIONS ARE LICENSED, NOT SOLD. BY INSTALLING AND USING THIS APP, YOU AGREE TO THESE TERMS. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER LICENSEE, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT LICENSEE TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS, DO NOT USE THE APP AND REMOVE IT FROM YOUR DEVICE;

#### 1. Definitions

"Authorized Use" - use on one mobile device running an applicable operating system. "IBM" - International Business Machines Corporation or one of its subsidiaries.

"App" - the accompanying software, including the original and all whole or partial copies: 1) machine-readable instructions and data, 2) components, files, and modules, 3) audio-visual content (such as images, text, recordings, or pictures), and 4) related licensed materials (such as keys and documentation).

#### 2. License Grant

The App is owned by IBM or an IBM customer and

Disagree

Agree

## Hello Quantum



Play

Levels & Scores

How to play

About



# 量子計算

## HELLO QUANTUM



Your puzzle is composed of various states.

States can either be on (white), off (black), or random (outline). Random states can become either "on" or "off."

-  On State
-  Off State
-  Random State

Back ... Next

Your game controls are called gates.

Each gate (X, Z, H and CZ) performs a different action upon the state. As you play your gates, keep track of how they affect the states.

-  X Gate
-  Z Gate
-  H Gate
-  CZ Gate

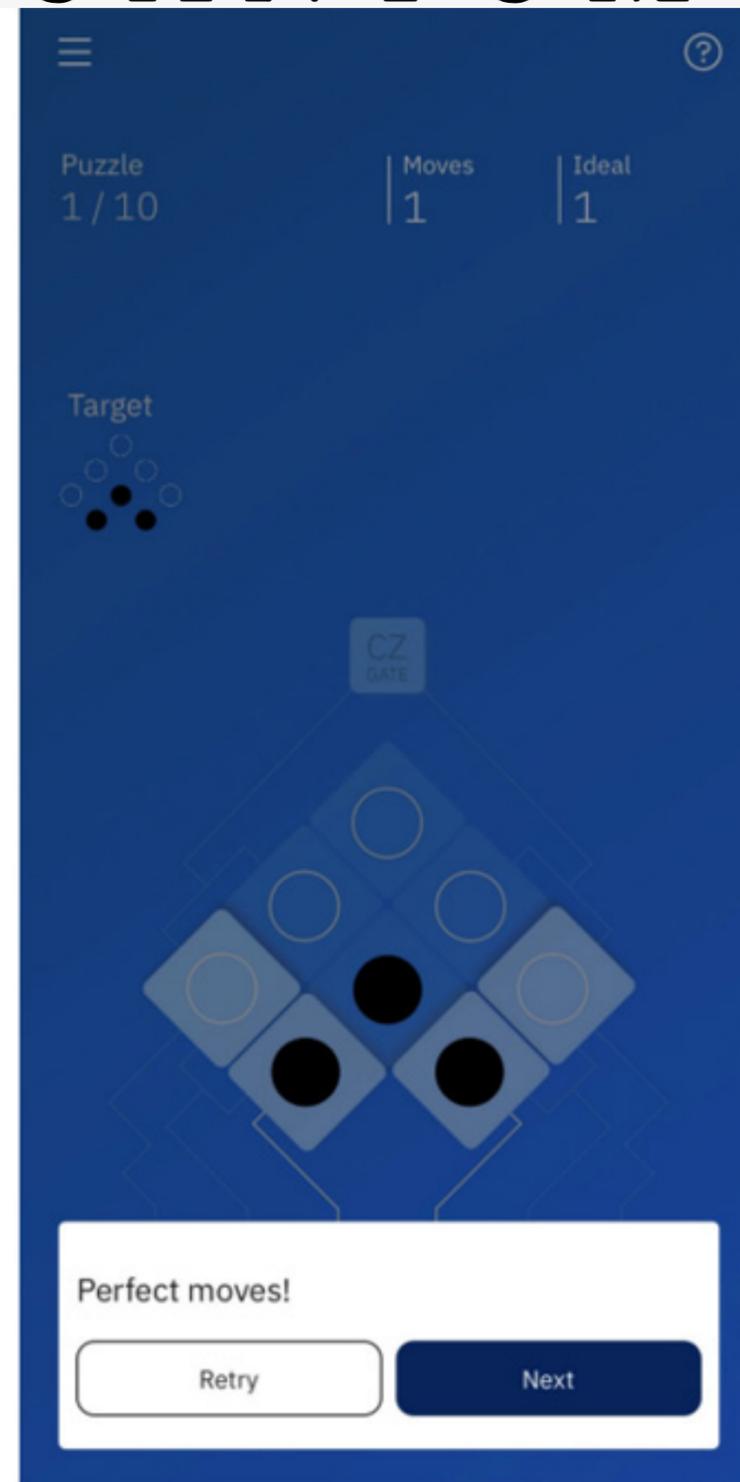
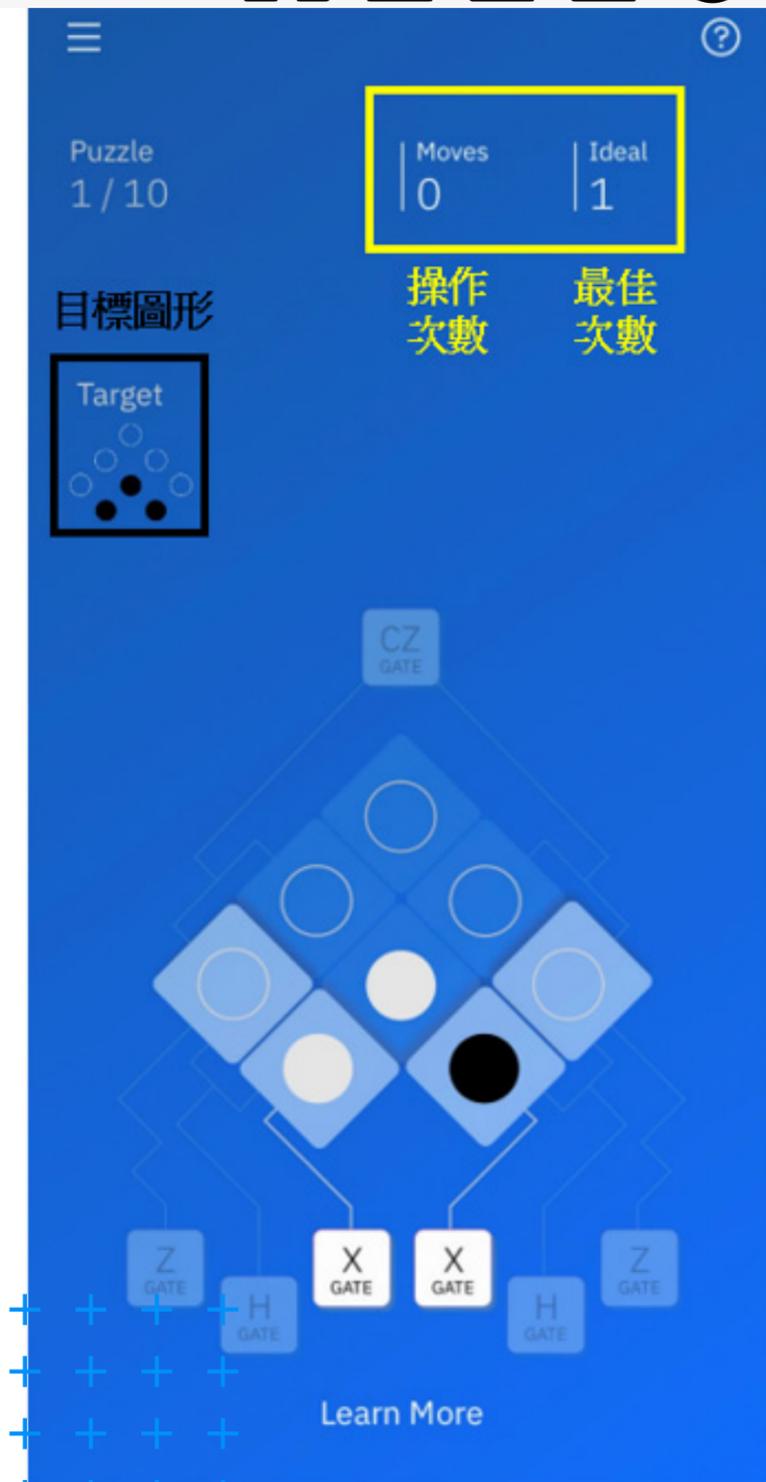
Back ... Next

每一道關卡會由下列三種不同的狀態 (STATE) 隨機組合，分別為：白色圓圈的「ON STATE」、黑色圓圈的「OFF STATE」以及透明圓圈的「RANDOM STATE」。其中RANDOM STATE可以轉換成ON STATE或OFF STATE。

可以透過操作量子閘 (GATE) 來改變狀態，遊戲中能使用的量子閘共分四種：X GATE、H GATE、Z GATE和CZ GATE，其中每一種量子閘都能對狀態執行不同的操作方式！

# 量子計算

## HELLO QUANTUM

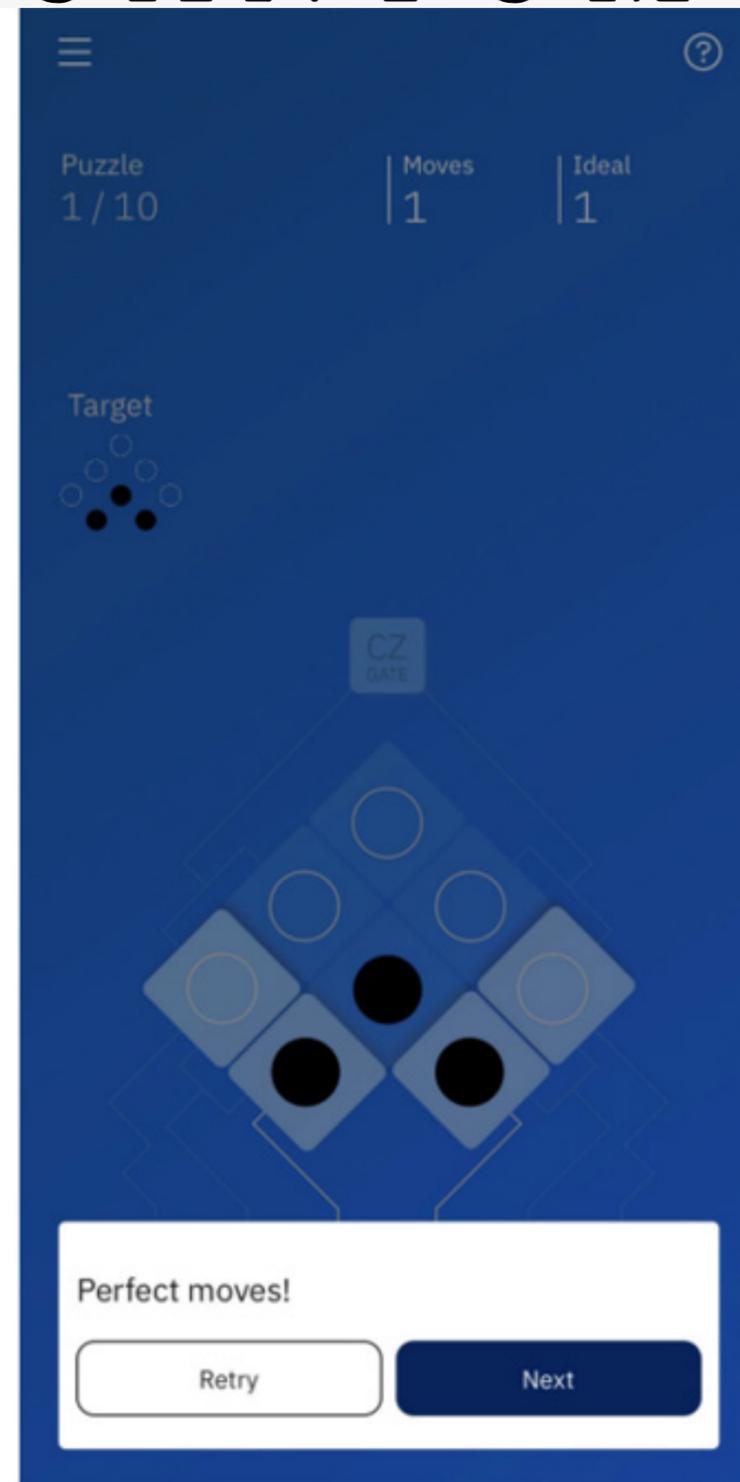
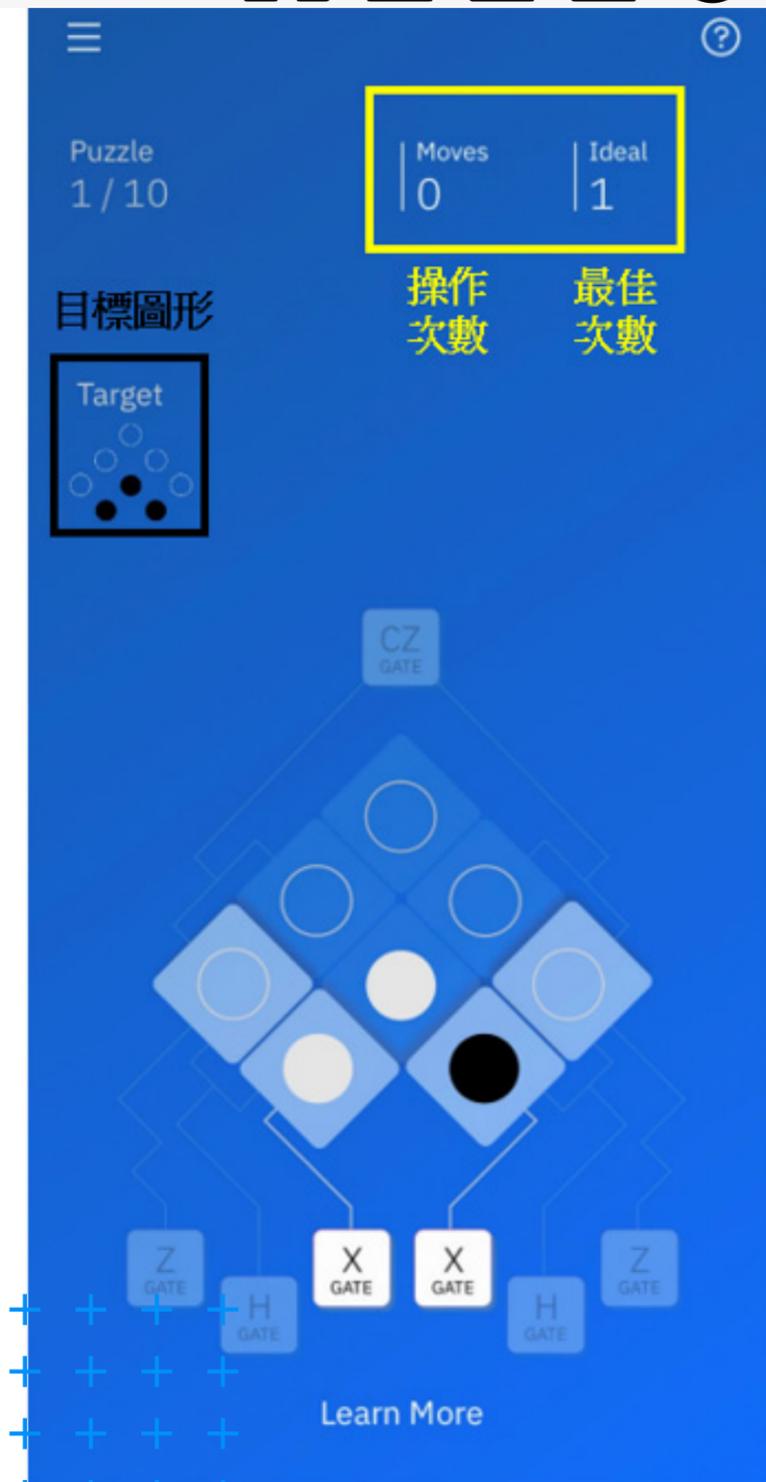


每一道關卡會由下列三種不同的狀態 (STATE) 隨機組合，分別為：白色圓圈的「ON STATE」、黑色圓圈的「OFF STATE」以及透明圓圈的「RANDOM STATE」。其中RANDOM STATE可以轉換成ON STATE或OFF STATE。

可以透過操作量子閘 (GATE) 來改變狀態，遊戲中能使用的量子閘共分四種：X GATE、H GATE、Z GATE和CZ GATE，其中每一種量子閘都能對狀態執行不同的操作方式！

# 量子計算

## HELLO QUANTUM



遊戲的每一道關卡會設計出一組目標圖形(黑色方框)

我們需要透過操作量子閘將初始狀態排列成跟目標狀態相同的圖形，即可解鎖下一關。

遊戲的右上方會顯示目前的操作次數以及完成所需的最少次數，盡量嘗試用最少的操作次數來完成每一關！



# 量子位元的表示法

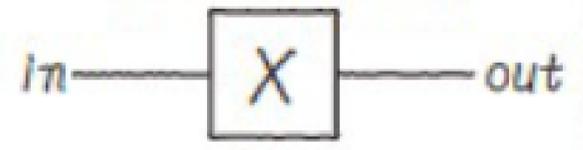


# 量子電腦計算

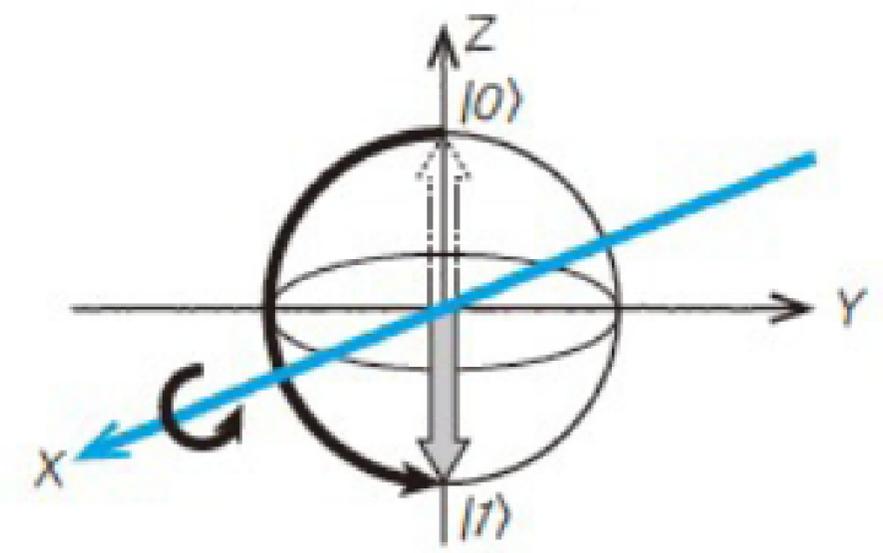


## 量子閘 (單一量子位元)

X閘  
(位元反轉閘)



in	out
$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$



以X軸為中心進行180度旋轉操作

Y閘  
(相位位元反轉閘)

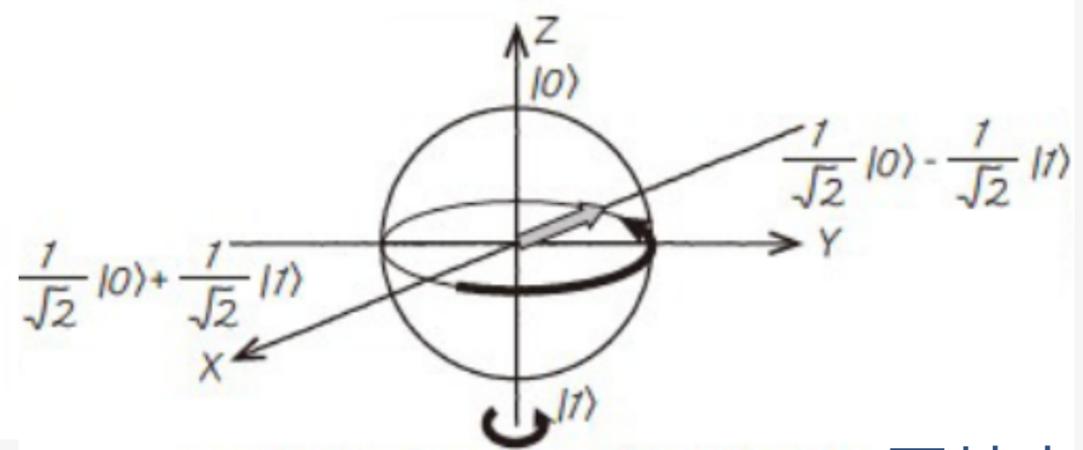


in	out
$ 0\rangle$	$i 1\rangle$
$ 1\rangle$	$-i 0\rangle$

Z閘  
(相位反轉閘)



in	out
$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$- 1\rangle$



以Z軸為中心進行180度旋轉操作 圖片來源：圖解量子電腦入門



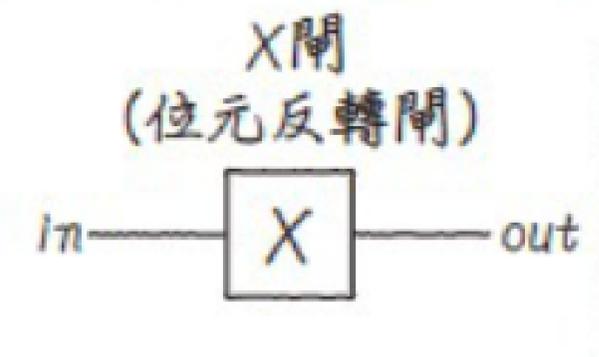
# 量子位元的表示法



# 量子電腦計算



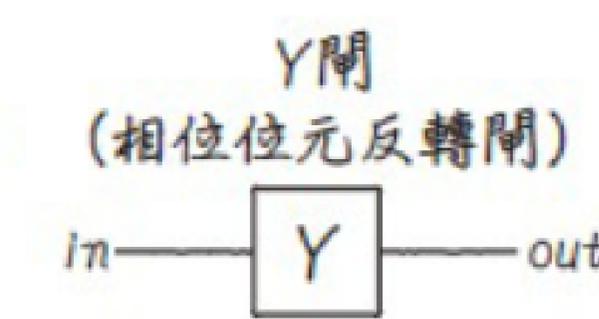
## 量子閘 (單一量子位元)



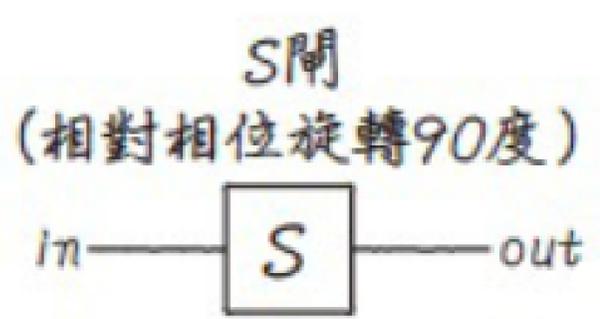
in	out
$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$



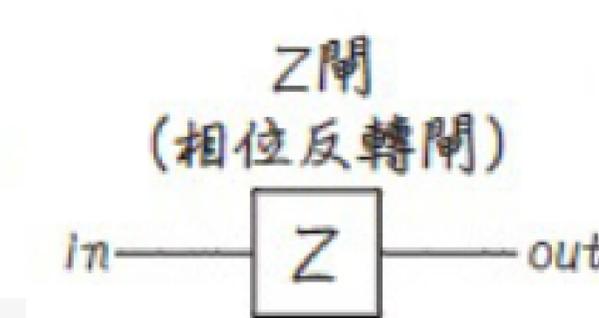
in	out
$ 0\rangle$	$\frac{ 0\rangle +  1\rangle}{\sqrt{2}}$
$ 1\rangle$	$\frac{ 0\rangle -  1\rangle}{\sqrt{2}}$



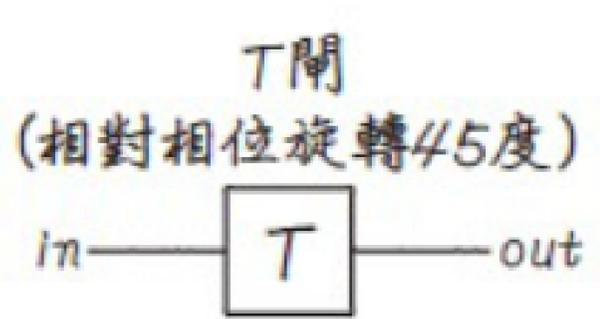
in	out
$ 0\rangle$	$i 1\rangle$
$ 1\rangle$	$-i 0\rangle$



in	out
$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$i 1\rangle$



in	out
$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$- 1\rangle$



in	out
$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$e^{i\pi/4} 1\rangle$

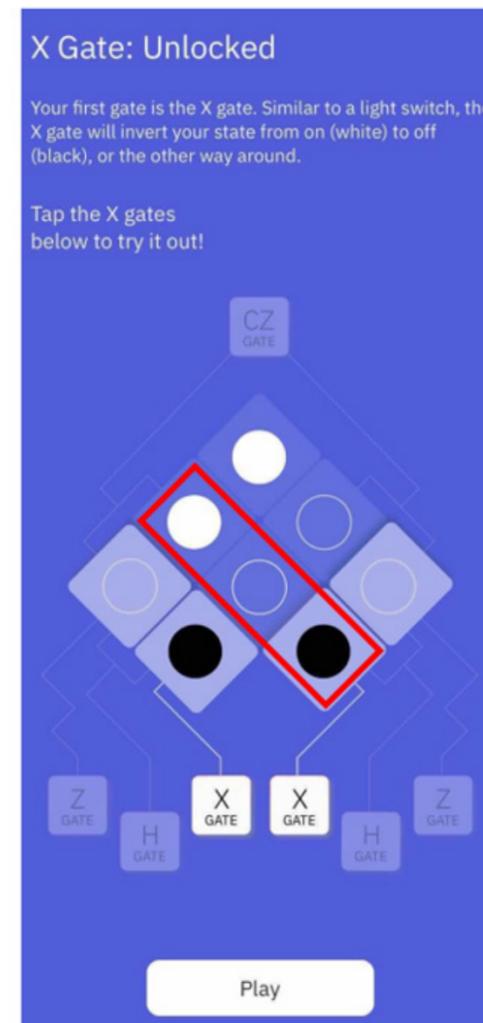
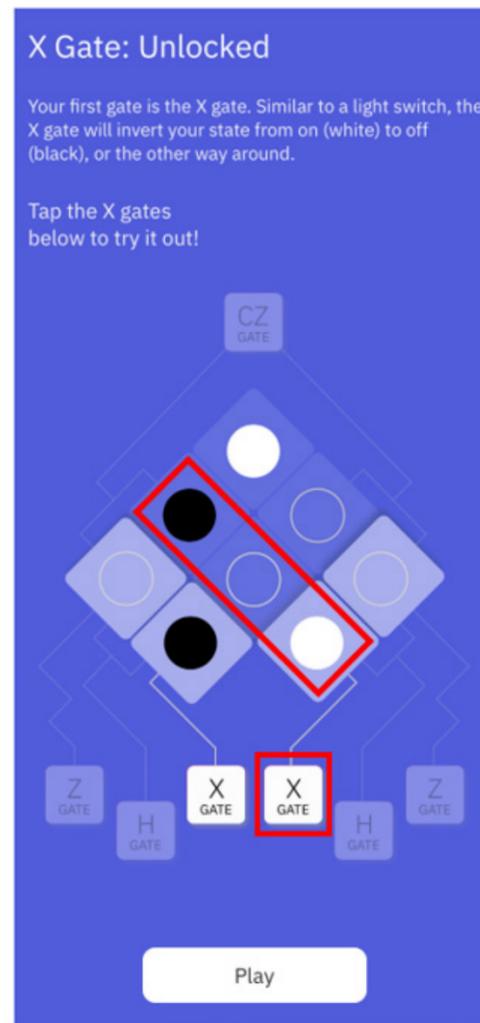
# 量子計算遊戲 APP

## HELLO QUANTUM



**X GATE**：將 ON STATE 切換至 OFF STATE (或是將 OFF STATE 切換至 ON STATE)，但並不會對 RANDOM STATE 造成影響。

黑色圓圈會轉換成白色圓圈，白色圓圈則會轉換成黑色圓圈 (如圖中紅色方框所示)，而透明圓圈不會有任何變化。



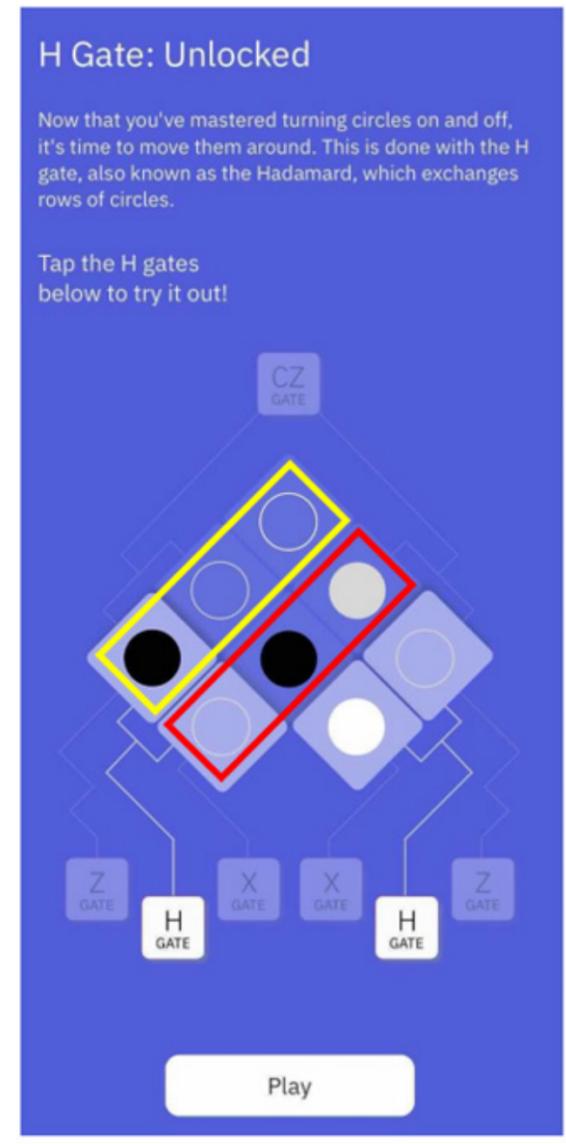
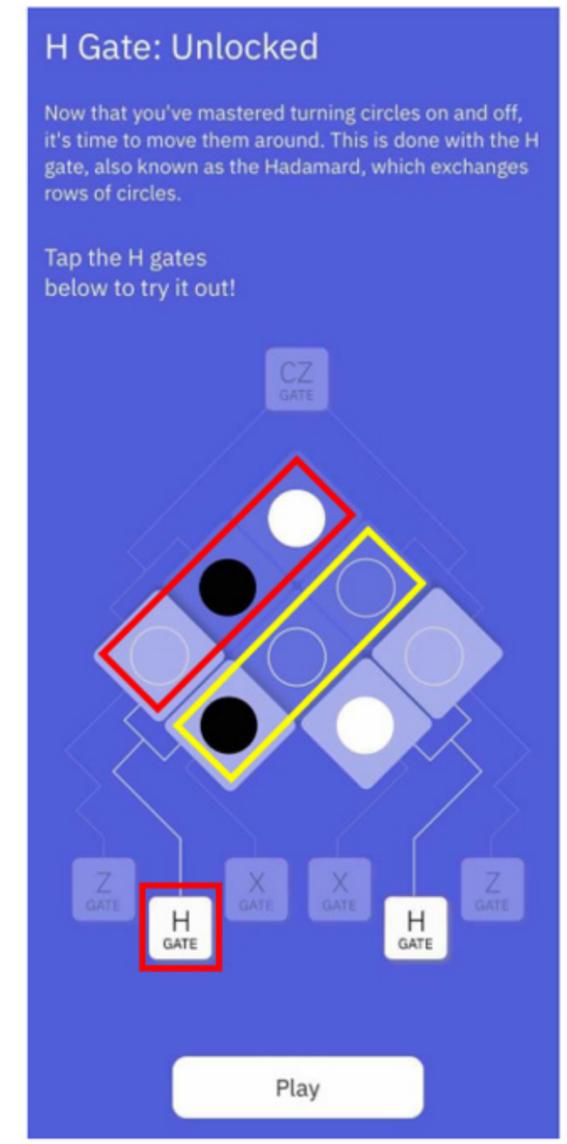
# 量子計算遊戲 APP

## HELLO QUANTUM



H GATE : HADAMARD GATE .

連接於H GATE的兩排狀態就會相互交換(紅色及黃色方框)。





Z GATE：作用類似於X GATE，但Z GATE控制的是不同列上的狀態

Z Gate: Unlocked

The X gate is not working in these puzzles. Fortunately you can use Z instead. These gates work just like X, but on different rows.

Tap the Z gates below to try it out!

Play

Z Gate: Unlocked

The X gate is not working in these puzzles. Fortunately you can use Z instead. These gates work just like X, but on different rows.

Tap the Z gates below to try it out!

Play

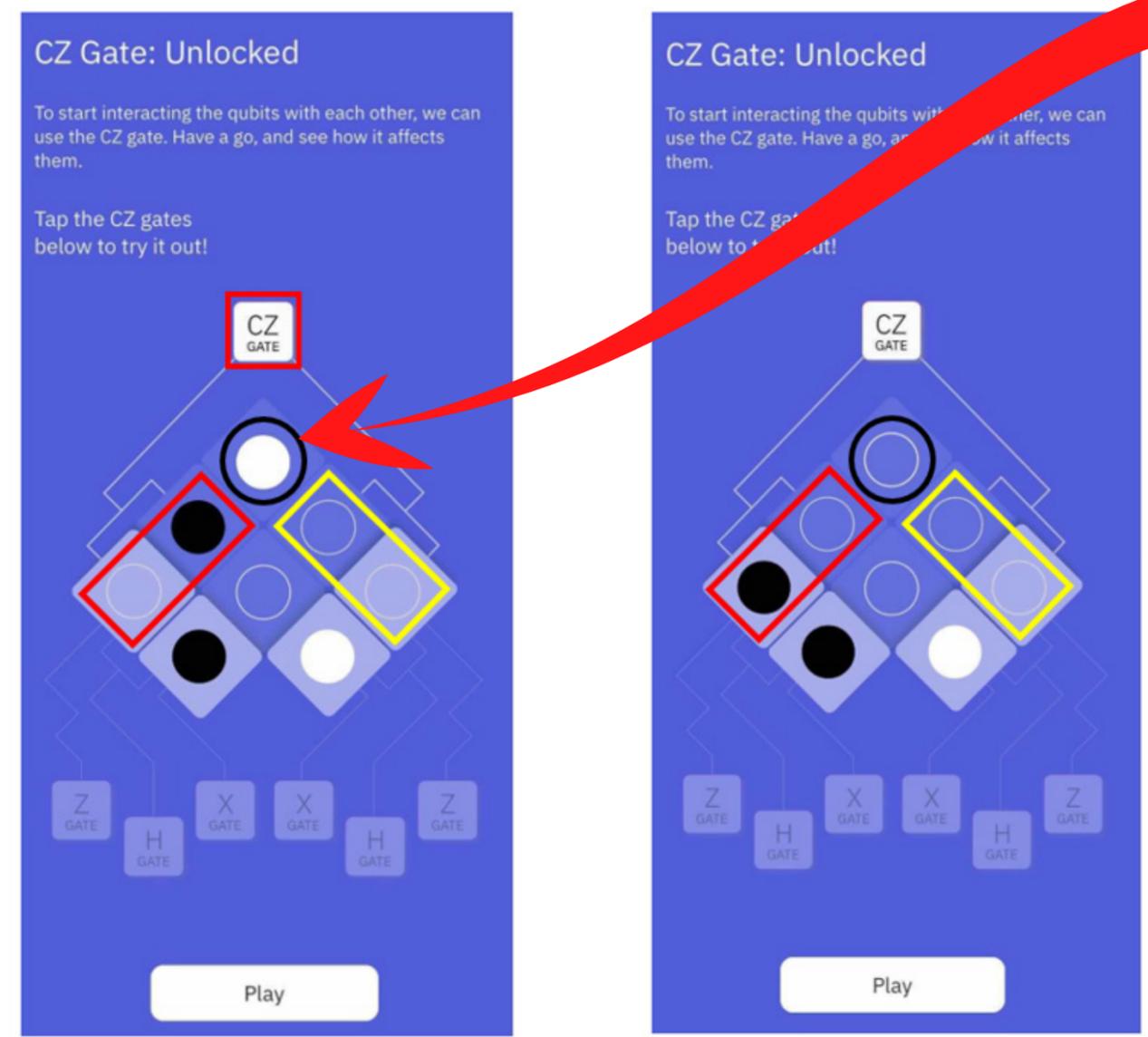


# 量子計算遊戲 APP

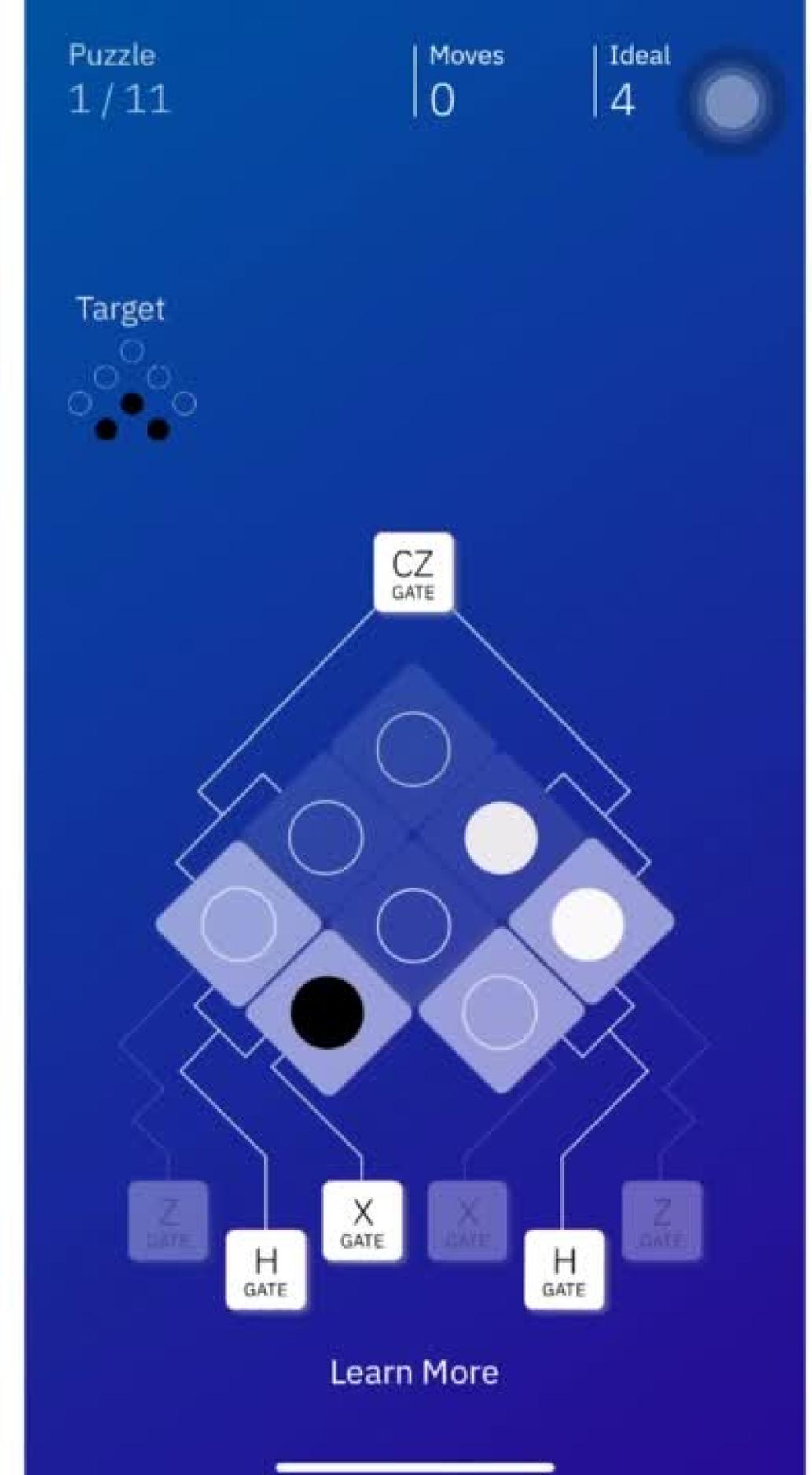
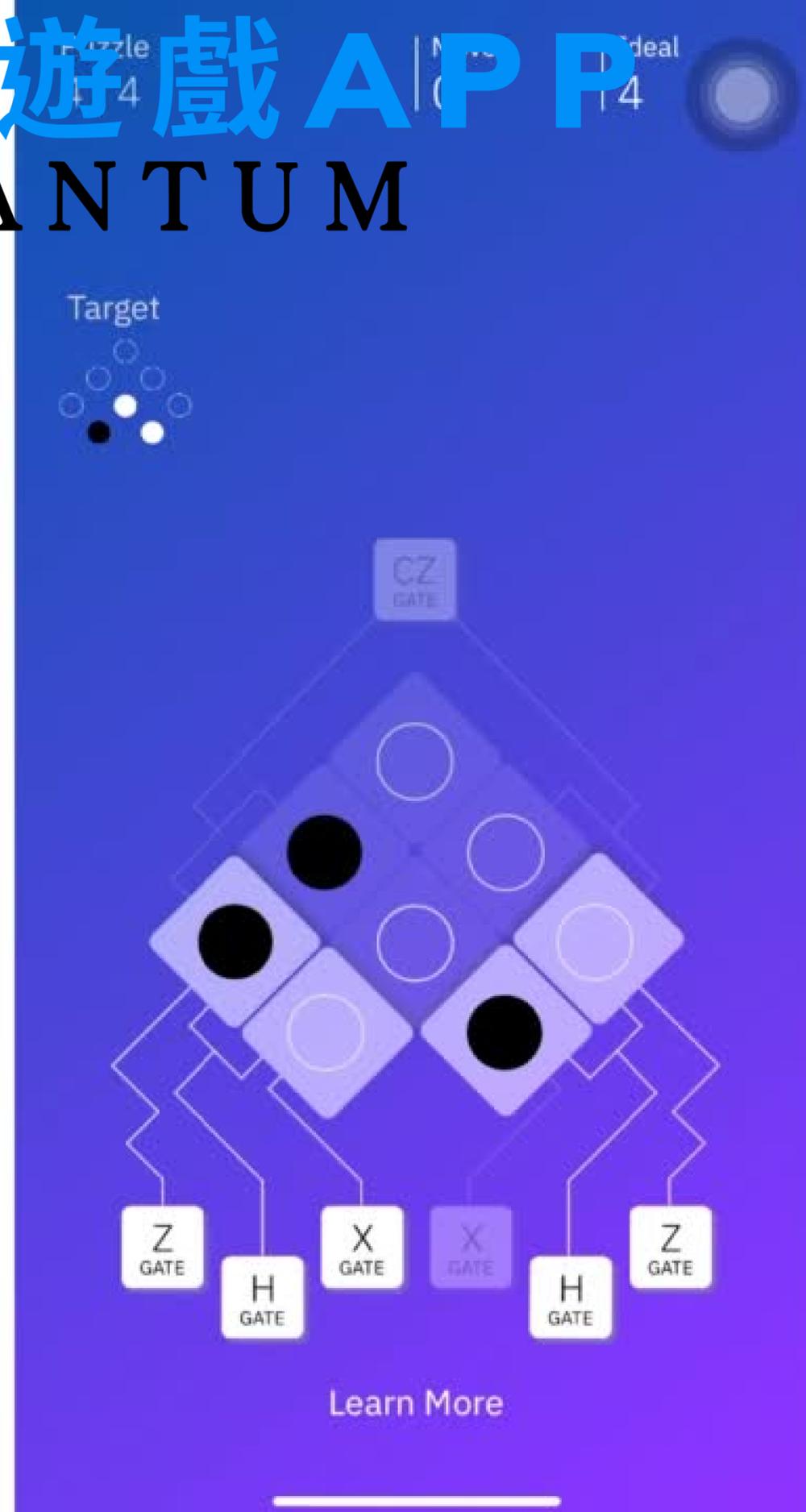
## HELLO QUANTUM



**CZ GATE :** CZ GATE 影響的狀態共有五個，線路直接連接到的兩組狀態 (紅色和黃色方框) 會各自交換。需要注意的是，最靠近 CZ GATE 的狀態 (黑色圓形)，會進行的狀態轉換。



操作  
示範影片



影片來源：量子線上學院



# 量子計算桌遊

## IBM ENTANGLION



All Search



BoardGameCandy.com



SHOP MAKE NEWS COMMUNITY HELP DEAL OF THE DAY CROWD SALES

Not Signed In



歡迎來到量子宇宙，船長！

恭喜，你的船長已經退休，讓你接替負責銀河航運業務！現在是時候進行一些升級了，您將踏上重建由古代種族開發的量子計算機的旅程。

ENTANGLION 是一款為兩名玩家設計的合作棋盤遊戲。

了解量子計算，與你的隊友一起在量子宇宙的三個星系中航行，避免被古人留下的防禦機制發現，並重建量子計算機。

認為你準備好迎接挑戰了嗎？

ENTANGLION

Buy for \$55.99





# 量子計算桌遊

## IBM ENTANGLION



# IBM ENTANGLION 桌遊套件下載



ENTANGLION

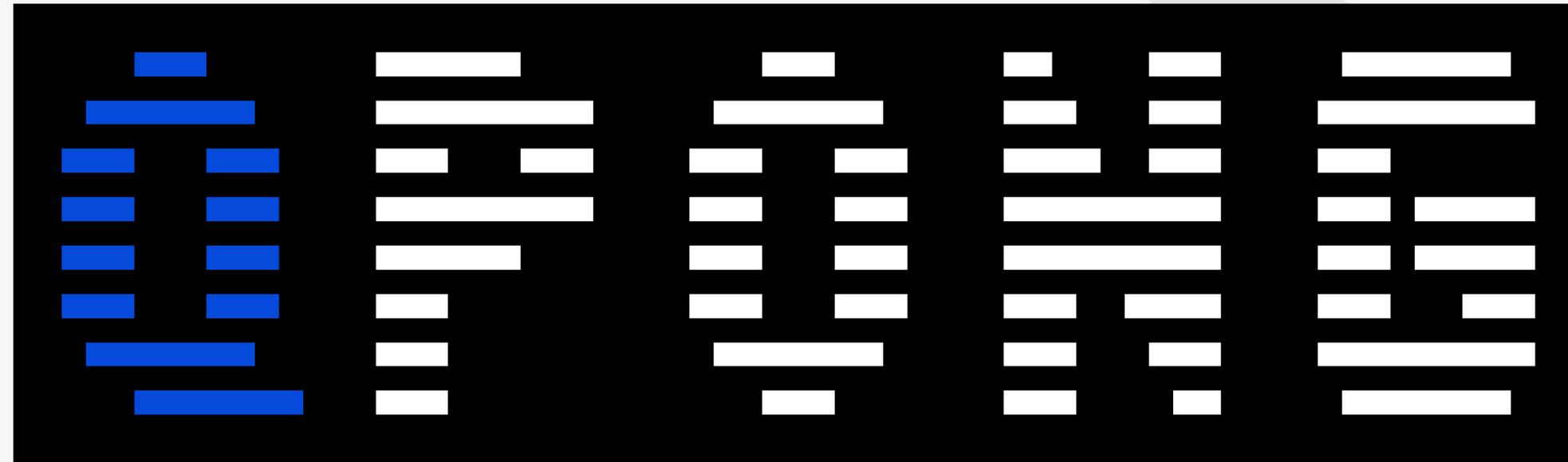
 Buy for \$55.99





# 量子計算電腦遊戲

## QPONG



在量子時代的黎明，一台原始的 3 量子比特量子計算機正試圖挑戰計算機帝國的長期統治者經典計算機。你的任務是利用你的人類智能幫助量子計算機擊敗經典計算機，並在人類歷史上首次展示“量子霸權”。經典量子大戰的戰場，莫過於經典的 QPONG 遊戲。





# 量子計算電腦遊戲

## QPONG



Product ▾ Team Enterprise Explore ▾ Marketplace Pricing ▾

Search



Sign in

Sign up

HuangJunye / QPong-Unity Public

Notifications

Fork 4

Star 17

Code Issues 18 Pull requests Actions Projects 2 Wiki Security Insights

master 6 branches 1 tag

Go to file

Code ▾

HuangJunye Merge pull request #67 from HuangJunye/develop ... 6f228a7 on 23 Aug 2020 238 commits

QPong-Python-Server	added the hold down of start to quit out of game	3 years ago
QPong-Unity-Version	open project in the latest Unity version 2019.4.8f1	2 years ago
.gitattributes	Initial commit	3 years ago
.gitignore	Update GUI properly	3 years ago
Logo.png	Add QPong logo for README	3 years ago
Readme.md	Update Readme.md	3 years ago

Readme.md

### About

QPong rewritten in Unity

game unity pong quantum ibm

Readme

17 stars

3 watching

4 forks

### Releases 1

Qiskit Camp Europe Version Latest  
on 13 Sep 2019





# 量子計算電腦遊戲

## QPONG



### Installation

Download the last [release](#) and unzip the files to the "Downloads" folder on your computer. You need to run the `QPong-Python-Server` before running QPong game. To do that you need to install Python and three required packages. You will also need to run the server in command line tool (Command Prompt for Windows or Terminal for macOS).

#### Install Python

If you don't have Python installed on your computer, you can install Python from <https://www.python.org/> or install Anaconda from <https://www.anaconda.com>

#### Open command line tool

On Windows, open Command Prompt by typing "Command Prompt" on the search box of Start menu. Check this link if you are not sure about how to do that: <https://www.wikihow.com/Open-the-Command-Prompt-in-Windows>

On macOS, press Command + Space to open Spotlight. Type "Terminal" on Spotlight to open Terminal. Check this link if you are not sure about how to do that: <https://www.wikihow.com/Open-a-Terminal-Window-in-Mac>

#### Install required packages

The Python libraries that you need to use are `qiskit`, `flask`, and `json_tricks`. To install these python libraries, `cd` to `QPong-Python-Server` in the command line and run

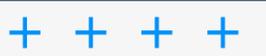
```
pip install -r requirements.txt
```

The Python server lives in a subfolder of the project. `cd` to `QPong-Python-Server/flask_server` and run

```
python3 server.py
```



Run the game and have fun!



### 安裝

下載最新版本並將文件解壓縮到計算機上的“下載”文件夾。您需要 `QPong-Python-Server` 在運行 QPong 遊戲之前運行。為此，您需要安裝 Python 和三個必需的軟件包。您還需要在命令行工具（Windows 的命令提示符或 macOS 的終端）中運行服務器。

#### 安裝 Python

如果您的計算機上沒有安裝 Python，您可以從 <https://www.python.org/> 安裝 Python 或從 <https://www.anaconda.com> 安裝 Anaconda

#### 打開命令行工具

在 Windows 上，通過在開始菜單的搜索框中鍵入“命令提示符”來打開命令提示符。如果您不確定如何操作，請查看此鏈接：<https://www.wikihow.com/Open-the-Command-Prompt-in-Windows>

在 macOS 上，按 Command + Space 打開 Spotlight。在 Spotlight 上鍵入“終端”以打開終端。如果您不確定如何操作，請查看此鏈接：<https://www.wikihow.com/Open-a-Terminal-Window-in-Mac>

#### 安裝所需的軟件包

您需要使用的 Python 庫是 `qiskit`、`flask` 和 `json_tricks`。要安裝這些 python 庫，`cd` 請 `QPong-Python-Server` 在命令行中運行

```
pip install -r requirements.txt
```

Python 服務器位於項目的子文件夾中。運行 `cd QPong-Python-Server/flask_server`

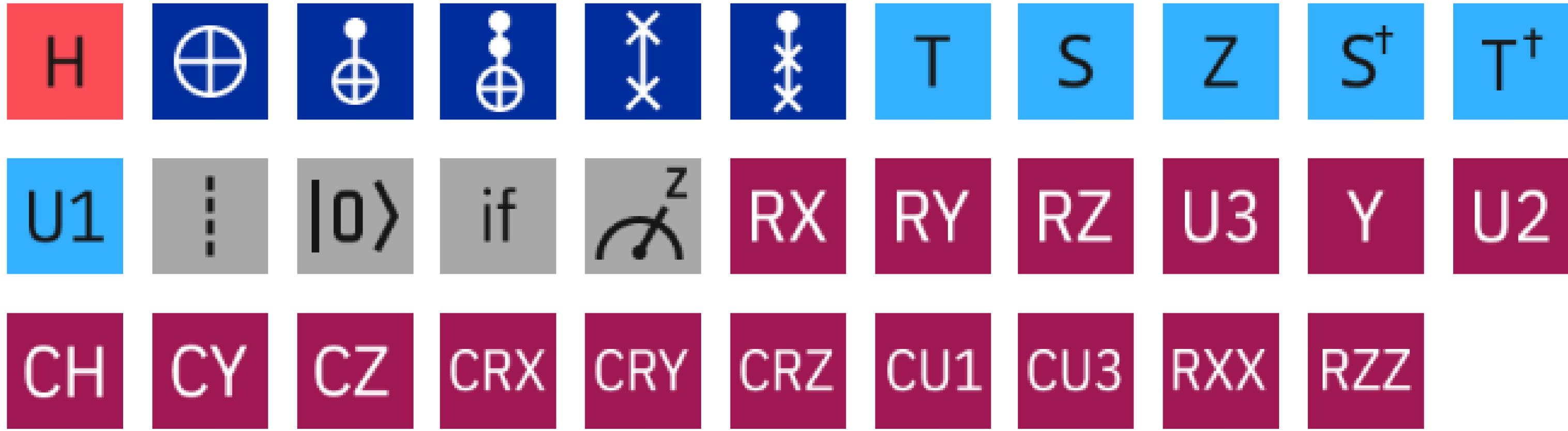
```
python3 server.py
```

運行遊戲，玩得開心！

# 量子計算

## IBM QUANTUM EXPERIENCE

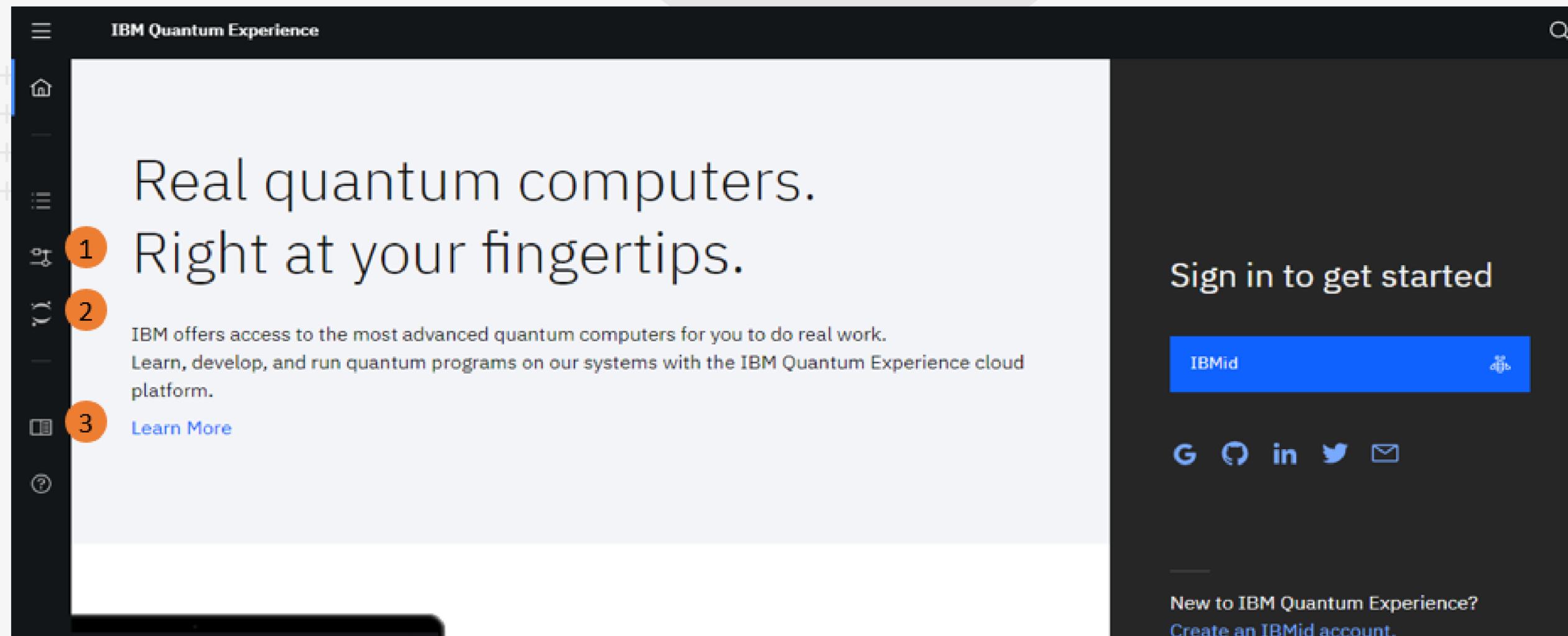
常見的量子閘有HADAMARD GATE、PAULI GATE、PHASE GATE、DAGGER GATE、CNOT GATE等。





# IBM QUANTUM EXPERIENCE

平台共有提供2種系統讓大家可以選擇，一個是圖型化介面，一個則是結合高階語言。



1. **CIRCUIT COMPOSER**：提供透過圖形化介面直接進行量子計算。
2. **QUANTUM LAB**：提供透過**PYTHON**及**QISKIT**進行量子計算。
3. **DOCS**：線上說明文件。



# 量子計算 IBM QUANTUM EXPERIENCE



File Edit Inspect View Help

1. 功能列

2. 執行程式

Run on ibmq\_qx3as

Circuit 3. titled circuit saved

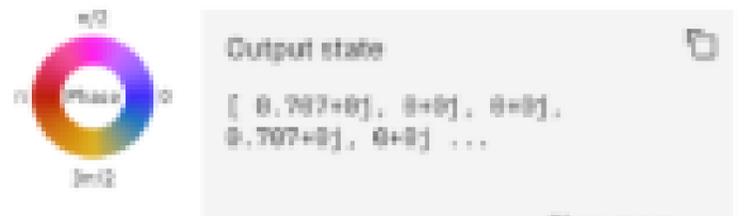
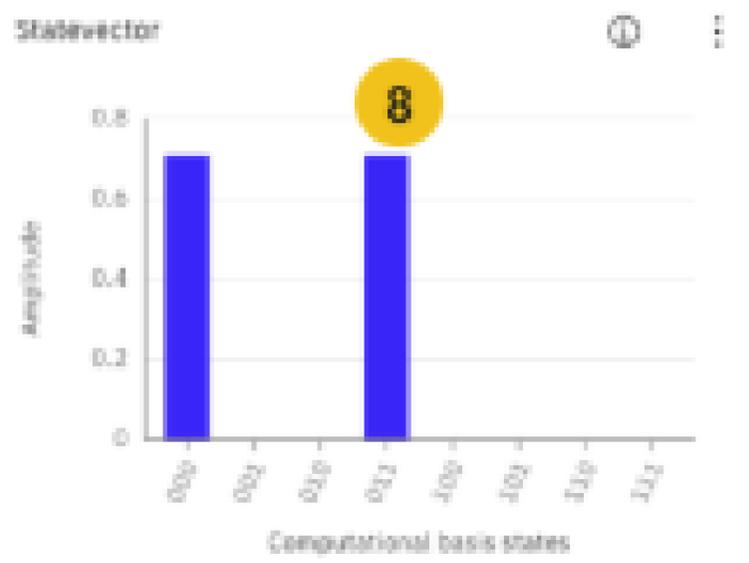
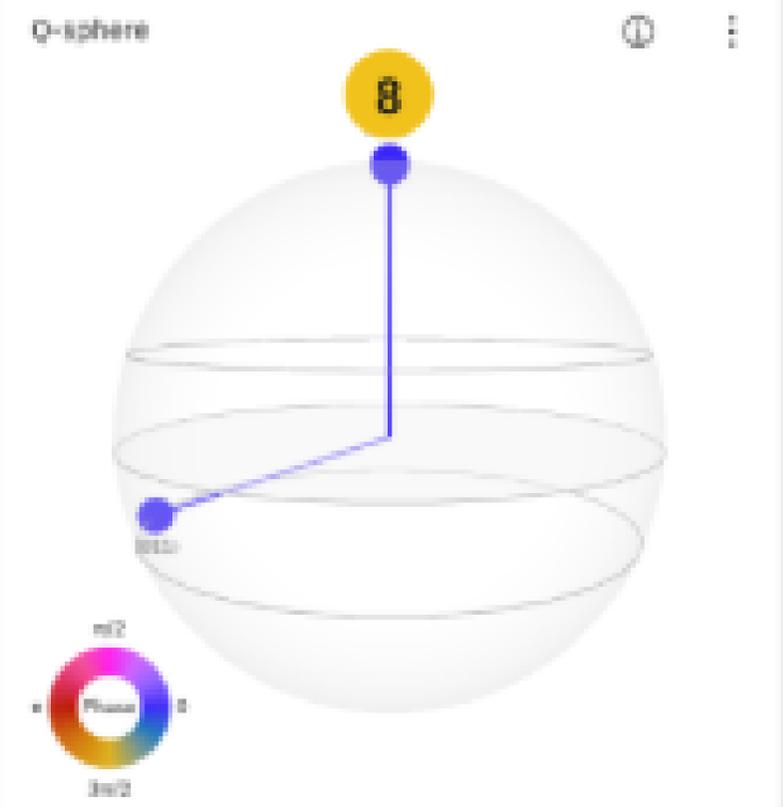
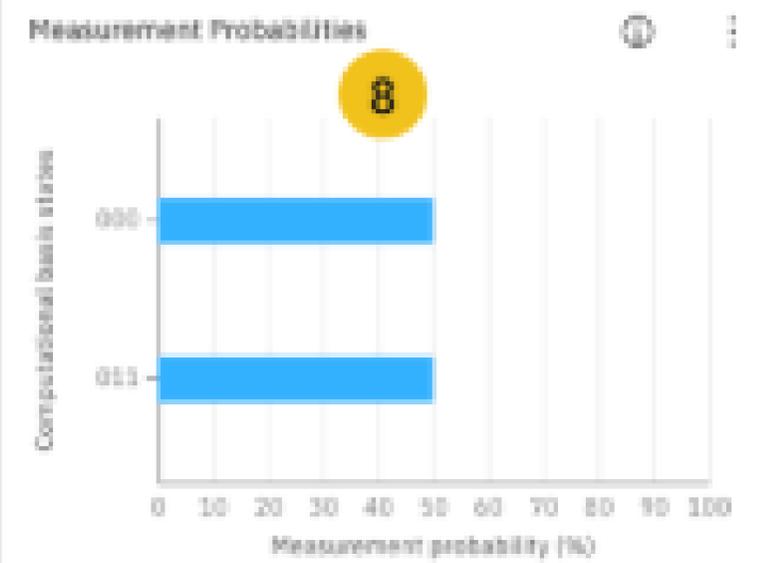
4. </> Code Docs Jobs

3. 開啟及編輯量子線路 (QUANTUM CIRCUIT) :  
所謂量子線路即由數個量子閘組合所形成的運算元件，由於圖形和傳統電子電路相似，故稱量子線路。此處我們可以開啟及編輯已儲存的量子線路。



Code editor Open in Quantum Lab

```
QASM
1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 qreg q[3];
5 creg c[3];
6
7 h q[0];
8 cx q[0],q[1];
9 measure q[0] -> c[0];
10 measure q[1] -> c[1];
```





# 量子計算 IBM QUANTUM EXPERIENCE

2

## 2. 執行程式

1

### 1. 功能列

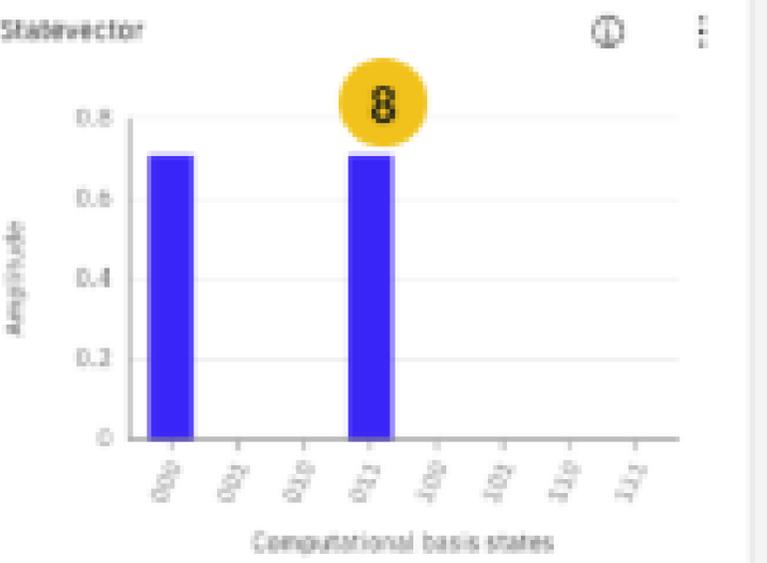
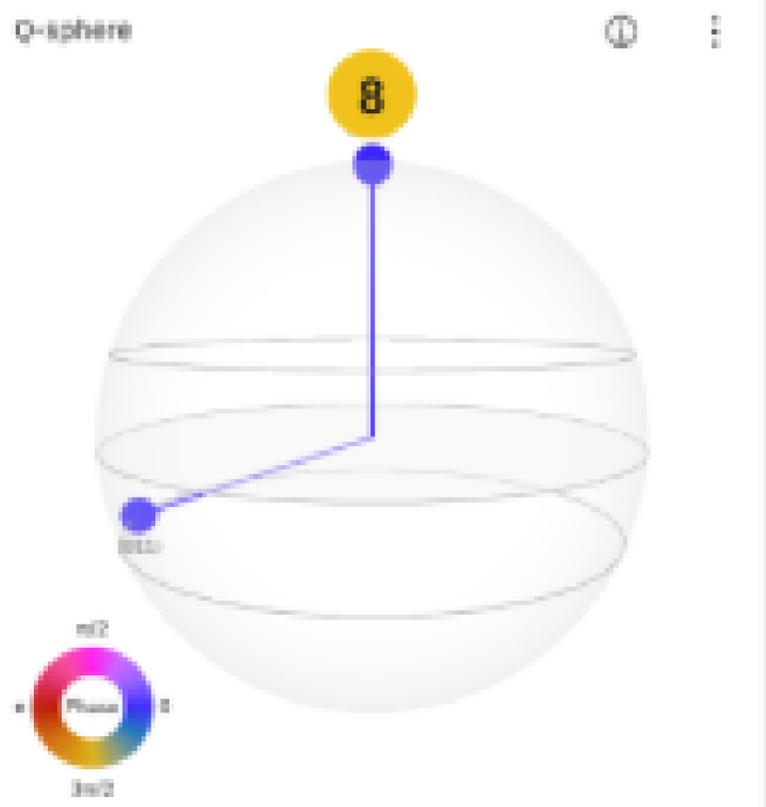
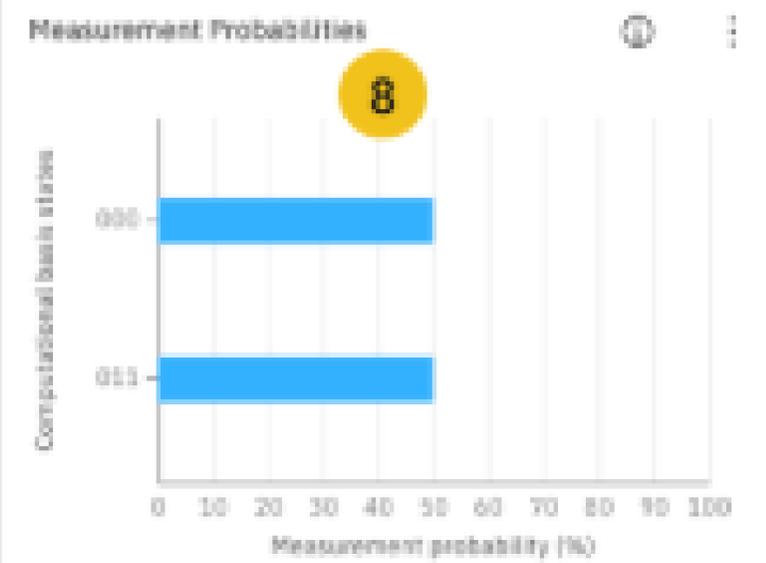
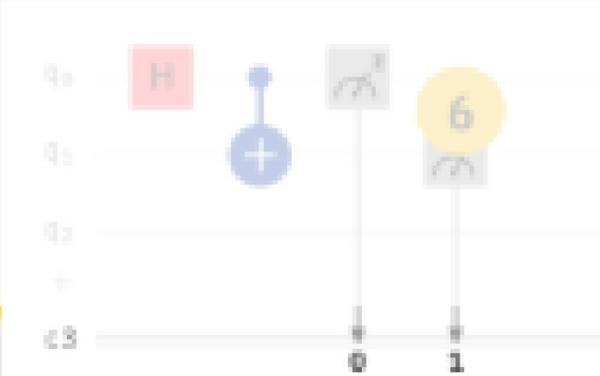
3

4

4. 開啟及編輯 OPENQASM 和 QISKIT 程式碼：當我們編輯量子線路後，系統會自動產生對應的 OPENQASM 和 QISKIT 程式碼，該等程式碼可結合 PYTHON 語言在平台的另一系統 QUANTUM LAB 執行運作。

File Edit Inspect View Help

Circuit 3 titled circuit saved



```

5  creg c[3];
6
7  h q[0];
8  cx q[0],q[1];
9  measure q[0] -> c[0];
10 measure q[1] -> c[1];

```

Output state  
 $[0.707+0j, 0+0j, 0+0j, 0.707+0j, 0+0j, \dots]$



# 量子計算 IBM QUANTUM EXPERIENCE

2

## 2. 執行程式

1

## 1. 功能列

3

4

5

## 5. 量子閘元件：各類量子閘功能請參考 OPERATIONS GLOSSARY。

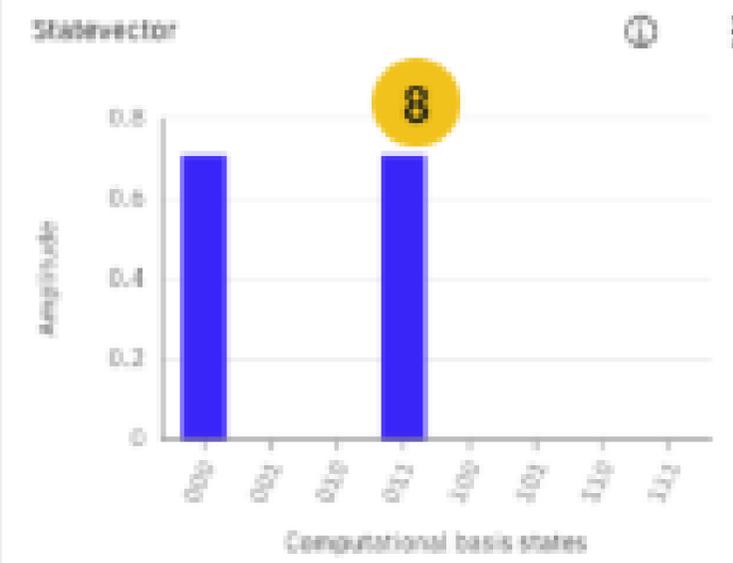
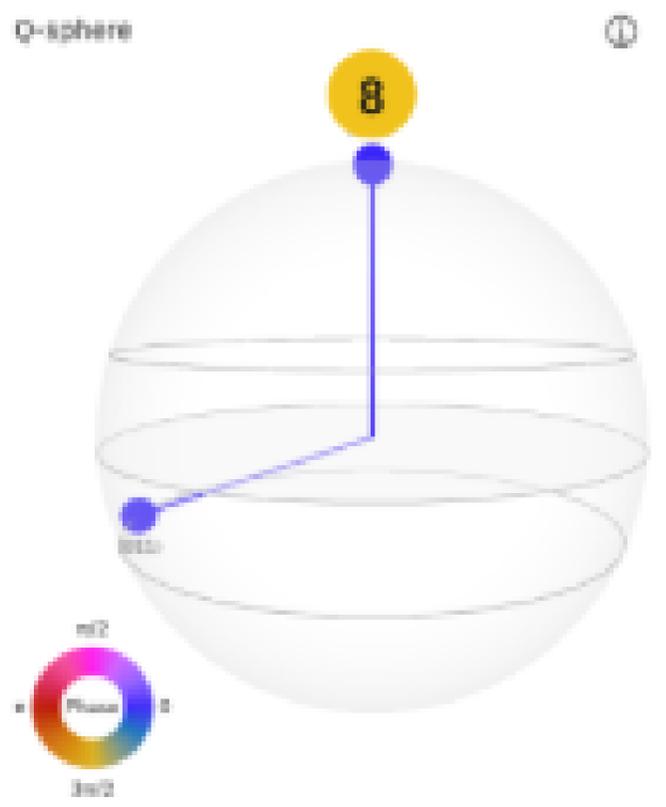
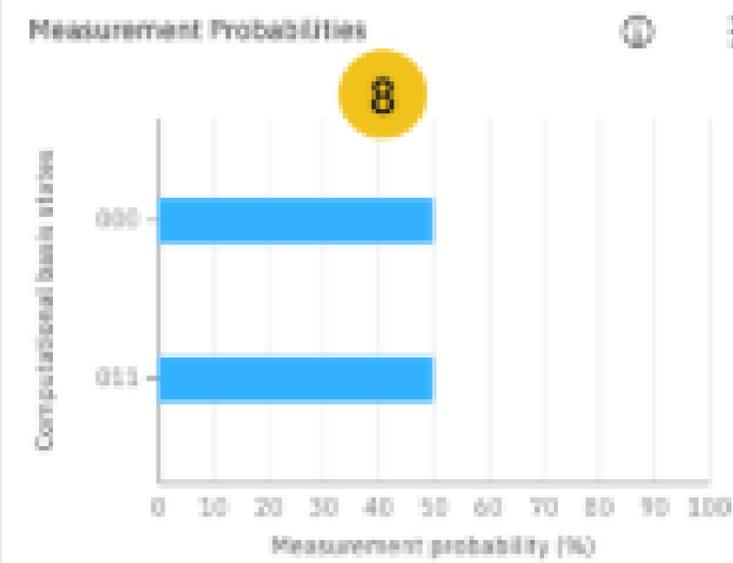
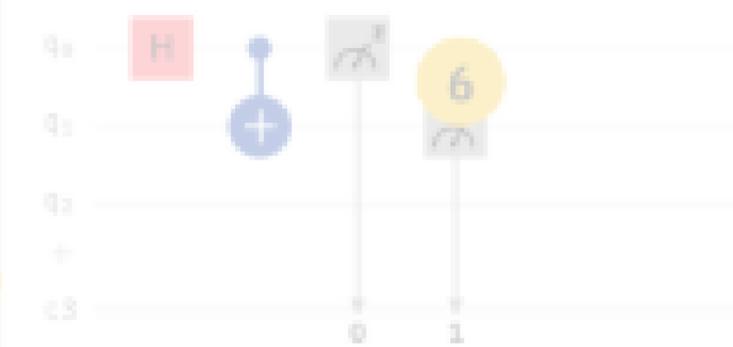
6

7

8

8

8



```
QASM
1 qubit q[0];
2 qubit q[1];
3 include "qelib1.inc";
4 qreg c[3];
5 creg c[3];
6
7 h q[0];
8 cx q[0],q[1];
9 measure q[0] -> c[0];
10 measure q[1] -> c[1];
```

Output state  
[ 0.707+0j, 0+0j, 0+0j,  
0.707+0j, 0+0j] ...



2

## 2. 執行程式

1

## 1. 功能列

3

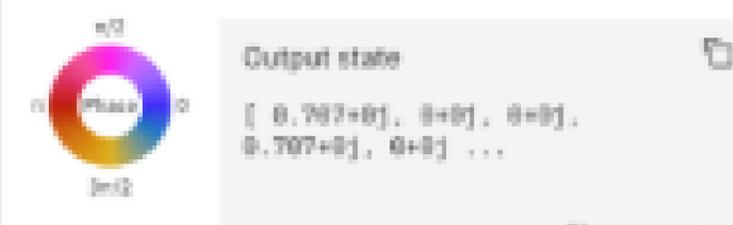
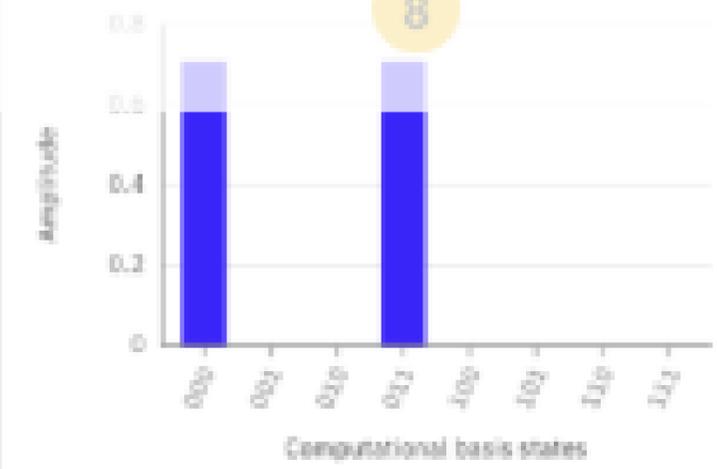
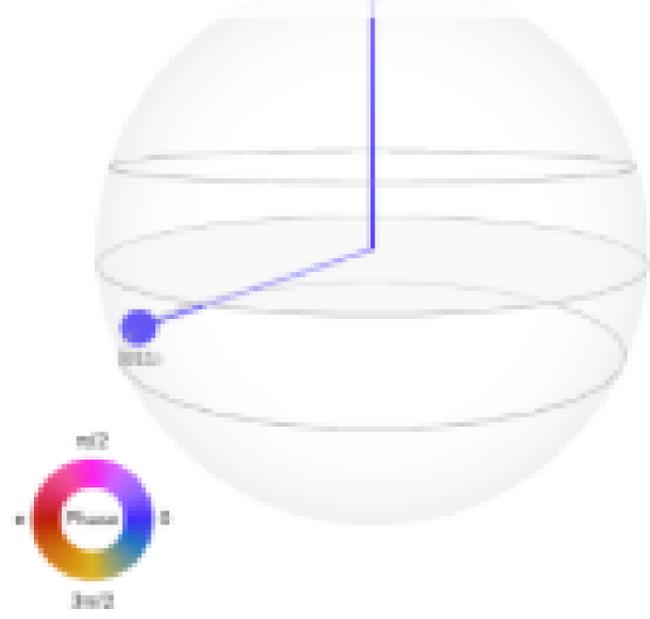
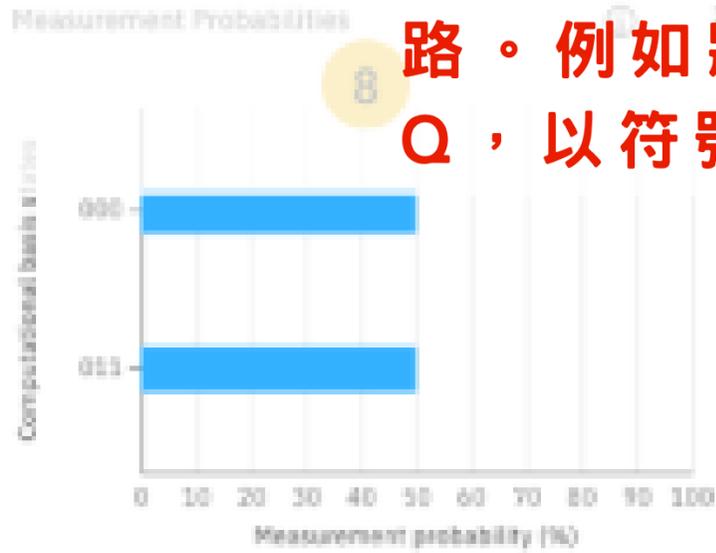
4

5

6. 量子線路 (QUANTUM CIRCUIT) 編輯區：此處即我們編輯量子線路執行量子計算的區域。編輯區左邊 Q0-Q1 即代表量子位元，C1 代表傳統位元用以顯示執行結果。編輯區中間用來置放量子閘，形成量子線路。例如將某量子閘 A 拉到某位元 Q 後面，即代表 A 作用於該量子位元 Q，以符號表示即為  $A|Q\rangle$ 。

8

8





# 量子計算 IBM QUANTUM EXPERIENCE

2

## 2. 執行程式

1

## 1. 功能列

3

4

5

6

7

## 7. 工具列

8

8

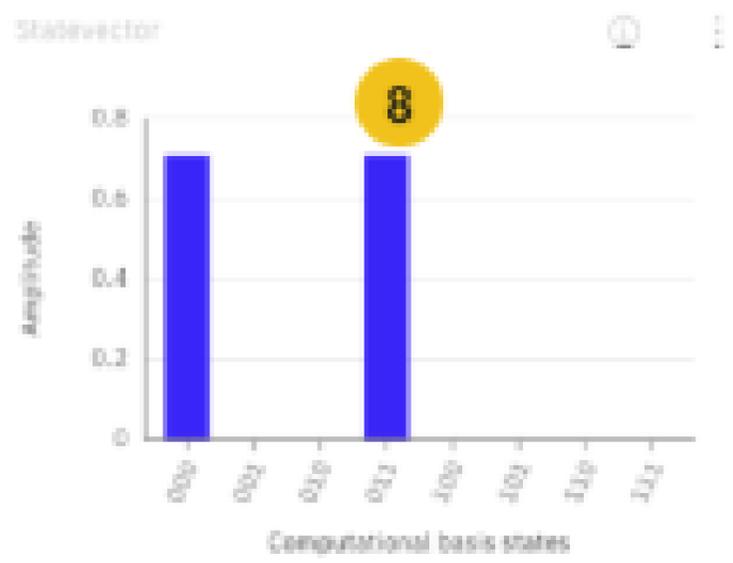
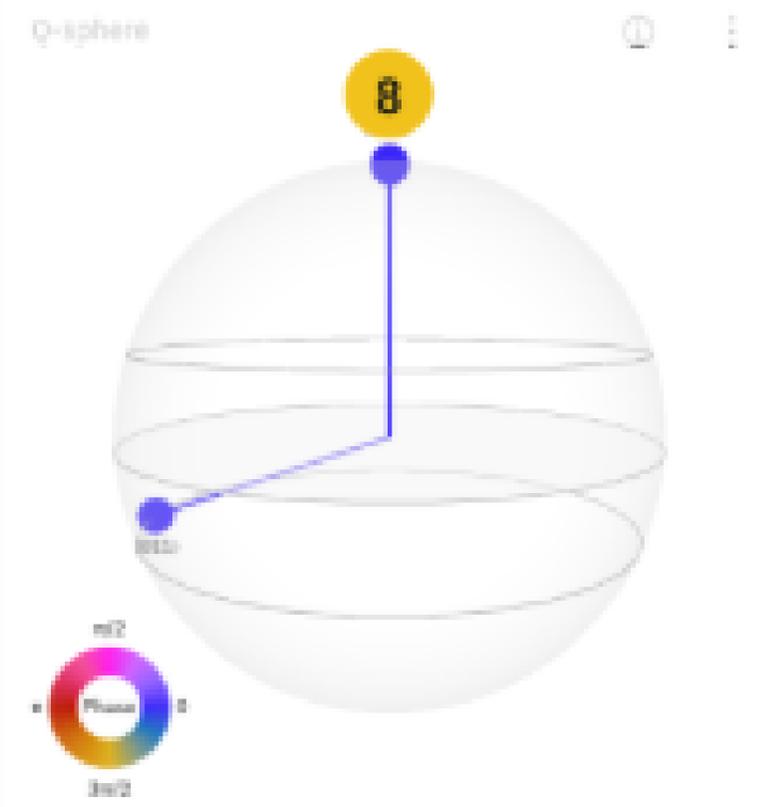
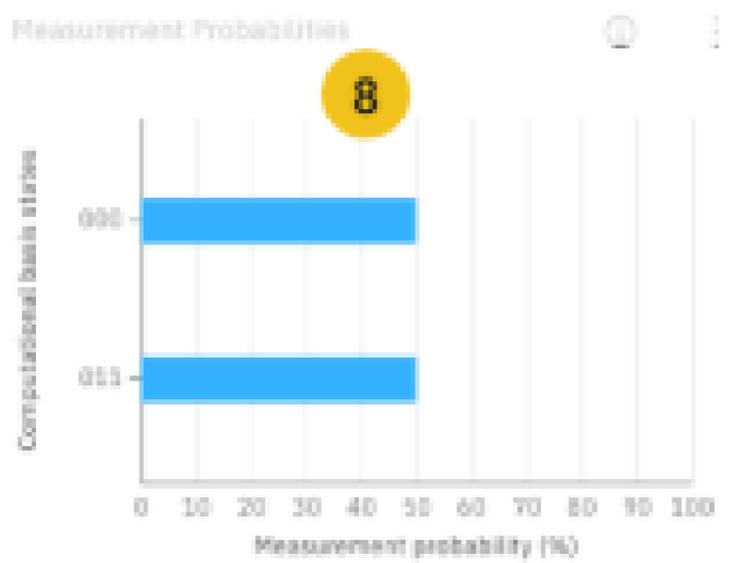
8



```

QASM
1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 qreg q[3];
5 creg c[3];
6
7 h q[0];
8 cx q[0],q[1];
9 measure q[0] -> c[0];
10 measure q[1] -> c[1];

```



Output state  
 $[\ 0.707+0j, \ 0+0j, \ 0+0j, \ 0.707+0j, \ 0+0j \dots]$



# 量子計算 IBM QUANTUM EXPERIENCE



File Edit Inspect View Help

1. 功能列

2. 執行程式

2 Run on ibmq\_qx3as

3 titled circuit saved

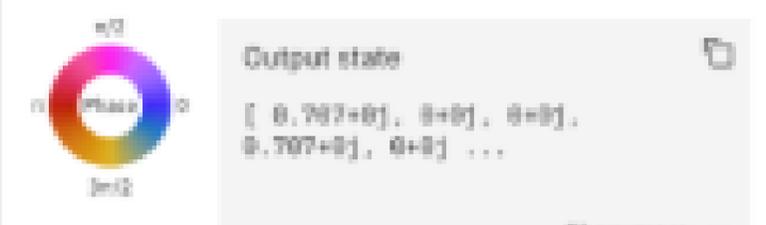
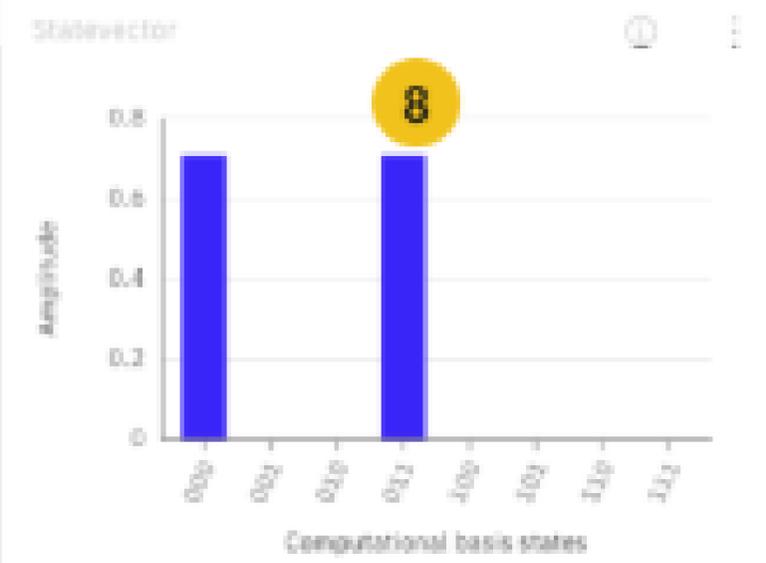
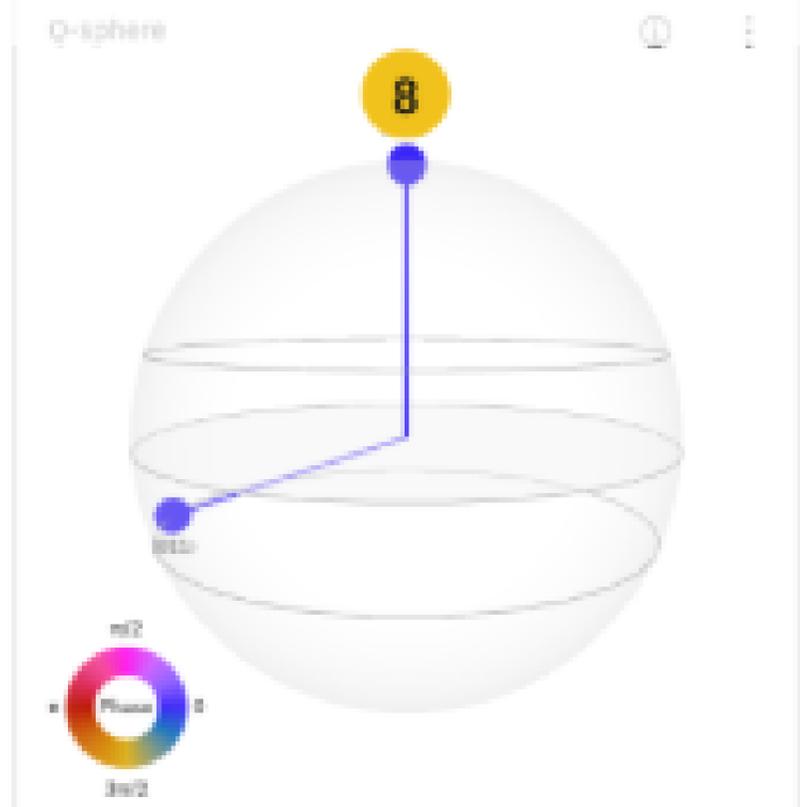
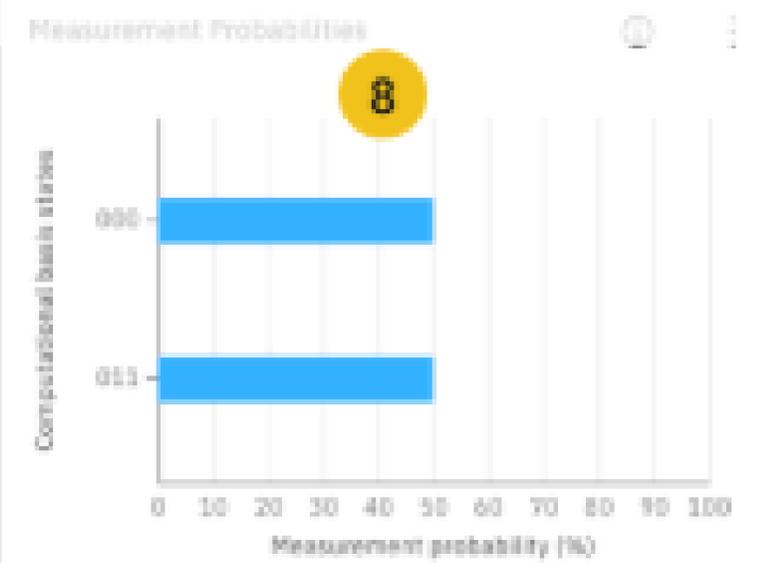
4 </> Code Docs Jobs

5

H ⊕ CNOT CNOT T S Z S† T† U1 |0⟩ |1⟩ RX RY RZ U3 Y U2 CH CY CZ CRX CRY CRZ CU1 CU3 RXX RZZ + Add



8. 視覺化圖區：即時提供量子線路執行成果。



Code editor Open in Quantum Lab

```

QASM
1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 qreg q[3];
5 creg c[3];
6
7 h q[0];
8 cx q[0],q[1];
9 measure q[0] -> c[0];
10 measure q[1] -> c[1];

```





# 量子計算



量子計算是運用量子力學的物理特性，  
例如：疊加態、糾纏態等，進行計算工作，藉以解決複雜的問題。

在此，我們是以量子邏輯閘構成的量子電路為主要的討論範疇。





# 量子計算



## 狄拉克表示法（或稱BRA-KET表示法）

1933年，狄拉克與薛丁格因為「發現原子理論中很有用的形式」，即量子力學的基本方程式：狄拉克方程式與薛丁格方程式，共同獲得諾貝爾物理學獎。

BRA-KET表示法是一種複數希爾伯特空間的向量表示法，適合用來表示量子狀態。BRA-KET其實是英文BRACKET拆字的結果，其中BRACKET是「括號」的意思，數學符號為  $\langle \quad \rangle$ 。





## 狄拉克表示法 (或稱 BRA-KET 表示法)

### 定義：Ket

Ket 可以定義為希爾伯特空間  $H$  的向量，表示成：

$$|a\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

其中  $a_1, a_2, \dots, a_n \in \mathbb{C}$ ， $n$  稱為維度(Dimension)。

Ket 是一個行向量(Column Vector)，也可以表示成  $|a\rangle = (a_1, a_2, \dots, a_n)^T$ ，其中向量的元素  $a_1, a_2, \dots, a_n$  可以是實數，也可以是複數，因此可以用來表示許多不同的量子狀態。換言之，複數希爾伯特空間中， $|a\rangle$  是一個  $n$  維的複數向量，因此也

可以表示成  $|a\rangle \in \mathbb{C}^n$ 。

由於 Ket 表示法中包含一個右括號，因此也經常稱為右向量<sup>2</sup>。此外，Ket 表示法中的  $a$ ，可以是數字、英文字母或數學符號，例如： $|0\rangle$ 、 $|1\rangle$ 、 $|a\rangle$ 、 $|b\rangle$ 、 $|\psi\rangle$ 、

$|\phi\rangle$  等。

### 定義：Bra

Bra 可以定義為希爾伯特空間  $H$  的向量，表示成：

$$\langle a| = (a_1^*, a_2^*, \dots, a_n^*)$$

其中， $a_1, a_2, \dots, a_n \in \mathbb{C}$ ，星號\*代表共軛複數。

Bra 稱為 Ket 的伴隨向量(Co-Vector)。由於 Bra 表示法中包含一個左括號，因此也經常稱為左向量<sup>3</sup>。請注意，Ket 為行向量，Bra 為列向量。

兩者的關係可以表示成： $|a\rangle^\dagger = \langle a|$  或  $\langle a|^\dagger = |a\rangle$

其中，符號 $\dagger$ 稱為共軛轉置(Conjugate Transpose)，或稱為厄米特共軛(Hermitian Conjugate)<sup>4</sup>。因此，Bra 與 Ket 的關係也可以表示如下：

$$|a\rangle^\dagger = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}^\dagger = (a_1^*, a_2^*, \dots, a_n^*) = \langle a| \quad \text{或} \quad \langle a|^\dagger = (a_1^*, a_2^*, \dots, a_n^*)^\dagger = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = |a\rangle$$

以下是典型的 Bra 範例 (分別對應上述的 Ket 範例)：

$(1, 0)$ 、 $(0, 1)$ 、 $\frac{1}{\sqrt{2}}(1, 1)$ 、 $\frac{1}{\sqrt{2}}(1, -i)$ 、 $(1, 0, 0, 0)$ 、 $(0, 1, 0, 0)$  等





## 狄拉克表示法 (或稱 BRA-KET 表示法)

### 定義：Ket

Ket 可以定義為希爾伯特空間  $H$  的向量，表示成：

$$|a\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

其中  $a_1, a_2, \dots, a_n \in \mathbb{C}$ ， $n$  稱為維度(Dimension)。

Ket 是一個行向量(Column Vector)，也可以表示成  $|a\rangle = (a_1, a_2, \dots, a_n)^T$ ，其中向量的元素  $a_1, a_2, \dots, a_n$  可以是實數，也可以是複數，因此可以用來表示許多不同的量子狀態。換言之，複數希爾伯特空間中， $|a\rangle$  是一個  $n$  維的複數向量，因此也可以表示成  $|a\rangle \in \mathbb{C}^n$ 。

由於 Ket 表示法中包含一個右括號，因此也經常稱為右向量<sup>2</sup>。此外，Ket 表示法中的  $a$ ，可以是數字、英文字母或數學符號，例如： $|0\rangle$ 、 $|1\rangle$ 、 $|a\rangle$ 、 $|b\rangle$ 、 $|\psi\rangle$ 、 $|\phi\rangle$  等。

### 定義：Bra

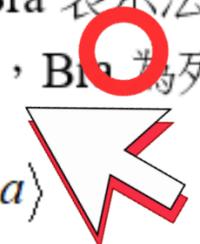
Bra 可以定義為希爾伯特空間  $H$  的向量，表示成：

$$\langle a| = (a_1^*, a_2^*, \dots, a_n^*)$$

其中， $a_1, a_2, \dots, a_n \in \mathbb{C}$ ，星號\*代表共軛複數。

Bra 稱為 Ket 的伴隨向量(Co-Vector)。由於 Bra 表示法中包含一個左括號，因此也經常稱為左向量<sup>3</sup>。請注意，Ket 為行向量，Bra 為列向量。

兩者的關係可以表示成： $|a\rangle^\dagger = \langle a|$  或  $\langle a|^\dagger = |a\rangle$

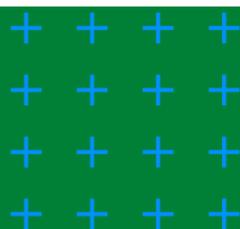


其中，符號  $\dagger$  稱為共軛轉置(Conjugate Transpose)，或稱為厄米特共軛(Hermitian Conjugate)<sup>4</sup>。因此，Bra 與 Ket 的關係也可以表示如下：

$$|a\rangle^\dagger = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}^\dagger = (a_1^*, a_2^*, \dots, a_n^*) = \langle a| \quad \text{或} \quad \langle a|^\dagger = (a_1^*, a_2^*, \dots, a_n^*)^\dagger = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = |a\rangle$$

以下是典型的 Bra 範例 (分別對應上述的 Ket 範例)：

$(1, 0)$ 、 $(0, 1)$ 、 $\frac{1}{\sqrt{2}}(1, 1)$ 、 $\frac{1}{\sqrt{2}}(1, -i)$ 、 $(1, 0, 0, 0)$ 、 $(0, 1, 0, 0)$  等





# 我與量子的距離



謝謝聆聽

