獨立集問題的生物分子解的量子加速與數學解 IBM 量子計算機

Weng-Long Chang [1]*、Ju-Chin Chen [2]*、Wen-Yu Chung [3]*、Chun-Yuan Hsiao [4]*、Renata Wong [5]* 和 Athanasios V. Vasilakos [6]*

抽象的- **在本文中，我們提出了一種生物分子演算法，具有 O(n^2+m) 生物操作、O(2^n) DNA 鏈、O(n) 管和最長 DNA 鏈 O(n)，用於求解獨立的- 任意具有 *m 個邊*和 *n 個*頂點的圖 G 的設定問題。接下來，我們展示了從生物分子解決方案產生的一種新型簡單布林電路，其中 *m 與非*門，($m+n\times(n+1)$)*與*門和 ($(n\times(n+1))/2$) *NOT 閘可以找到*任何具有 *m 個邊*和 *n 個頂點的圖 G* 的獨立集問題的最大獨立集。我們表明，所提出的一種新型量子分子演算法可以透過下界Ω($2^{n/2}$) 查詢和上限O($2^{n/2}$) 查詢找到最大獨立集。這項工作提供了明顯的證據，證明為了解決任何具有 *m 個邊*和 *n 個頂點的圖 G* 中的獨立集問題，生物分子計算機能夠生成一種新型的簡單布爾電路，以便通過實現它量子計算機可以提供二次加速。這項工作還提供了一個明顯的證據，表明量子電腦可以顯著加快生物分子電腦的速度並增強其可擴展性。接下來，輸入 *n* 位的元素不同性問題是確定給定的 $2^{n\times(2/3)}$ *個*實數是否不同。在量子行走演算法的情況下，解決元素獨特性問題的量子下界是Ω($2^{n\times(2/3)}$) *查詢*，我們進一步證明，所提出的量子分子演算法將量子下界減少為Ω(($2^{n/2}$) / ($2^{1/2}$)) 查詢。此外，為了證明所提出的量子分子演算法的可行性，我們透過在具有 5 個量子位元的後端 ibmqx4 和具有 32 個量子位元的後端模擬器上進行實驗，成功解決了具有兩個頂點和一條邊的圖 G 的典型獨立集問題。**

索引術語— **資料結構與演算法、獨立集問題、分子演算法、分子計算、量子演算法、量子計算**

I. 介紹

FEYNMAN [1] 是第一個提出分子計算的人，但他自己卻沒有實現這個想法。幾十年後，透過處理 DNA 鏈，Adleman [2]成功地解決了試管中哈密頓路徑問題的一個例子。1982 年，費曼[3]提出了計算理論中最重要的問題之一，即基於量子理論的計算設備是否能夠比標準圖靈機[4]更快完成計算。貝尼奧夫 [5] 也考慮了量子計算的可能性。Deutsch 設計了量子計算的通用模型—量子圖靈機 [6]。

圖 $G=(V, E)$ 是根據頂點定義的 邊，其中 V *是 n 個頂點*的集合，E *是 m 個邊*的集合。在數學上，圖 $G=(V, E)$ *的獨立集合*是子集 $V^1 \subseteq V$ 的頂點，使得對於 V^1 中的每兩個頂點，沒有邊連接這兩個頂點 [7]。獨立集問題是尋找 G 中*最大*尺寸*的獨立集*。 這個問題是 NP 完全問題 [7]。

我們在本文的主要貢獻如下。

- 我們證明任何具有 *m 個*邊和 *n 個頂點的*圖 *G 的獨立集問題*可以透過以下方式解決 所提出的分子演算法具有 O(n^2+m) 生物操作、O(2^n) DNA 鏈、O(n) 管和最長 DNA 鏈 O(n)。

- 我們證明了從生物分子解決方案中獲得的一種新型簡單布林電路，其中 *m 與非*門，($m+n\times(n+1)$)*與*門和 $\left(\frac{n\times(n+1)}{2}\right)$ *NOT 閘可以找到任何有 m 個*邊和 *n 個頂點的圖 G 的*獨立集合問題的最大獨立集合。

- 我們表明，所提出的量子分子演算法用於實現從生物分子解決方案生成的一種新型簡單布林電路，可以為相同問題提供二次加速。這是最著名*的*加速，因為 解決問題的下限是$\Omega\left(2^{n\times\frac{1}{2}}\right)$查詢，上限是 $O\left(2^{n\times\frac{1}{2}}\right)$查詢。

- 任何具有 *m 個*邊和 *n 個頂點的*圖 *G* 的獨立集問題提供了明顯的證據。生物分子電腦能夠產生一種新型的簡單布林電路，當由量子電腦實現時，它可以提供二次加速，這是已知針對給定問題的最佳加速。

- 我們證明，這項工作提供了另一個明顯的證據，表明量子電腦可以顯著加速和增強生物分子電路的速度和可擴展性。

本作品於 2020 年 10 月 19 日提交。

這項工作得到了中華民國國家科學基金會 MOST 105-2221-E-151-040 -的支持。

W.-L。張就職於國立高雄科技大學電腦科學與資訊工程系，地址：台灣高雄市三民區建工路 415 號，郵編 807-78（電子郵件： changwl@cc.kuas.edu.tw ）。

J.-C。陳教授，國立高雄科技大學電腦科學與資訊工程系，地址：台灣高雄市三民區建工路 415 號，郵編 807-78 （電子郵件： jc.chen@nkust.edu.tw) .

W.-Y。鐘先生就職於國立高雄科技大學電腦科學與資訊工程系，地址：台灣高雄市三民區建工路 415 號，郵編 807-78 （電子郵件：wychung@nkust）。

C.-Y。Hsiao 就職於國立高雄科技大學電腦科學與資訊工程系，地址：台灣高雄市三民區建工路 415 號，郵編 807-78 （電子郵件：cyhsiao@nkust）。

R. Wong 就職於南京大學電腦科學與技術系，地址：仙林路 163 號，江蘇省南京市，郵編：210023 （電子郵件：renata.wong@protonmail.com ）。

A. V. Vasilakos 澳洲雪梨科技大學電機與資料工程學院福州大學數學與電腦科學學院 福州 350116 電腦科學系、電機與太空工程系，呂勒奧科技大學，呂勒奧 97187，瑞典（電子郵件：th.vasilakos@gmail.com ）。

•我們展示了相同生物分子解決方案的數學解決方案如何用有限維希爾伯特的單位向量進行編碼。

•我們也證明，NP 完全問題之間的約簡處理不僅無法加快量子演算法的效能，反而會減慢其效能。

•我們 證明 NP 完全問題的簡化對於量子電腦來說是無用的，因此人們應該獨立開發一種新的量子演算法，以二次加速解決任何 NP 完全問題。

•具有 *n 個*頂點和 *m 個邊的圖 G* 中的獨立集問題的具有二次加速比的量子分子演算法不是*最好*或*最優的*量子演算法。

•輸入 *n* 位的元素不同性問題是決定給定的 $2^{n\text{個}}$實數是否不同。解決此問題的量子下界是對$\Omega\left(2^{n\times\frac{2}{3}}\right)$量子行走演算法的查詢。我們表明，所提出的量子分子演算法減少了查詢的量子下限$\Omega\left(\sqrt{\frac{2^n}{2}}\right)$。

•具有 5 個量子位元的 IBM 後端 *ibmqx4 和*具有 32 個量子位元的後端*模擬器*上實驗性地解決了具有兩個頂點和一條邊的圖中的獨立集問題的實例。

本文的其餘部分組織如下：在第二節中，給出了這項工作的動機。在第三節中，我們闡述了分子電腦和量子電腦的發展。在第四節中，提出了解決任意具有 *m* 條邊和 *n 個頂點的圖 G* 的獨立集合問題的分子演算法。在第五節中，我們提出了一種量子演算法，用於解決任何具有 *m 個*邊和 *n 個*頂點的圖上的獨立集問題。在第六節中，我們分析了所提出的解決相同問題的量子分子演算法的時間複雜度和空間複雜度。在第七節中，我們展示如何將相同問題的分子解的數學解編碼為有限維希爾伯特空間中的單位向量。在第八節中，我們證明了減少 NP 完全問題是*沒有用的*，為每個 NP 完全問題獨立開發更好的量子演算法是解決這個問題的正確方法。在第九節中，我們表明量子下界是Ω( ) 查詢量子遊走演算法，該演算法透過$2^{n\times\frac{2}{3}}$*n 位*輸入解決元素獨特性問題。元素相異性問題是判斷給定的 $2^{n\text{個}}$實數是否相異。我們給了為什麼它被簡化為( )查詢$\sqrt{\frac{2^n}{2}}$的證明（原因）Ω。在第 X 節中，我們在具有 5 個量子位元的 IBM 後端 *ibmqx4* 和具有 32 個量子位元的後端*模擬器*上實驗性地解決了具有兩個頂點和一條邊的圖中的獨立集問題的實例。在第 XI 節中，我們在具有 32 個量子位元的 IBM 後端*模擬器*上實驗性地解決了具有三個頂點和兩條邊的圖中的獨立集問題的實例。在第十二節中 = 12 * ROMAN ，我們給出一個簡短的結論。

II. 動機

貝內特等人。 [8] 證明了解決輸入大小為 *n 位的*任何 NP 完全問題的量子演算法的下界是Ω($2^{n\times\frac{1}{2}}$) 。 此結果表明，一種用於解決任何 NP 完全問題的新型量子演算法可以提供二次加速，如果其上限為O( )，則這是該問題$2^{n\times\frac{1}{2}}$*已知的最佳*加速。一個有趣的開放問題是「解決任何 NP 完全問題的分子解的數學解是什麼」？任何具有 *m 個*邊和 *n 個頂點的*圖上的獨立集問題都是 NP 完全問題 [7, 9]，其分子解、其量子解以及同一分子解的數學解尚未提出。我們寫這篇文章的動機是尋找這三種解決方案。

III. 分子和量子計算機的發展

DNA 演算法的一個潛在重要應用領域是破解加密方案 [10-13]。為了解決許多眾所周知的計算問題，提出的 DNA 演算法包括 3-SAT 問題[14]、三頂點著色[15]、二元整數規劃問題[16]、子集產生式[17]和實數背包問題的 DNA 實驗 [18]。其他著名的 DNA 演算法包括集合劃分問題 [19]、基於規則的系統的分子驗證 [20]、生物分子資料庫的實現 [21] 以及複雜向量算術運算的實現 [22]。伍茲等人。 [23] 報告了 DNA 瓦片集的設計和實驗驗證，該 DNA 瓦片集包含 355 個單鏈瓦片，並且可以透過簡單的瓦片選擇重新編程以實現各種 6 位元演算法。值得注意的是 DNA 計算演算法和模型的最新發展，例如 [24-26]。

第一個量子演算法是 Deutsch-Jozsa 演算法，它展示瞭如何利用量子圖靈機的一些固有的量子力學特徵 [27]。 1994 年，Shor 提出了他的量子演算法，用於有效解決因式分解和離散對數問題 [28]。 1996 年，[29] 中提出了在未排序資料庫中搜尋答案的 Grover 演算法。量子計算和量子資訊的詳細描述在[30-32]中給出。 Aaronson 和 Shi 在[33]中針對碰撞和元素獨特性問題提出了量子遊走演算法的量子下界。Huo 和 Long 在[34]中表明，在線性相互作用狀態下，可以以簡單的方式產生單光子糾纏態，並且在非線性相互作用狀態下，提出了一種利用固體中三波混頻生成微波壓縮態的方案提出了-態電路。楊等人。在[35]中提出了多量子位元 Grover 搜尋的實現。 Long 和 Xiao 在[36]中實現了具有七個量子位元的 NMR 量子資訊處理器。 Boneh 和 Lipton 在 [37] 中證明，量子電腦能夠基於他們所謂的「隱藏線性形式」在量子多項式時間內破解任何密碼系統。 Lukac 和 Perkowski 在[38]中提出了一種量子符號邏輯綜合的演化方法，並使用遺傳演算法來綜合量子電路。莫伊利特等。 [39] 展示了有界度圖的旅行商問題的量子加速。張等人。 [40] 解決了在量子加速圖中尋找最大團的問題。佩洛夫斯克等。在[41]中演示了量子退火器上的大最小頂點覆蓋問題。阿魯特等人。在[42]中確定他們的 Sycamore 處理器需要大約 200 秒來對量子電路的一個實例進行一百萬次採樣——他們的基準目前表明，一台最先進的經典超級計算機的等效任務將需要約 10,000 年。[43-44]中介紹如何編寫量子程式來解決 IBM 量子電腦和量子處理單元上的實際應用。

IV. 解決獨立集問題的分子演算法

*m 個*邊和 *n 個頂點*的圖的獨立集合問題的定義。接下來介紹[2]中提出的 DNA 鍊和生物操作。它們將被應用於設計分子電路來解決獨立集問題。然後，給出了所提出的分

子演算法的時間複雜度和空間複雜度。之後，給出了由生物分子解決方案產生的簡單布林電路，以解決任何具有 *m 個*邊和 *n 個*頂點的圖上的獨立集問題。最後，我們使用資料依賴性分析來證明，解決任何具有 *m 個*邊和 *n 個*頂點的圖上的獨立集問題的簡單布林電路是該問題中最著名的。

A. 獨立集問題的定義

設 G 為圖，$G=(V,E)$，其中 V 是一組頂點，E 是 G 中的一組邊。我們假設 V 是 $\{v_1,\ldots,v_n\}$ 且 E 是 $\{(v_a,v_b)|$ v_a 和 v_b 分別是 $V\}$ 中的元素。我們進一步假設 $|V|$ 表示 V 和 |中的頂點數*電子*|表示 E 中的邊數。我們也假設 $|V|$ 等於 n 且 |*電子*|等於 m 。 m 的值最多為 $((n\times(n-1))/2)$。圖 G 的*獨立集合*是子集 $V^1\subseteq V$ 的頂點使得對所有 $v_a,v_b\in V^1$，邊 (v_a,v_b) 不在 *E [7, 9]* 。 [7, 9]中所引用的**定義 4-1** 用來表示具有 m *條*邊和 *n 個頂點的圖 G 的獨立集合問題*。

定義 4-1 ：具有 *n 個*頂點和 *m 個邊的圖 G* 的獨立集合問題是找出 *G 中最大尺寸的獨立集合*。

$v_3,v_2,v_1\}$和兩邊$\{(v_1,v_2),(v_1,v_3)\}$組成的圖 G^1。G^1中的獨立集合是{}（空集合）、$\{v_1\}$、$\{v_2\}$、$\{v_3\}$和$\{v_3,v_2\}$。 G^1的最大獨立集合是$\{v_3,v_2\}$。從[7, 9]我們知道找到最大尺寸的獨立集合是一個 NP 完全問題。因此，我們可以將其表述為「計算搜尋」問題。

B. 生物分子操作的介紹與實施

DNA（去氧核糖核酸）編碼細胞有機體的遺傳訊息。它由作為 DNA 鏈的聚合物鏈組成。合成 DNA 鍊是透過使用自動化流程來訂購的。每條鏈可以由連接至糖-磷酸「主鏈」的核苷酸或鹼基序列組成。四種 DNA 核苷酸是腺嘌呤、鳥嘌呤、胞嘧啶和胸腺嘧啶，通常分別縮寫為 *A* 、 *G* 、 *C* 和 *T*。根據化學慣例，每條鏈都有一個 5' 末端和一個 3' 末端。因為單鏈的一端有一個遊離的（即未連接到另一個核苷酸上）5'磷酸基團，另一端有一個遊離的 3'脫氧核糖羥基，因此，任何單鏈都具有天然的方向，如[45]中所述。

當兩條單獨的單股結合時，這種結合形成經典的 DNA 雙螺旋。透過鹼基的成對吸引力發生鍵結： *A* 與 *T 鍵合*， *G* 與 *C 鍵結*。因此，(A,T) 和 (G,C) 稱為*互*補鹼基對 [45]。同樣在[45]中，我們發現將溶液加熱到由鏈組成決定的溫度可能會使雙股 DNA 變性為單股。加熱會破壞*互*補鏈之間的氫鍵（[45]中的（圖 4-1））。由於 *G* - *C* 對由三個氫鍵連接，因此斷裂它所需的溫度略高於僅由兩個氫鍵連接的 *A* - *T 對[45]*。這是設計表示計算元素的序列時最重要的因素。

退火是熔化的逆過程，即冷卻單鏈溶液，並允許*互*補鏈結合在一起（[45]中的（圖 4-1））。在雙股 DNA 中，如果其中一條單股包含不連續性（即，一個核苷酸未與其相鄰核苷酸結合），則可以透過 DNA 連接酶進行修復 [45]。這使我們能夠將多個鏈透過各自的*互*補鏈結合在一起創建一個統一的鏈。

我們將使用[2, 45-46]中引用的以下生物分子操作來為任何具有 *m 個*邊和 *n 個*頂點的圖構建獨立集問題的分子解決方案。下面描述了[45]的**定義 4-2** 至**定義 4-9** 中指定的八種生物操作的實施。每個實作僅說明完成一種生物操作的計算行為的一種可能方式。在實驗室技術中，未來的改進很可能會產生更有效和抗錯誤的生物操作實施，但這並不紅。

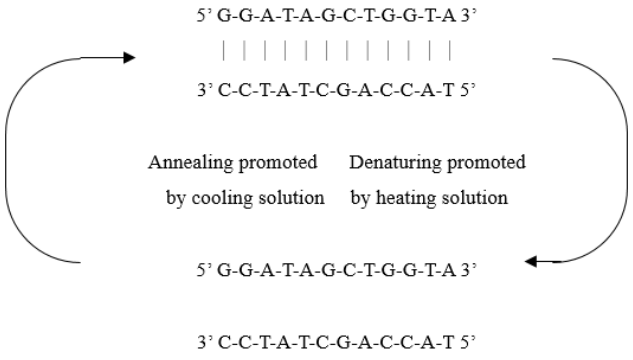


圖 4-1 。 DNA 變性和退火。

提高模型的理論能力。我們只是提供實施的描述，以證明在體外執行生物操作原則上的可行性（也就是說，使用現有的實驗室技術，每個生物操作都是完全可行的）。從生物學角度來看，代表比特的所有序列都必須經過檢查，以確保它們編碼的 DNA 鏈不會彼此形成不需要的二級結構（即，鏈始終保持分離，只有在需要時才結合在一起）。我們已經詳細解決了基於 DNA 的計算的鏈設計問題，並且我們使用[45]中描述的方法來最大限度地減少不必要的結合的可能性。

定義 4-2 ：給定集合 $X=\{x_n x_{n-1}\ldots x_2 x_1|\ \forall xd\in\{0,1\}$ 為 $1\le d\le n\}$ 和一個位 x_j，生物分子操作「Append-Head」將 x_j *附加到集合* X 中每個元素的頭部。形式化表示為 Append-Head$(X,x_j)=\{x_j xn x_{n-1}\ldots x_2 x_1|\ \forall xd\in\{0,1\}$ 為 $1\le d\le n$ 和 $x_j\in\{0,1\}\}$。

定義 4-3 ：給定集合 $X=\{x_n x_{n-1}\ldots x_2 x_1|\ \forall xd\in\{0,1\}$ 為 $1\le d\le n\}$ 和一個位元 x_j，生物分子操作"Append-Tail"，將 x_j *附加到集合* X 中每個元素的末尾。正式表示形式為 Append-Tail$(X,x_j)=\{x_n x_{n-1}\ldots x_2 x_1 x_j|\ \forall xd\in\{0,1\}$ 為 $1\le d\le n$ 和 $x_j\in\{0,1\}\}$。

兩條鏈（圖 4-2 中標示為 *S* 和 *T* ）可依下列方式連接：建立一條*接頭鏈，其中包含 S* 的補體序列，後接 *T 的補體序列*。此連接鏈透過磁珠固定在表面上（圖 4-2(a)）。然後，將鏈 *S* 添加到溶液中，並在適當的位置與連接鏈退火（圖 4-2（b））。然後，將 *T 鏈*加入溶液中，並在緊鄰 *S 鏈的位置與連接鏈退火*（圖 4-2(c)）。然後，我們將連接酶添加到溶液中，以密封 *S* 和 *T 之間的「切口」* ，形成一條單鏈，可以透過加熱溶液以破壞其與連接鏈的鍵來釋放該單鏈（圖 4-2 (d)）。上面提到的 *concatenate () 操作*的實作可以很容易地用於將特定序列 *s 附加到管 X* 中每條鏈的頭部。在這種情況下，序列 *s* 對應圖 4-2 中的 *S 股，以及*圖 4-2 中的 *T 股 對應於管 X* 中每條鏈的起始序列。另外，只是開始每條鏈的序列與連接鏈退火。顯然，在一條鏈上完成**定義 4-2 中定義的**一系列 *append - head* ()操作後，其序列將由多個代表比特串的序列組成。可以應用類似的實作來完成**定義 4-3 中定義的** *append - tail* () 操作。

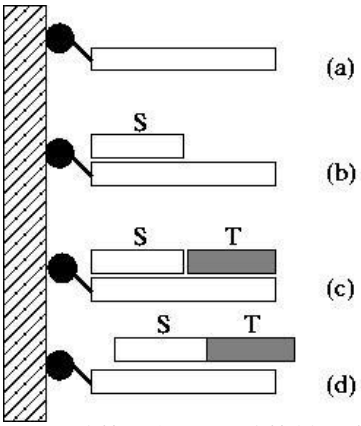


圖 4-2。連接過程：(a) 連接鏈固定到表面。 (b) S 退火至連接鏈。 (c) T 退火至與 S 相鄰的連接鏈。

定義 4-4 ：給定集合 $X=\{x_n x_{n-1}\ldots x_2 x_1 \mid \forall xd \in \{0,1\}$ 為 $1\le d\le n\}$，生物分子操作「Discard(X)」將 X *重設為*空集，可以表示為「 $X=\varnothing$ 」。

定義 4-4 中定義的*丟棄*(X)操作丟棄管 X *的內容物，並*使用新的空管取代管 X。由於管的數量通常為一個，因此這是恆定時間操作。

定義 4-5 ：給定集合 $X=\{x_n x_{n-1}\ldots x_2 x_1 \mid \forall xd \in \{0,1\}$ 為 $1\le d\le n\}$，生物分子操作「Amplify(X, $\{X_i\}$ *建立*集合 X *的*多個相同副本 X_i，然後藉助「Discard(X)」丟棄 X。

定義 4-5 中定義的 Amplify (X, $\{X_i\}$) 操作是透過應用聚合酶鍊式反應 (*PCR*) 來實現的，其初始輸入是管 X。此反應用於大量擴增（可能是少量）以特定引子開頭和結尾的 DNA 序列。因為使用這些序列界定了 X 管中的每條鏈，所以它們都會在反應中被複製。然後，將 PCR 結果均分到指定的管中（因此，無論管的數量如何，可以調整 PCR 循環的數量以確保每管的 DNA 體積恆定）。

定義 4-6 ：給定集合 $X=\{x_n x_{n-1}\ldots x_2 x_1 \mid \forall xd \in \{0,1\}$ 為 $1\le d\le n\}$ 和位 x_j，生物分子*提取*操作有兩種表示法。第一個表示法是 $+(X, x_j{}^1)=\{x_n x_{n-1}\ldots x_j{}^1\ldots x_2 x_1 \mid \forall xd \in \{0,1\}$ 為 $1\le d\ne j\le n\}$ 且 $-(X, x_j{}^1)=\{x_n x_{n-1}\ldots x_j{}^0\ldots x_2 x_1 \mid \forall xd \in \{0,1\}$ 為 $1\le d\ne j\le n\}$ *如果* x_j 的值等於 1。第二種表示法是 $+(X, x_j{}^0)=\{x_n x_{n-1}\ldots x_j{}^0\ldots x_2 x_1 \mid \forall xd \in \{0,1\}$ 為 $1\le d\ne j\le n\}$ 且 $-(X, x_j{}^0)=\{x_n x_{n-1}\ldots x_j{}^1\ldots x_2 x_1 \mid \forall xd \in \{0,1\}$ 為 $1\le d\ne j\le n\}$ *如果* x_j 的值等於 0。

為了實現**定義 4-6** 中定義的*提取操作*，使用親和純化從由短鏈 *s 組成的管 X 中提取任何鏈*，該短鏈 s 編碼位值 x_j。該過程使用與搜尋的目標序列互補的探針序列。探針可以附著在表面上，並透過對由目標序列組成的任何鏈進行退火來捕獲鏈。然後，在其餘群體中，透過將捕獲的鏈放入單獨的溶液中來分離它們，然後加熱溶液以破壞探針和目標序列之間的鍵。因此，使用的探針是 *s 的互補序列*。保留的鏈被放置在一個新管中， $U=+(X, s)$，其餘的被放置在另一個新管中， $V=-(X, s)$。

定義 4-7 ：給定 m 組 $X_1\ldots X_m$，生物分子*合併*操作為 $\cup(X_1,\ldots,X_m)=X_1\cup\ldots\cup X_m$。

定義 4-7 中定義的*合併*操作是透過將管（組）$\{X_i\}$的內容倒入指定管中來實現的。管數通常較少，因此是*恆定時間*操作。

定義 4-8 ：給定集合 $X=\{x_n x_{n-1}\ldots x_2 x_1 \mid \forall xd \in \{0,1\}$ 為 $1\le d\le n\}$，生物分子操作 如果 X 不是空管，“Detect(X)” 回傳 *true* 。否則，它會傳回 *false* 。

定義 4-8 中定義的*檢測*操作是透過使管 X *運行*凝膠電泳過程來實現的，該過程通常用於按長度對 DNA 鏈進行分類。X 中存在的任何 DNA 都會在凝膠中表現為可見條帶；如果存在適當長度的 DNA 鏈，則該操作傳回 *true*。如果沒有與正確長度的 DNA 對應的可見條帶，則該操作傳回 *false* 。長度標準確保目前的 DNA 片段不會導致假陽性結果。如果後續處理步驟需要條帶中對應於 X 含量的 DNA，則切割條帶可能會將其從凝膠中切除。然後將帶子浸泡以除去線股以供進一步使用。

定義 4-9 ：給定集合 $X=\{x_n x_{n-1}\ldots x_2 x_1 \mid \forall xd \in \{0,1\}$ 為 $1\le d\le n\}$，生物分子操作 「Read(X)」*描述* X 中的任何元素。即使 X 包含許多不同的元素，該操作也可以給出其中一個元素的明確描述。

定義 4-9 中定義的*讀取*操作是透過使用*凝膠電泳*對管 X *中的 DNA 鏈*按大小進行排序來實現的。電泳是帶電分子在電場中的運動。由於 DNA 分子帶有負電荷，因此當置於電場中時，它們往往會向正極遷移。分子在*水溶液*中的遷移速率取決於其形狀和電荷。由於 DNA 分子每單位長度具有相同的電荷，因此它們在水溶液中都以相同的速度遷移。然而，如果一條 DNA 鏈在*凝膠（通常由瓊脂糖、聚丙烯酰胺或兩者的組合製成）中完成*電泳，那麼它的大小也會影響分子的遷移速率。因此，凝膠是分子必須穿過的緻密孔隙網絡。因此，較小的分子在凝膠中遷移得更快，從而根據大小對它們進行分類。*鹼基*長度適中的 DNA 鏈 *對*進行測量。

C. *解決獨立集問題的分子演算法*

從**定義 4-1** 可知，對於任何具有 n *個*頂點和 m *條邊的圖* G ，所有可能的獨立集合是 G 中包含合法和非法獨立集的 2n 個可能*選擇*。每個可能的選擇對應於 G *中的頂點子集*。因此，假設 Y *是* 2^n *個*可能選擇的集合，且 Y 等於 $\{y_n y_{n-1}\ldots y_2 y_1 \mid \forall$ *年* $\in\{0,1\}$ 為 $1\le d\le n\}$。這樣， Y 中每個元素的長度為 n *位，每個元素代表* 2^n *個可能的選擇*之一。為了方便表述，我們假設 $y_d{}^0$ *表示* y_d 的值為 0， $y_d{}^1$ *表示* y_d 的值為 1。如果一個元素 $y_n y_{n-1}\ldots y_2 Y$ 中的 y_1 是合法的獨立集合且 y_d *的值為* $1\le d\le n$ 為 1，則 $y_d{}^1$ 表示第 d 個頂點在合法獨立集中。如果一個元素 $y_n y_{n-1}\ldots y_2 Y$ 中的 y_1 是合法的獨立集合且 y_d *的值為* $1\le d\le n$ 為零，則 $y_d{}^0$ 表示第 d 個頂點沒有出現在合法獨立集中。

我們提出以下分子演算法來解決任何具有 n *個*頂點和 m *個邊的圖 G 的獨立集問題*。第一個參數是 n 個空管（一組） Y_0 視為輸入管（組）；第二個參數 n 表示頂點數，第三個參數 m 表示邊數。**程式解獨立集問題**（ Y_0, n, m ）中的每個管都是空管，被視為輔助記憶體。

程式 求解獨立集合問題(Y_0, n, m)

(0a) 追加尾部($X_1, y_n{}^1$)。

(0b) 追加尾部($X_2, y_n{}^0$)。

(0c) $Y_0 = \cup(X_1, X_2)$。

(1)**對於** $d = n-1$ **減至** 1

(1a)放大(Y_0, X_1, X_2)。

(1b) 追加尾部($X_1, y_d{}^1$)。

(1c)追加尾部($X_2, y_d{}^0$)。

(1d) $Y_0 = \cup(X_1, X_2)$。

結束於

(2)**對於**每條邊， $ek = (vi, v_j)$，在 G *中*，其中 $1 \le k \le m$ 和位元 y_i 和 y_j 分別表示頂點 *vi* 和 v j 。

(2a) $P^1 = +(Y_0, y_i{}^1)$ 且 $P^3 = -(Y_0, y_i{}^1)$。

(2b) $P^2 = +(P^1, y_j{}^1)$ 且 $P^4 = -(P^1, y_j{}^1)$。

(2c) $Y_0 = \cup(P^3, P^4)$。

(2d) 丟棄(P^2)。

結束於

(3)**對於** *我* = 0 **到** n -1

(4)**對於** j =*我* **降至** 0

(4a) $Y_{j+1}{}^{ON} = +(Y_j, y_{i+1}{}^1)$ 且 $Y_j = -(Y_j, y_{i+1}{}^1)$。

(4b) $Y_{j+1} = \cup(Y_{j+1}, Y_{j+1}{}^{ON})$。

結束於

結束於

(5)**對於** $c = n$ **降至** 1

(5a)**如果**（檢測（ Y_c ））**則**

(5b)讀取(*Yc*)並終止演算法。

結束如果

結束

結束程式

引理 4-1 ：具有 m *個*邊和 n *個頂點的圖* G 的獨立集合問題可以透過分子演算法求解**獨立集問題**(Y_0, n, m)。

證明：

每次執行步驟(0a)和步驟(0b)時，分別為 y_n *作為集合* X_1 中每個元素的第一位，值為“ 0 ” y_n 作為集合 X_2 中每個元素的第一位。這表示 $X_1 = \{y_n{}^1\}$ 且 $X_2 = \{y_n{}^0\}$。接下來，每次執行步驟(0c) 都會為兩個集合 X_1 和 X_2 建立集合並集，使得 $Y_0 = X_1 \cup X_2 = \{y_n{}^1, y_n{}^0\}$，且 $X_1 = \varnothing$ 且 $X_2 = \varnothing$。

集合 Y_0 的兩個相同副本 X_1 和 X_2，並且 $Y_0 = \varnothing$。每次執行步驟 (1b) 後都會附加價值“1” 對於 X_1 中的每個元素， y_d *到* $y_n \dots y_{d+1}$ 的末尾。類似地，每次執行步驟(1c)也會附加價值“0” 對於 X_2 中的每個元素， y_d *到* $y_n \dots y_{d+1}$ 的末尾。接下來，每次執行步驟 (1d) 都會為兩個集合 X_1 和 X_2 建立集合並集，使得 $Y_0 = X_1 \cup X_2$，且 $X_1 = \varnothing$ 且 $X_2 = \varnothing$。反覆執行後 步驟(1a)至(1d)， $Y_0 = \{y_n y_{n-1} \dots y_2 y_1 | \forall$ *牟* $\in \{0, 1\}$ 為 $1 \le d \le n\}$。這就是說，管 Y_0 *中的* 2^n *個* DNA 鏈編碼 2^n 個可能的選擇（獨立集）。

G 中第 k *個邊*的形式評估每個公式 $\left(\overline{y_i \wedge y_j}\right)$，其中 $1 \le k \le$ *米*。每次執行步驟(2a)時，管 P^1 由那些具有 $y_i = 1$ 的 DNA 鏈組成，管 P^3 包含那些具有 $y_i = 0$ 的 DNA 鏈，管 Y_0 變成空管。接下來，在每次執行步驟（2b）時，管 P^2 包含那些具有 $y_i = 1$ 和 $y_j = 1$ 的 DNA 鏈，管 P^4 包含那些具有 $y_i = 1$ 和 $y_j = 0$ 的 DNA 鏈，且管 P^1 變成空管。這顯示 P^2 *管中的分子解在第* k *條*邊上包含兩個頂點，是*非法*獨立集； P^4 管中的分子解在第 k *條*邊上只包含一個頂點，是*合法的*獨立集合； P^3 管中的分子解在第 k *條*邊上有一個頂點或沒有頂點，是*合法的*獨立集合。然後，在每次執行步驟(2c)時，管 Y_0 包含編碼合法獨立集的那些 DNA 鏈，管 P^3 是空管，管 P^4 也是空管。接下來，在每次執行步驟(2d)時，由管 P^2 中的 DNA 鏈編碼的*非法獨立組*被丟棄。重複執行步驟（2a）至（2d）後，管 Y_0 *由那些滿足* 1 的≤G 中第 k *條邊*的真實值的 DNA 鏈組成 $\wedge_{k=1}^{m}\left(\overline{y_i \wedge y_j}\right)$ $k \le$ *米*。

接下來，步驟(3)和(4)依序為唯一嵌套循環的外循環和內循環，*第一個*循環索引變數 i *的範圍*為 0 到 $n-1$ ，*而第二個*循環索引變數 j *的*範圍是從 i 到 *0* 。（組） Y_{j+1} 和 Y_j 中的個數。每次執行步驟(4a)時，*提取操作*從管（組） Y_j 形成兩個不同的管（組） $Y_{j+1}{}^{ON}$ 和 Y_j。這就是說，tube (set) $Y_{j+1}{}^{ON}$ 具有 $y_{i+1} = 1$， tube (set) Y_j 具有 $y_{i+1} = 0$ 。套循環， y_{i+1} 對 1 數量的影響是在管（組） $Y_{j+1}{}^{ON}$ *中記錄單一* 1 ，並且在管（組） Y_j *中記錄零個* 1。接下來，在兩級嵌套循環中的迭代(i ， j ）*處每次執行步驟（4b）時，應用合併*操作將 tube（set） $Y_{j+1}{}^{ON}$ 的內容倒入 tube（set）中 Y_{j+1}。這表明，在二級嵌套循環中的迭代(i, j)*時*， y_{i+1} 對個數的影響是在管(集合) Y_{j+1} 中記錄單個個。接下來，從迭代 ($i, j-1$ ）透過迭代（ $n-1$, 0) 在兩級嵌套循環中，應用類似的處理來計算 y_{i+1} 至 y_n 對 1 個數的影響。因此，每次操作完成後， *Y 管中的那些 DNA 鏈*為≤0 *我* ≤ n *有* i 個包含 i *個頂點*。接下來，步驟(5)是一個循環，讀取最大獨立集合的分子解。每次執行步驟(5a)時，如果管 Y_c 中有 DNA 鏈，則傳回「真」。接下來，在每次執行步驟（5b）時，讀取最大尺寸獨立集的答案並且演算法終止。因此，推斷具有 m 條邊和 n *個頂點的圖* G *的獨立集問題*可以透過分子演算法 **Solve-independent-set-problem** (Y_0, n, m)來求解。■

D. 求解獨立集問題的分子演算法的時空複雜度

以下引理用於描述分子演算法**求解獨立集問題**($Y_0, n,$ *米*）。

引理 4-2 .具有 n *個*頂點和 m *個邊的*任何圖 G 的獨立集問題可以透過 O($n^2 + m$) = O(n^2) 生物操作、O(2^n) DNA 鏈、O(n) 管和最長的解決方案來解決 DNA 鏈，O(n)。

證明：

在 分子演算法，**求解獨立集合問題**(Y_0, n, m)，步驟 (0a)、(0b) 和 (0c) 需要兩次「Append-Tail」操作和一次「Merge」操作。接下來，在執行步驟(1a)至步驟(1d)時，進行($n-1$)

「放大」操作、(2 ×(n −1))「追加尾部」操作和(n −1)「合併」操作。接下來，因為 m 的值（邊數）最多為 ((n ×(n −1)) / 2)，執行步驟(2a)到步驟(2d)時，最多(n ×(n −1))「提取」操作，((n ×(n −1)) / 2) “合併”操作和 ((n ×(n −1)) / 2) 需要「丟棄」操作。接下來，在執行步驟(4a)至步驟(4b)時，需要((n ×(n + 1)) / 2) “提取”操作和 ((n ×(n + 1)) / 2) “合併”操作。最後，在執行步驟(5a)至步驟(5b)時，最多需要(n) *次「偵測」操作和一次「讀取」操作。*

引理 4-1 的證明，我們建構了編碼 2^{n} *個可能的獨立集合的* 2^{n} *個 DNA 鏈*，（2× 使用 n +7) *個管*。由於每個可能的獨立集合的長度為 n *位，且每個位都可以由長度恆定*的短 DNA 鏈編碼，因此最長的 DNA 鏈為 O(n)。因此，從上面的陳述可以立即推斷出任何具有 n *個*頂點和 m *個邊的圖 G 的獨立集問題可以透過* $O(n^2+m) = O(n^2)$ 生物操作來解決， $O(2^n)$ DNA 鏈，O(n) 管和最長的 DNA 鏈，O(n)。■

E. 簡單的布林電路 用於從生物分子解決方案中確定獨立集

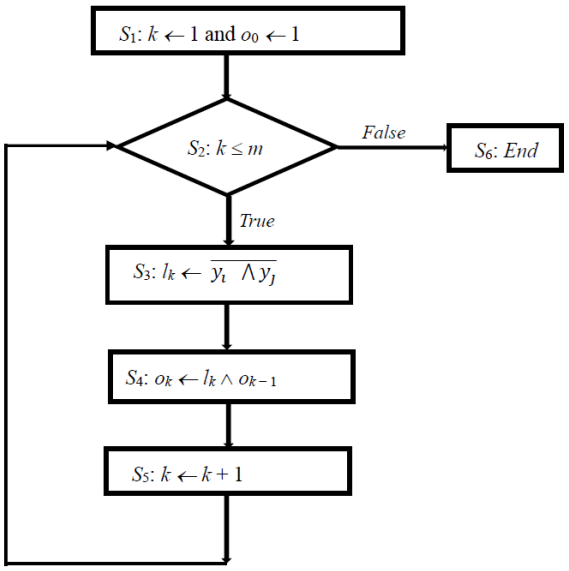
分子演算法**求解獨立集問題**(Y_0, n, m) 中步驟 (0a) 到 (1d) 的每個生物操作完成後，管 Y_0 *中的* 2^{n} *個*DNA 鏈編碼 2^{n} *個*可能的選擇。接下來，在從步驟（2a）到步驟（2d）的每個生物操作之後，以相同的迭代 k 為 $1 \le k \le m$ *完成後，管* P^2 中的生物分子解在第 k *條邊上包含*兩個頂點，為*非法獨立集；管* Y_0 中的生物分子解在*第* k *條邊上*包含 1 個頂點或 0 個頂點，為*合法*獨立集。因此，表 4.1 中出現的用於實現 NAND 運算的真值表可以表示在同一迭代 k 為 1 時從步驟 (2a) 到步驟 (2d) 產生≤**的簡單布林電路** $k \le$ *米*。

輸入		輸出
是我	*yj*	$\overline{y_i \wedge y_j} = l_k$
0	0	1
0	1	1
1	0	1
1	1	0

表 4-1： **NAND 運算**的真值表。

位 y_i 和 y_j 是其兩個輸入，位 l_k 為 $1 \le k \le m$ 是其輸出。如果 l_k *位的值為* $1 \le k \le m$ 等於 1，則對應的頂點子集在第 k *條邊* (v_i, v_j) *中只包含* 1 *個頂點或 0 個頂點*，是合法的獨立集合。否則，對應的頂點子集包含第 k 條邊 (v_i, v_j) *中的兩個頂點*，並且是非法獨立集合。因此，從迭代一到迭代 m *重複執行步驟(2a)至(2d)後，管* Y_0 中的生物分子解在每條邊上包含一個頂點或零個頂點，並且不包含任何一條邊的兩個頂點。這就是說，管 Y_0 *中的生物分子解對*那些頂點子集進行編碼，其中對於所有頂點 v_i 和 v_j ，邊 (v_i, v_j) *不在* E 中，E 是圖中的邊集*格*。這也意味著管 Y_0 中的生物分子解滿足兩個輸入 y_i 和 y_j *的每個與*非運算都具有真值的事實。因此，在所有 m *次迭代*中從步驟 (2a) 到步驟 (2d) 產生的簡單布林電路就是實作布林公式 $(\wedge_{k=1}^{m} (\overline{y_i \wedge y_j}))$ 並找出哪些頂點子集滿足 $\wedge_{k=1}^{m} (\overline{y_i \wedge y_j}$ 具有真值的布林公式 ( ))。

圖 4-3 顯示了針對具有 n *個*頂點和 m *個邊的圖* G 的獨立集問題識別獨立集的流程圖。在圖 4-3 中，在語句 $S_{1\text{中}}$，第一個迴圈的索引變數 k 設定為一 (1)。接下來，在語句 $S_{2\text{中}}$，執行第一個迴圈的條件判斷。如果 k 的值小於或等於 m *的值*，則 *下一執行的*指令是語句 S_3。否則，在語句 $S_{6\text{中}}$，執行 *End* 指令，結束獨立集辨識任務。



具有 n *個*頂點和 m *個邊的圖* G 的獨立集合問題的獨立集合。

在語句 $S_{3\text{中}}$，一個***與非***門“ $l_k \leftarrow \overline{y_i \wedge y_j}$ ”已實施。位(布林變數) y_i 和 y_j 分別編碼由 n *個*頂點和 m *條邊的*圖 G *中的第 k 條邊*連接的頂點 v_i 和頂點 v_j。位元（布林變數） l_k 為 $1 \le k \le m$ *儲存執行 ()*的結果（$\overline{y_i \wedge y_j}$第 k *個****與非***閘）。接下來，在語句 $S_{4\text{中}}$，邏輯與運算「 $o_k \leftarrow lk \wedge ok_{1}$ ”被執行，-即( ) 中的 $\wedge_{k=1}^{m}(\overline{y_i \wedge y_j})$ *第 k 個子句*。位元（布林變數） l_k *儲存第* k 個與***非閘***的執行結果，是邏輯與運算的第一個運算元。位元（布林變數） o_{k-1} 與 $1 \le k \le m$ 是邏輯與運算的第二個運算元，儲存前一個邏輯與運算的結果。位元（布林變數）*可以為*≤l $k \le m$ *儲存執行* l_k 的結果 $\wedge o_{k-1}$（第 k 個子句，即第 k 個***與***門）。接下來，在語句 $S_{5\text{中}}$，第一次迴圈的索引變數 k 的值遞增。重複執行語句 S_2 至 S_5，直至語句 $S_{2\text{的}}$條件判斷結果為*假*值。從圖 4-3 可以看出，與***非門***的總數為 m 。邏輯與運算總數為 m ***與***門。因此，識別獨立集合的成本對應於 m 與***非***門和 m ***與***門。

資料依賴性由兩個存取或修改相同資源的語句所引起。*資料依賴分析是為了確定重新排序*或*並行化語句*是否安全。如果兩個語句中的第一個先修改相同的資源，然後兩個語句中的第二個讀取相同的資源，則第一個語句和第二個語句之間存在***真正的依賴關係。***如果兩個語句中的第一個先讀取相同的資源，然後兩個語句中的第二個修改相同的資源，則***第***一個語句和第二個語句之間存在***反***依賴。如果兩個語句中的第一個先修改相同的資源，然後兩個語句中的第二個修改相同的資源，則第一個語句和第二個語句之間存在***輸出依賴。我們使用資料依賴性分析來表明，圖 4-3 中***

用於識別具有 *n 個*頂點和 *m 個邊的圖 G* 的獨立集問題的獨立集的簡單布林電路是該問題*已知的最佳*布林電路。

引理 4-3 .對於任意具有 *n 個*頂點和 *m 個邊的圖 G 的獨立集合問題，在圖 4-3 中，具有 m* 的布林電路 與***非***門和 *m* 在分子演算法中，從步驟 (2a) 到步驟 (2d) 在所有 *m 次迭代中產生的* ***AND 門，解決獨立集問題***(Y_0, n, m) 是已知用於識別獨立集的最佳布爾電路（ s) $2^{n\text{個}}$可能的選擇。

證明：

在圖 4-3 中，語句 $S_{3\text{中}}$，有一個***與非***門「 $l_k \leftarrow \overline{y_i \wedge y_j}$ 」被執行，並且執行()的結果$\overline{y_i \wedge y_j}$被寫入布林變數 l_k。接下來，在語句 $S_{4\text{中}}$，邏輯與運算「 $o_k \leftarrow lk \wedge$ 執行“o_{k-1}”，需要讀取布林變數 l_k（第一個操作數）的值。因此，語句 S_3 和語句 $S_{4\text{之間}}$存在著***真正的依賴關係***。***真正的***依賴 語句 S_3 和語句 $S_{4\text{之間}}$不能被破壞。因此，圖 4-3 中的每條語句都必須以順序模式執行。因此，從上面的陳述可以立即推斷出，在圖 4-3 中， *m 的簡單布林電路* 與***非***門和 *m* 在分子演算法中，從步驟 (2a) 到步驟 (2d) 在所有 *m 次迭代中產生的* ***AND 門，解決獨立集問題***(Y_0, n, m) 是已知用於識別獨立集的最佳布爾電路（ s) $2^{n\text{個}}$可能的選擇。■

F. 簡單的布林電路 用於根據生物分子解計算獨立集中的頂點數

在分子演算法中步驟（2a）到（2d）的每個生物操作完成後，**解決獨立集問題**（ Y_0, n, m ）*完成，管* Y_0 中的 DNA 鏈編碼每個解決方案（獨立集)的 o_m 值等於一 (1)。為了計算頂點數，我們需要輔助布林變數 $w_{i+1,j}$ 和 $w_{i+1,j+1}$ with $0 \leq i \leq n-1$ 和 $0 \leq j \leq i$。輔助布林變數 $w_{i+1,j}$ 和 $w_{i+1,j+1}$ 與 $0 \leq i \leq n-1$ 和 $0 \leq j \leq i$ 設定為初始值 0（零）。布林變數 $w_{i+1,j+1}$ 與 $0 \leq i \leq n-1$ 和 $0 \leq j \leq$ 在計算出對第 (i + 1) 個頂點進行編碼的布林變數 $y_{i+1\text{對}1}$s（頂點）數量的影響後，i 存儲解中的頂點數。如果布林變數 $w_{i+1,j+1\text{的}}$值為 $0 \leq i \leq n-1$ 和 $0 \leq j \leq i$ 等於 1（一），則這表示解中有 (j + 1) *個（頂點）*。布林變數 $w_{i+1,j}$ 為 $0 \leq i \leq n-1$ 和 $0 \leq j \leq i$ *在計算出*布林變數的影響後，儲存解中的頂點數 y_{i+1} 對第(i + 1) 個頂點進行編碼 s（頂點）的數量。如果布林變數 w_{i+1}的值，$_j$ 對於 $0 \leq i \leq n-1$ 和 $0 \leq j \leq i$ 等於 1（一），那麼這表示解中有 j 個（頂點）。

在一個解中（ n 個獨立集）當位 o_m 的值等於 1 時，位 y_1 對第一個頂點 $v_{1\text{進行編碼}}$。如果位 y_1 的值等於一 (1)，則第一個頂點 v_1 出現在解中，並且它增加解的頂點數量（1 的數量）。否則，第一個頂點 v_1 不會出現在解中，而且它保留解的頂點數（個數）。在分子演算法中，**求解獨立集問題**(Y_0, n, m)，在迭代中 (i = 0, j = 0) 執行步驟 (4a) 時，使用提取操作形成兩個不同的管， $Y_1{}^{ON}$ 和 Y_0 在管（組） $Y_{0\text{之外}}$。因此，管 $Y_1{}^{ON}$ 中的 DNA 鏈編碼具有 $y_1 = 1$ 並且包含頂點 $v_{1\text{的溶液}}$，管 Y_0 中的 DNA 鏈具有 $y_1 = 0$ 且不包含頂點 v_1。也就是說， $y_{1\text{的影響（頂點}}$ v_1的影響）對 1 的數量（頂點的數量）在管 $Y_1{}^{ON\text{中記錄為單一1，並且在管}}$ Y_0 中記錄零個 1 。接下來，在同一迭代中執行步驟(4b)時(i = 0， j = 0)，透過將管 $Y_1{}^{ON}$ 的內容物倒入管 $Y_{1\text{中來應用}}$*合併*操作。這顯示在迭代（ i = 0， j = 0 *）中，* y_1 對 1 的數量的影響在管（組） $Y_{1\text{中被記錄為}}$單一 1 。因此，對於第一個頂點 v_1 的影響，增加每個解的頂點數就是滿足公式 ($o_m \wedge y_1$) 並保留頂點數即滿足公式 ($o_m \wedge \overline{y_1}$ ）。

$1 \leq i \leq n-1$ *的第 (i + 1) 個頂點*的影響是決定每個解中的頂點數（個數）是增加還是保留。為了增加每個解中的頂點數量（個數），必須滿足兩個條件。*第一個*條件是第 (i + 1) 個頂點在解內，*第二個*條件是每個解目前有 j *個*頂點。為了保留每個解中的頂點數量（個數），必須滿足兩個條件。*第一個*條件是第 (i + 1) 個頂點不在解內，*第二個*條件是每個解目前也有 j *個*頂點。接下來，在迭代(i ， j)中每*次*執行步驟(4a)時，使用*提取操作來形成兩個不同的管(組)*，$Y_{j+1}{}^{ON}$ 和從管(組) $Y_{j\text{出來的}}$ Y_j 。因此，管 $Y_{j+1}{}^{ON}$ 中的 DNA 鏈編碼具有 $y_{i+1} = 1$ 且包含頂點 $vi_{+1\text{的每個解}}$。另一方面，管 Y_j 中的 DNA 鏈編碼具有 $y_{i+1} = 0$ 且不包含頂點 $v_{i+1\text{的每個解}}$。這顯示在迭代 (i , j) *中，* y_{i+1} 對個數（頂點數）的影響在管 $Y_{j+1}{}^{ON\text{中記錄為}}$ (j + 1) *個*，也記錄為 j 管 $Y_{j\text{中的那些}}$。接下來，在迭代(i ， j)中每次執行步驟(4b)時，透過將管(組) $Y_{j+1}{}^{ON}$ *的*內容倒入管(組) $Y_{j+1\text{中來應用}}$合併操作。這顯示在迭代 (i , j) *中，* y_{i+1} 對 1 的數量（頂點數）的影響被記錄為管 $Y_{j+1\text{中有}}$(j + 1) 個 1 。因此，對於每個解中 $1 \leq i \leq n-1$ *的第* (i + 1) 個頂點的影響，增加每個解中的頂點數（個數）的兩個條件是滿足布林公式($y_{i+1} \wedge w_{i,j}$)。保留每個解中的頂點數的兩個條件是滿足布林公式（ $(\overline{y_{i+1}}) \wedge w_{i,j}$)。

圖 4-4 給出了統計每個解的頂點數的邏輯流程圖。在圖 4-4 中，語句 $S_{1\text{中}}$，邏輯與運算「 $w_{1,1} \leftarrow$ *奧姆* $\wedge y_1$ ” 的實作對應於一個 ***AND*** 門。布林變數 $w_{1,1\text{儲存實現一個}}$***與閘***的結果 ($o_m \wedge y_1$)。若 $w_{1,1}$ 的值等於 1（一），則頂點數增加，使得具有第一頂點 $v_{1\text{的每個解中的頂點數}}$為一。接下來，在語句 $S_{2\text{中}}$，邏輯與運算「 $w_{1,0} \leftarrow$ *奧姆* $\wedge \overline{y_1}$ ” 的實現對應於一個***與***門。布林變數 $w_{1,0\text{儲存實現一個}}$***與閘***的結果($o_m \wedge \overline{y_1}$ ）。若 $w_{1,0}$ 的值等於 1（一），則保留頂點數，使得*沒有*第一個頂點 $v_{1\text{的每個解中的頂點數}}$為零。

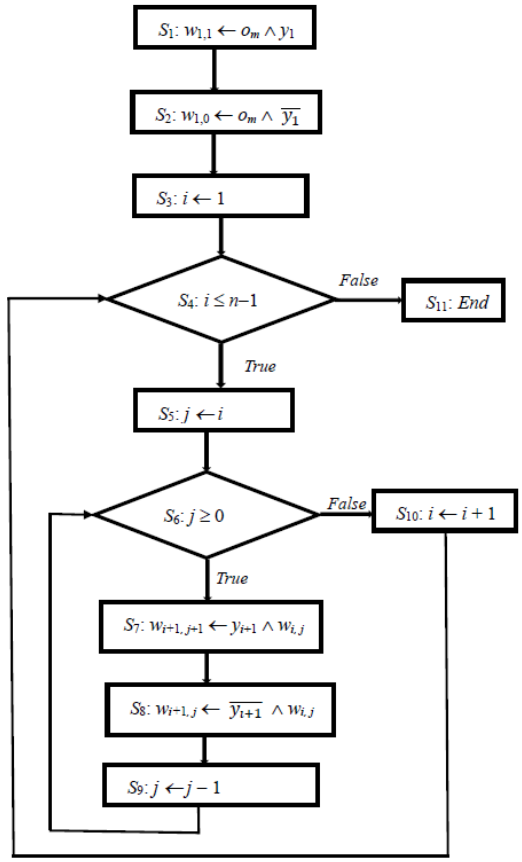


圖 4-4：計算每個解（獨立集）中頂點數量的流程圖。

接下來，在語句 $S_{3中}$，將第一個迴圈的索引變數 i 設為 1。然後，在語句 $S_{4中}$，執行第一個迴圈的條件判斷。如果 i 的值小於或等於 $(n-1)$，則 *下一條執行的*指令是語句 S_5。否則，在語句 $S_{11中}$，執行 *End* 指令，結束統計各解的頂點數的任務。在語句 $S_{5中}$，第二個迴圈的索引變數 j *被設定為*第一個迴圈中的索引變數 i 的值。接下來，在語句 $S_{6中}$，執行*第二次迴圈*的條件判斷。如果 j 的值大於或等於 0，則下一執行的指令是語句 S_7。否則，下一條執行的指令是語句 S_{10}。

在語句 $S_{7中}$，邏輯與運算「$w_{i+1,j+1} \leftarrow$ *我*$_{+1} \wedge w_{i,j}$」的實現對應於一個***與***閘。布林變數 y_{i+1} *對第*$(i+1)$ 個頂點進行編碼，並且是邏輯與運算的第一個運算元。布林變數 $w_{i,j}$ 是邏輯與運算的第二個運算元。布林變數 $w_{i,j}$ *確定*布林變數的影響後，儲存解中的頂點數 *是*$_{我}$ 對 1 s（頂點）的數量上的第 i 個頂點進行編碼。若 $w_{i,j}$ 的值等於 1（一），則表示解中有 j 個（頂點）。布林變數 $w_{i+1,j+1}$ *儲存執行邏輯與運算*「$w_{i+1,j+1}$」的結果 $\leftarrow$ *我*$_{+1} \wedge w_{i,j}$」。也就是說 $w_{i+1,j+1}$ *在決定了*布林變數的影響後儲存了解中的頂點數 y_{i+1} 對第$(i+1)$ 個頂點進行編碼 s（頂點）的數量。如果 $w_{i+1,j+1}$ 的值等於 1（一），則這表示解中有 $(j+1)$個（頂點）。

接下來，在語句 $S_{8中}$，邏輯與運算「$w_{i+1,j} \leftarrow \overline{y_{i+1}} \wedge w_{i,j}$」的實現對應於一個***與***閘。布林變數 y_{i+1} *對第*$(i+1)$ 個頂點進行編碼，其否定$\overline{y_{i+1}}$是邏輯與運算的第一個運算元。布林變數 $w_{i,j}$ 是邏輯與運算的第二個運算元。它在確定布林變數的影響後儲存解中的頂點數 *是*$_{我}$ 對 1 s（頂點）的數量上的第 i 個頂點進行編碼。若 $w_{i,j}$ 的值等於 1（一），則表示解中有 j 個（頂點）。布林變數 $w_{i+1,j}$ *儲存執行邏輯與運算*「$w_{i+1,j}$ 的結果 $\leftarrow \overline{y_{i+1}} \wedge w_{i,j}$」。這顯示在確定了布林變數的影響後，$w_{i+1,j}$ 儲存了一個解中的頂點數 y_{i+1} 對第$(i+1)$ 個頂點進行編碼 s（頂點）的數量。$w_{i+1,j}$ 的值等於 1（一）表示解中有 j 個（頂點）。

接下來，在語句 $S_{9中}$，第二次迴圈中索引變數 j 的值被遞減。重複執行語句 S_6 到語句 S_9，直到語句 $S_{6的}$條件判斷得到*假*值。接下來，在語句 $S_{10中}$，第一個迴圈中的索引變數 i 的值遞增。重複執行語句 S_4 到 S_{10}，直到 $S_{4中的}$條件判斷達到*假*值。當發生這種情況時，下一個執行的語句是 S_{11}。 S_{11}、執行 *End* 指令，結束統計各解的頂點數的任務。圖 4-4 中每個操作的成本為 $(n \times (n+1))$ ***AND*** 閘和 $(\frac{n\times(n+1)}{2})$ ***NOT*** 閘。因此，計算每個解決方案的頂點數的成本是實現 $(n \times (n+1))$ ***AND*** 閘和 $(\frac{n\times(n+1)}{2})$ ***NOT*** 閘。我們使用資料依賴性分析來表明，圖 4-4 中用於計算每個解中頂點數量的簡單布林電路是*已知的解決*該問題的*最佳*布林電路。

引理 4-4 .在圖 4-4 中，布林電路 $(n \times$在分子演算法的每次迭代中，從步驟 (4a) 到 (4b) 產生的$(n+1))$ ***AND*** 閘和 $(\frac{n\times(n+1)}{2})$ ***NOT 門，求解獨立集問題***(Y_0, n, m) 是*最好的*布林電路以計算每個解中的頂點數而*聞名*。

證明：

如圖 4-4 所示，在語句 $S_{7中}$，*有*一個***與***閘「$w_{i+1,j+1} \leftarrow$ *我*$_{+1} \wedge w_{i,j}$」被執行，執行結果 $(y_{i+1} \wedge w_{i,j})$寫入布林變數 $w_{i+1,j+1}$ 為 $1 \leq$ *我* $\leq (n-1)$ 和 *我* $\geq j$ 0. 在≥語句 S *中*的迭代 $(i = 1, j = 1)$中 $_{7}$ 布林變數 $w_{2,2}$ 的值被修改，隨後，在語句 $S_{7的迭代（}i = 2, j = 2$）中，讀取 $w_{2,2}$ 的值。在以後的迭代中，也有類似的在語句 $S_{7中修改和讀取相同資源的情況}$。因此，語句 S_7 存在***真正的****依賴性*。接下來，在語句 $S_{8中}$，邏輯與運算「$w_{i+1,j} \leftarrow \overline{y_{i+1}} \wedge w_{i,j}$」被執行，執行結果$(\overline{y_{i+1}} \wedge w_{i,j})$ 被寫入布林變數 $w_{i+1,j}$ 與 $1 \leq$ *我* $\leq (n-1)$ 和 *我* $\geq j$ 0. 在≥語句 $S_{8的}$迭代 $(i = 1, j = 1)$中，布林變數 $w_{2,1}$ 的值被修改，隨後，在語句 S *8 的迭代* $(i = 2, j = 1)$中，修改布林變數 w 2 , 1 的值 $w_{2,1}$ 被讀取。在以後的迭代中也有類似的修改和讀取語句 $S_{8中相同資源的情況}$。因此，語句 S_8 存在著*真正****的依賴關係****。****真正的****依賴* 在這兩個語句中 S_7 和 S_8 都不能被破壞。

下一個， 在語句 $S_{7中}$的迭代 $(i = 1, j = 1)$ 中，布林變數 $w_{2,2}$ 的值被修改，隨後，在語句 $S_{8中的迭代（}i = 2, j = 2)$ 中，修改 w_2 的值,$_2$ 被讀取。在後面的迭代中，語句 S_7 和 S_8 之間也存在類似的修改和讀取相同資源的情況。因此， S_7 和 S_8 之間存在***真正的依賴性***。接下來，在語句 $S_{8中}$的迭代$(i = 1, j = 1)$中，布林變數 $w_{2,1}$ 的值被修改，隨後，在語句 $S_{7中的迭代（}i = 2, j = 1)$中，布林變數 w 2 , 1 的值被修改 $w_{2,1}$ 被讀取。在後面的迭代中，語句 S_8 和 S_7 之間也存在類似的修改和讀取相同資源的情況。因此， S_8 和 S_7 之間存在***真***

正的依賴性。接下來，在語句 $S_{8\text{中}}$的迭代 ($i = 1, j = 1$) 中寫入布林變數 $w_{2,1}$的值，接著在語句 $S_{7\text{中的迭代}}$ ($i = 1, j = 0$) 中寫入 w_2的值，讀取 $_1$。在後面的迭代中，在語句S_8和 S_7之間修改相同資源也存在類似的情況。因此，語句 S_8和 $S_{7\text{之間}}$存在***輸出依賴性***。這顯示語句 S_7和 $S_{8\text{之間}}$同時存在兩個***真實相關性***和一個***輸出相關性***。語句 S_7和語句 S 8 之間的兩個*真實*$_{\text{依賴}}$關係***和***輸出***依賴關係***無法被打破。因此，圖 4-4 的各語句只能使用順序模式。從上面的陳述可以立即得出，在圖 4-4 中，帶有 (n ×在分子演算法的每次迭代中，在步驟 (4a) 到步驟 (4b) 中產生的(n +1)) ***AND*** 閘和 ($\frac{n\times(n+1)}{2}$) ***NOT 門，求解獨立集問題***(Y_0 , n, m) 是以計算每個解決方案中的頂點數而聞名的最佳布林電路。■

V. 用於從解決獨立集問題的分子解決方案實現簡單布林電路的量子演算法

在本節中，我們介紹量子位元和量子閘。然後，我們使用它們設計一種新的量子演算法來實現由分子解決方案產生的簡單布林電路，以解決任何具有 m *個*邊和 n *個頂點*的圖上的獨立集問題。

A. *量子位元和量子閘簡介*

在二維希爾伯特空間中[30-32，43-44]，*一個量子位元有兩個計算基向量 |0> 和 |1>*，對應於經典位值 0 和 1 。量子暫存器可以由任何 2^n 維計算基礎向量、 n *個*大小的量子位元或這些向量的任意疊加組成[30-32，43-44]。如果量子暫存器的量子位元的內容已知，則可以透過張量積以以下方式計算量子暫存器的狀態： $|\partial\rangle = (|q_n\rangle \otimes |q_{n-1}\rangle \otimes \dots \otimes |q_2\rangle \otimes |q_1\rangle)$ 。如果大小為 n 的量子暫存器的狀態是 $2^{n\text{維計算基底向量}}$的任意疊加，那麼它可以表示為 $|\gamma\rangle = (\sum_{a=0}^{2^n-1} b_a |a\rangle)$，其中每個加權因子 $b_a \in \mathbf{C}$ 是所謂的機率幅；因此它們必須滿足 $(\sum_{a=0}^{2^n-1} |b_a|^2) = 1$。

酉算符通常被稱為量子閘[30-32，43-44]。使用量子閘，我們可以對量子暫存器狀態的時間演化進行建模。因此，量子閘是一種基本量子計算設備，它在固定週期內對選定的量子位元完成固定的酉運算。如[30-32]所給出的，[圖 43-44]，**哈達馬**門 H 是一個量子位元 (2 ××2 矩陣) 的量子閘。其四個條目分別為 $H_{1,1} = 1 / (2^{1/2})$ 、 $H_{1,2} = 1 / (2^{1/2})$ 、 $H_{2,1} = 1 / (2^{1/2})$ 和 $H_{2,2} = -1 / (2^{1/2})$。具有一個量子位元的**非**閘僅將 (目標) 位元設為其否定。當且僅當第一個量子位元 (控制**量子位元）等於 1 時，具有兩個量子位元的 CNOT** (*受控非*) 閘才會翻轉第二個量子位元 (目標量子位元)。當且僅當第一和第二量子位元 (兩個控制量子位元) 均為 1 時，具有三個量子位元的受控非 (CCNOT) 閘才會翻轉第三量子*位元 (***目標***量子***位元***）*。量子閘 $H^{\otimes n}$ 代表 n *個*量子位元的連接哈達瑪閘，應用於具有 $|000\dots0\rangle_n$ *個*量子位元的初始狀態向量，其結果為 $|\lambda\rangle = (\frac{1}{\sqrt{2^n}} \sum_{a=0}^{2^n-1} |a\rangle)$ 。

B. *獨立集問題分子解的計算狀態空間*

基於分子演算法，**求解獨立集問題**(Y_0 , n, m)，在步驟 (0a)到(1d)中產生 $2^{n\text{個}}$可能的選擇(獨立集)，並儲存在集合 (管)中 Y_0等於$\{ y_n y_{n-1} \dots y_2 y_1 | \forall$ *年* $\in \{0, 1\}$ 為 $1 \le d \le n \}$。我們使用以下引理來描述分子解的計算狀態空間，用於求解具有 n *個*頂點和 m *個邊的圖 G 的獨立集合問題*。

引理 5 -1 ：為了解決具有 m *個*邊和 n *個頂點*的圖上的獨立集問題，產生的 $2^{n\text{個}}$可能選擇 (獨立集) 的相應計算狀態向量的集合 分子演算法**求解獨立集問題**(Y_0 , n, m)中的步驟 (0a) 到 (1d)形成 2^n 維希爾伯特空間 (複向量空間) $\mathbb{C}^{2^n}$ 的標準正交基底。

證明：

具有 $2^{n\text{二進制數元組的}}$唯一*計算基向量*來表示集合 (管) Y_0中的每個元素。*第一個*元素 $y_n{}^0$的第一個對應計算基底向量 $y_{n-1}{}^0 \dots y_2{}^0 y_1{}^0$為($[1 \quad 0 \quad \dots \quad 0]^T_{1\times 2^n}$)。*第二個*元素 $y_n{}^0$的第二個計算基底向量 $y_{n-1}{}^0 \dots y_2{}^0 y_1{}^1$為($[0 \quad 1 \quad \dots \quad 0]^T_{1\times 2^n}$)。依此類推，*最後*一個元素的*最後一個*對應的計算基底向量 $y_n{}^1 y_{n-1}{}^1 \dots y_2{}^1 y_1{}^1$為($[0 \quad 0 \quad \dots \quad 1]^T_{1\times 2^n}$)。因此，集合 (管) Y_0中每個元素 (每個可能的獨立集合) 對應的計算基向量的集合為 $D = \{ [1 \quad 0 \quad \dots \quad 0]^T_{1\times 2^n}, [0 \quad 1 \quad \dots \quad 0]^T_{1\times 2^n}, \dots, [0 \quad 0 \quad \dots \quad 1]^T_{1\times 2^n} \}$。$D$ 中的每個計算基向量都是協調向量 [27] ，而這些向量一起跨越 $D = \mathbb{C}^{2^n}$。因此，立即推斷 $2^{n\text{個}}$可能選擇 (獨立集) 產生的對應計算狀態向量的集合 分子演算法**求解獨立集問題**(Y_0 , n, m)中的步驟 (0a) 到 (1d)形成 2^n 維希爾伯特空間 (複向量空間) 的$\mathbb{C}^{2^n}$ *標準正交基底*。■

C. *獨立集問題分子解計算狀態空間的量子電路與數學解*

根據**引理 5-1 ，為了解決具有** m *個*邊和 n *個頂點*的圖的獨立集合問題，產生 $2^{n\text{個可能的分子解}}$ 分子演算法**求解獨立集問題**(Y_0 , n, m)中的步驟 (0a) 到 (1d)形成希爾伯特空間 (複向量空間) 的標準正交基底$\mathbb{C}^{2^n}$。這就是說，每個可能的分子解對應於希爾伯特空間 ($\mathbb{C}^{2^n}$) 的正交基底中的一個元素。為了同時編碼 $2^{n\text{個}}$可能的分子解決方案，我們假設應用 n 位元量子暫存器 ($\otimes_{p=n}^{1} |y_p\rangle$) 來初始化具有$Q = 2$ "狀態的系統，這些狀態被標記為P_0 , P_1 , P_2 , …, $P_{\text{個}-1}$ ，其中每個狀態P_k為 $0 \le k \le 2n-1$對應於第k種可能的分子溶液。我們也假設具有一個量子位元 ( ) 的量子暫存器用於標記 $2^{n\text{個狀態}}$ $|1\rangle$中答案的幅度。為了完成這個目的，我們在狀態上使用一個哈達瑪門 $|1\rangle$ ，新的量子狀態向量是($\frac{1}{\sqrt{2}}$ ($|0\rangle - |1\rangle$)) 。

( ) 中的初始狀態 $\otimes_{p=n}^{1} |y_p\rangle$ 設定為 ($\otimes_{p=n}^{1} |y_p^0\rangle$)，我們假設 $|\lambda_0\rangle$ =

($\otimes_{p=n}^{1}|y_p^0\rangle$)。我們也假設初始量子態向量為 ($|\lambda_0\rangle$)。使用n *個*阿達瑪門對初始量子態向量 ($|\lambda_0\rangle$) 進行操作，系統具有$Q=2^n$狀態標示為$P_0, P_1, P_2, \ldots, P_{Q-1}$是

$$|\lambda_{5\text{-}1}\rangle=(H^{\otimes n})|\lambda_0\rangle=\frac{1}{\sqrt{2^n}}(\otimes_{p=n}^{1}(|y_p^0\rangle+|y_p^1\rangle))=\frac{1}{\sqrt{2^n}}(\Sigma_{y=0}^{2^n-1}|y\rangle)。 \quad (5\text{-}1)$$

在新的狀態向量 ($|\lambda_{5\text{-}1}\rangle$) 中，狀態 $|y_n{}^0 y_{n-1}{}^0 \ldots y_2{}^0 y_1{}^0\rangle$ 用幅度 ($\frac{1}{\sqrt{2^n}}$) 編碼*第一個*元素 $y_n{}^0 y_{n-1}{}^0 \ldots y_2{}^0 y_1{}^0$不包含任何頂點的分子解空間。狀態$|y_n{}^0 y_{n-1}{}^0 \ldots y_2{}^0 y_1{}^1\rangle$ 用幅度 ($\frac{1}{\sqrt{2^n}}$) 編碼*第二個*元素 $y_n{}^0 y_{n-1}{}^0 \ldots y_2{}^0$ 包含第一個頂點 v_1 的分子解空間的$y_1{}^1$。依此類推，有狀態$|y_n{}^1 y_{n-1}{}^1 \ldots y_2{}^1 y_1{}^1\rangle$ 振幅 ($\frac{1}{\sqrt{2^n}}$) 編碼*最後一個*元素 $y_n{}^1 y_{n-1}{}^1 \ldots y_2{}^1$ 包含 n *個*頂點的分子解空間 $y_1{}^1\{v_n v_{n-1} \ldots v_2 v_1\}$。

D. 用於實現 2^n 個可能選擇中合法獨立集分子解的量子電路和數學解

具有 n *個*頂點和 m *個邊的圖* G 的獨立集合問題的實例，在圖 4-3 中，在分子演算法 **Solve-independent-set-problem** (Y_0, n, m)的所有 m 次迭代中，在步驟 s (2a) 到 (2d)中產生的簡單布林電路用於辨識並在 2 n 個可能的選擇中標記獨立集。用於在圖 4-3 中的 2^n *個可能選擇*中標記合法獨立集的簡單布林電路是

$$(\wedge_{k=1}^{m}(\overline{y_i \wedge y_j})), \quad (5\text{-}2)$$

其中 y_i 和 y_j *位*分別表示第 k *條*邊上的*頂點* vi 和 v_j ， $ek=(vi, v_j)$，在 G *中*為 $1\le k\le$ *米*。布爾公式($\wedge_{k=1}^{m}()$) 由$\overline{y_i \wedge y_j}$ m *個* **NAND** 組成 營運 和 *米* **與**運算。這 操作 NAND 和 **AND** 分別由圖 5-1(a)和 5-1(b)的量子電路實現。因此，我們假設第二個量子暫存器具有 m *個*量子位， $|$*米 米*$_{-1} \ldots l_1\rangle$, 為 $1\le k\le m$ *，儲存*第 k *個* **NAND 閘**的計算結果，其形式為 ($\overline{y_i \wedge y_j}$)，對應於一次 **NAND** 運算。 | 中每個量子位元的初始狀態*米 米*$_{-1} \ldots l_1\rangle$ 在狀態|1> 中準備。第二個量子暫存器中的第 m 個量子位元 l_m *儲存了上一次*評估計算的結果 **NAND** 操作。

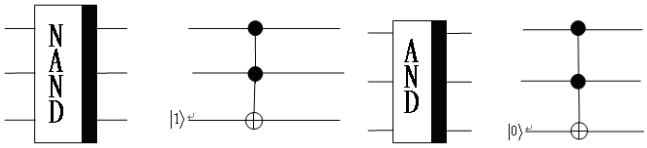


圖 5-1：(a)兩個布林變數的 **NAND** 運算，(b)兩個布林變數的 **AND** 運算。

接下來，為了評估前一個子句的 **AND 運算** (($k-1$)第) 與目前子句（第 k 個子句），第三個量子暫存器|*奧姆 奧姆*$_{-1} \ldots \emptyset_1 o_0\rangle$ 是必要的。第一個量子位元 |第三量子暫存器中的 o_0 *>最初在狀態*|1> *中*準備好。第三量子暫存器中的其他 m *位元最初處於*狀態|0>。第($m+1$) 個量子位元 $|o_m\rangle$ 在第三個量子暫存器中儲存了評估的結果 前一個子句的 **AND** 運算（ ($m-1$) 第一個子句）和*最後*一個子句（第 m 個子句）。這顯示第($m+1$)個量子位元|第三個暫存器中的 $o_{m>儲存}$所有子句的評估計算結果。我們使用**引理 5-2** 來展示圖 5-2 中的量子電路如何實現等式（5-2）中的簡單布林電路，以識別 2^n *個可能選擇*中的合法獨立集。

引理 5-2 ：為了解決任何具有 n *個*頂點和 m *個邊的*圖 $G=(V, E)$的獨立集問題，圖 5-2 中的**量子**電路 **LIS** 具有 (2× m) ***CCNOT*** 閘可以實現方程式 (5-2) 中的簡單布林電路 ($\wedge_{k=1}^{m}(\overline{y_i \wedge y_j})$)，並且是*已知在 2^n 個可能選擇*中標記合法獨立集的*最佳*量子電路。

證明：

具有 n *個*頂點和 m *個邊的圖* G 的獨立集合問題的實例，在圖 4-3 中，在分子演算法 **Solve-independent-set-problem** (Y_0, n, m)的所有 m 次迭代中，在步驟 s (2a) 到 (2d)中產生的簡單布林電路用於辨識並在 2 n 個可能的選擇中標記獨立的集合。我們使用等式（ 5-2）中的布林公式（ $\wedge_{k=1}^{m}$（ $\overline{y_i \wedge y_j}$））來表示簡單的布林電路，用於在圖 4-3 中的 2^n *個*可能選擇中標記合法的獨立集。我們展示如何實現圖 4-3 流程圖中的每條指令來完成證明。在圖 4-3 中，在語句 $S_{1中}$，循環索引變數 k 的值設定為一 (1)。接下來，在圖 4-3 的語句 $S_{2中}$，執行第一個迴圈的條件判斷。如果 k 的值小於或等於 m *的值*，則*下一個執行的*指令是圖 4-3 中的語句 S_3。否則，在圖 4-3 的語句 $S_{6中}$，執行 *End 指令，終止識別 2^n 個可能選擇*中合法獨立集的任務。

我們 假設第 k 條邊 e_k *為*(vi, v_j)，用位 y_i 和 y_j *分別表示頂點* vi *和* v_j。接下來，在圖 4-3 的語句 $S_{3中}$，包含一個頂點 (v_i 或者 v_j) 或零個*頂點被標記，並且包括*兩個頂點 vi *的*選項和 v_j 被丟棄。這就是說合法的獨立集合滿足( )$\overline{y_i \wedge y_j}$形式的公式。因此，一個 **CCNOT** 閘，($|l_k{}^1 \oplus$ *是我* $\bullet y_j\rangle$)，目標位 $l_k{}^1$和兩個受控位 y_i 和 y_j 用於實現與**非**門「 $l_k \leftarrow$ *圖 4-3* $\overline{y_i \wedge y_j}$語句 $S_{3中}$ *的*「 」，執行( )的結果$\overline{y_i \wedge y_j}$寫入目標位$|l_k{}^1\rangle$.接下來，在圖 4-3 的語句 $S_{4中}$，一個 **CCNOT** 閘, ($|o_k{}^0 \oplus lk \bullet o_{k-1}\rangle$)，以目標位 $o_k{}^0$和兩個受控位 l_k和 o_{k-1}來實現邏輯與運算「 $o_k \leftarrow lk \wedge ok_1$ 」即式(5-2)中($\wedge_{k=1}^{m}()$)$\overline{y_i \wedge y_j}$ *中的*第 k *個****與閘***。實施的結果 ($l_k \wedge ok_1$)寫入目標位 $|$*哦*$^0\rangle$ 。

接下來，在圖 4-3 的語句 S_5 中，第一個迴圈中索引變數 k 的值遞增。重複執行圖 4-3 的語句 S_2 到 S_5，*直到* $S_{2中}$的條件判斷得到*假*值。根據圖 4-3，***與非門***的總數為 m。邏輯和運算使用總數 m ***與***門。因此，量子閘在 2^n *個可能的選擇中*識別合法獨立集的標記的成本是（2× m) ***CCNOT*** 門。如**引理 4-3** 的證明所示，圖 4-3 中的語句 S_3 和 S_4 *之間*存在***真正的依賴關係***。***真正的***依賴 S_3 和 S_4 之間不能被打破。因此，圖 4-3 中的每條語句都必須以順序模式執行。基於上述陳述，圖 5-2 中的量子電路 **LIS 可以**實現等式（5-2）中的簡單布林電路（ $\wedge_{k=1}^{m}()$）$\overline{y_i \wedge y_j}$。

| 中的每一位*長*^{1} *米*$_{-1}{}^1$ … 圖 5-2 中的$l_1{}^1\rangle$是輔助量子位，用於儲存( )形式的各子句的求值結果$\overline{y_i \wedge y_j}$。因此，這

一步驟需要 m **NAND** 運算透過關係式 ($|l_k{}^1 \oplus 是_{我} \bullet y_j>$)，目標位 $l_k{}^1$和兩個控制位 y_i和 y_j，其中 $1 \le 我$和 $j \le n$ 和 $1 \le k \le 米$。在圖 5-2 中，|中的每一位$奧米_0$ $奧姆_{-1}{}^0$ … $Ø_1{}^0$ $o_0{}^1$ > 也是輔助量子位，用於儲存 ($k-1$) 第 k 個子句和$第\ k$ 個子句，其中 $1 \le k \le 米$。這一步需要 m 透過關係 ($|o_k{}^0 \oplus lk \bullet o_{k-1}>$) 目標位 $o_k{}^0$和兩個控制位 l_k和 o_{k-1}為 $1 \le k \le 米$。從上述陳述，立即推斷圖 5-2 **的**量子電路 **LIS** 與 ($2\times m$) ***CCNOT*** 閘可以實現方程式 (5-2) 中的簡單布林電路 ($\wedge_{k=1}^{m}(\overline{y_i} \wedge y_j)$)，並且是*已知在* $2^{n\ 個可能選擇}$中標記合法獨立集的*最佳*量子電路。 ■

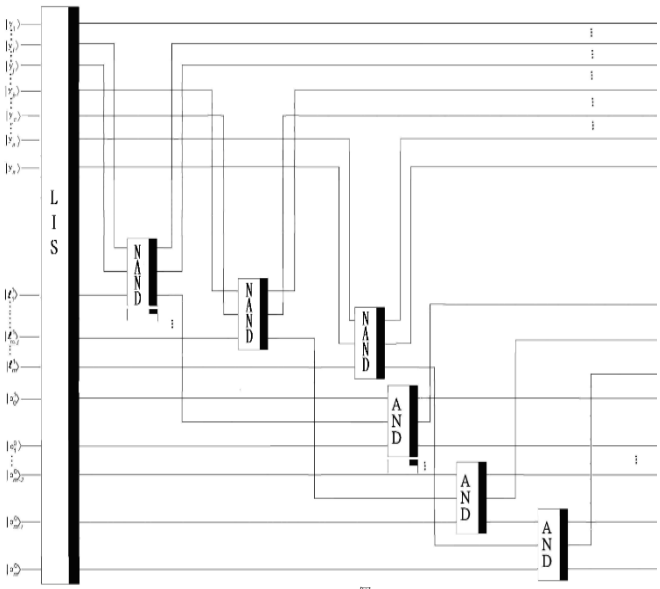


圖 5-2：量子電路 LIS，用於在 $2^{n\ 個}$可能的選擇中標記合法的獨立集。

E. 量子電路與最大獨立集分子解的數學解

圖 4-4 中的簡單布林電路在分子演算法**求解獨立集問題**(Y_0, n, m) 的每次迭代中從步驟(4a) 到(4b) 獲得，對每個解中的頂點數進行計數。圖 4-4 中用於計算每個合法獨立集合中頂點數量的簡單布林電路是

($w_{1,1} \leftarrow 奧姆 \wedge y_1$) 和 ($w_{1,0} \leftarrow 奧姆 \wedge \overline{y_1}$) 和 (5-3)

($w_{i+1,j+1} \leftarrow 我_{+1} \wedge w_{i,j}$) 和 ($w_{i+1,j} \leftarrow \overline{y_{i+1}} \wedge w_{i,j}$) 為 $1 \le 我 \le n-1$ 和 $0 \le j \le 我$。 (5-4)

輔助量子比特|$w_{i+1,j}$>和|執行這些操作需要 $w_{i+1,i+1}$>，其中 $0 \le i \le n-1$ 和 $0 \le j \le i$ 。對於 $0 \le 我 \le n-1$ 和 $0 \le j \le i$ ，|中的每個量子位 $w_{i+1,j}$>和| $w_{i+1,i+1}$> 最初在狀態 |0> 中準備。我們假設對於 $0 \le 我 \le n-1$ 和 $0 \le j \le 我$，量子比特| $w_{i+1,j+1}$>記錄 y_{i+1}對個數的影響後，有$(j+1)$*個個的管(組)* $Y_{j+1\ 的狀態}$。我們也假設對於 $0 \le 我 \le n-1$ 和 $0 \le j \le 我$，量子比特| $w_{i+1,j}$>是記錄受 y_{i+1} 個數影響後有 j *個的管(組)* $Y_{j\ 的狀態}$。我們使用**引理 5-3** 來展示圖 5-3 到 5-4 中的量子電路如何實現等式（5-3）和等式（5-4）中的簡單布林電路，以計算每個電路中的頂點數合法獨立設定。

引理 5-3 ：圖 5-3 中的量子電路 **CFFV** 實現了簡單的布林電路($w_{1,1} \leftarrow 奧姆 \wedge y_1$) 和 ($w_{1,0} \leftarrow 奧姆 \wedge \overline{y_1}$）如式（5-3）所示。圖 5-4 中的量子電路 **CMO** 實現了簡單的布林電路 ($w_{i+1,j+1} \leftarrow 我_{+1} \wedge w_{i,j}$) 和 ($w_{i+1,j} \leftarrow \overline{y_{i+1}} \wedge w_{i,j}$) 為 $1 \le 我 \le n-1$ 和 $0 \le j \le$ 方程式（5.4）中的 i 。

證明：

如果一個合法的獨立集合有第一個頂點 v_1，那麼它就滿足簡單的布林電路 ($w_{1,1} \leftarrow 奧姆 \wedge$ 式(5-3)中的 y_1)，否則，它滿足簡單的布林電路($w_{1,0} \leftarrow 奧姆 \wedge \overline{y_1}$）如式（5-3）所示。確定了 y_1對 1 個數的影響後，量子位元|編碼位元 $w_{1,1}$ $的 w_{1,1}$>將記錄哪些合法獨立集 s 只有一個值 1 並包含第一個頂點 v_1 。 量子位|編碼位元 $w_{1,0}$ $的 w_{1,0}$>將記錄哪些合法獨立集 s 有 *0* 個且不包含第一個頂點 v_1。因此，一 **CCNOT** 閘 ($|w_{1,1}{}^0 \oplus 奧姆 \bullet y_1>$) 與目標位 $|w_{1,1}{}^0>$ 和兩個控制位 $|Ø_{米}>$ 和 $|y_1>$ 實現 ($w_{1,1} \leftarrow 奧姆 \wedge y_1$)（等式（5-3）的第一個條件）。一個非門**操作**於$|y_1>$($|\overline{y_1}>$) 和另一個 **CCNOT** 閘 ($|w_{1,0}{}^0 \oplus 奧姆 \bullet \overline{y_1}>$) 與目標位 $|w_{1,0}{}^0>$ 和兩個控制位 $|Ø_{米}>$ 和 $|\overline{y_1}>$ 實施 ($w_{1,0} \leftarrow 奧姆 \wedge \overline{y_1}$）（式（5-3）的第二個條件）。接下來，另一個**非**門在 | 上運行。 $\overline{y_1}>(|y_1>)$將會恢復 $|y_1>$ 在 $|y_n \ldots y_1>$ 到其疊加狀態。這就是說，如果量子位元的值| $w_{1,1}>$ 等於 1，則量子位元| $w_{1,1}{}^1>$ 表示哪一個合法獨立集 s 只有一個值 1 且包含第一個頂點 v_1 。類似地，如果量子位元的值| $w_{1,0}>$ 等於 1，則量子位元| $w_{1,0}{}^1>$ 表示哪些合法獨立集 s 不包含第一個頂點 v_1 且有零個。根據上述陳述，圖 5-3 中的量子電路 **CFFV 實現了方程式（5-3）的第一個和第二個條件。**

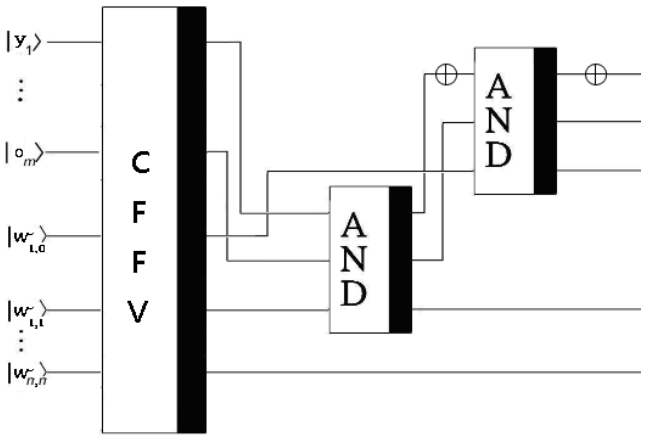


圖 5-3：使用量子電路 CFFV 實現等式（5-3）的第一個和第二個條件。

接下來，如果一個合法的獨立集合包含第 ($i+1$) 個頂點 v_{i+1}並且有 j 個，那麼它滿足簡單的布林電路 ($w_{i+1,j+1} \leftarrow 我_{+1} \wedge w_{i,j}$) 與 $1 \le 我 \le n-1$ 和 $0 \le j \le$ 式(5-4)中的 i 。如果合法獨立集不包含第 ($i+1$) 個頂點 v_{i+1}並且有 j 個，則它滿足簡單的布林電路 ($w_{i+1,j} \leftarrow \overline{y_{i+1}} \wedge w_{i,j}$) 與 $1 \le 我 \le n-1$ 和 $0 \le j \le$ 方程式（5.4）中的 i 。*確定了* y_{i+1}對 1 個數的影響後，量子位元| $w_{i+1,j+1}>$ 編碼位元 $w_{i+1,j+1}$將記錄哪些合法獨立集 s 有 $(j+1)$ 個並且包含第 ($i+1$) 個頂點 v_{i+1} 。量子位| $w_{i+1,j}>$ 編碼位 $w_{i+1,j}$將記錄哪些合法獨立集 s 有 j 個且不包含第 ($i+1$) 個頂點 v_{i+1} 。

因此，一個 **CCNOT** 閘 ($|w_{i+1,j+1}{}^0 \oplus ,j \bullet y_{i+1}>$) 與目標

位 $|w_{i+1,j+1}{}^0>$ 和兩個控制位 $|w_{i,j}>$ 且 $|y_{i+1}>$ 實施 ($w_{i+1,j+1} \leftarrow 我_{+1} \wedge w_{i,j}$) 為 $1 \le 我 \le n-1$ 和 $0 \le j \le i$（等式（5-4）的第一個條件）。運行在量子位元上**的**一個非閘 $|y_{i+1}>$ ($|\overline{y_{i+1}}>$) 和另一個 **CCNOT** 閘 ($|w_{i+1,j}{}^0 \oplus ,j \bullet \overline{y_{i+1}}>$) 與目標位 $|w_{i+1,j}{}^0>$ 和兩個控制位 $|w_{i,j}>$ 且 $|\overline{y_{i+1}}>$ 實施 ($w_{i+1,j} \leftarrow \overline{y_{i+1}} \wedge w_{i,j}$) 為 $1 \le 我 \le n-1$ 和 $0 \le j \le i$（等式（5-4）的第二個條件）。接下來，使用另一個對量子位元進行操作的 $\overline{y_{i+1}}$**非閘** $|>(|y_{i+1}>)$ 將會恢復 $|y_{i+1}>$ 在 $|$ 中 $y_n \ldots y_1>$ 到其疊加態。這意味著如果量子位元的值 $|w_{i+1,j+1}>$等於 1，則量子位元 $|w_{i+1,j+1}{}^1>$ 將指示哪個合法獨立集 s 有 $(j+1)$ 個並且包含第 $(i+1)$ 個頂點 v_{i+1}。類似地，如果量子位元的值 $|w_{i+1,j}>$ 等於 1，則量子位元 $|w_{i+1,j}{}^1>$ 將指示哪個合法獨立集 s 不包含第 $(i+1)$ 個頂點 v_{i+1} 並且有 j 個。根據上述陳述，圖 5-4 中的量子電路 **CMO** 實現了方程式（5-4）的第一個和第二個條件。

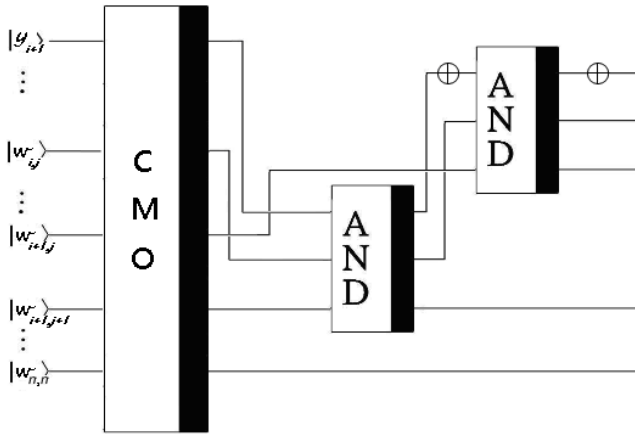


圖 5-4：使用量子電路 CMO 實現(5-4)的第一個和第二個條件。

因此，從上面的陳述可以推斷，圖 5-3 中的量子電路 **CFFV** 可以實現簡單的布林電路 ($w_{1,1} \leftarrow 奧姆 \wedge y_1$) 和 ($w_{1,0} \leftarrow 奧姆 \wedge \overline{y_1}$）在方程式（5-3）。類似地，可以推論圖 5-4 中的量子電路 **CMO** 可以實現簡單的布林電路($w_{i+1,j+1} \leftarrow 我_{+1} \wedge w_{i,j}$) 和 ($w_{i+1,j} \leftarrow \overline{y_{i+1}} \wedge w_{i,j}$) 為 $1 \le 我 \le n-1$ 和 $0 \le j \le$ 方程式（5.4）中的 i。 ■

F. 用於讀取最大獨立集分子解的量子電路和數學解

分子演算法中的步驟 (0a)到(1d)所創建的 $2^{n個}$可能的分子解 **求解獨立集合問題**(Y_0, n, m) 在分佈中初始化：$(\frac{1}{\sqrt{2^n}}\frac{1}{\sqrt{2^n}}\frac{1}{\sqrt{2^n}}\ldots\frac{1}{\sqrt{2^n}})$。這表明 $2^{n種}$可能的分子溶液中的每一種都有相同的振幅。先前提出的量子電路已經標記了答案，但找到答案的幅度或機率將呈指數下降。因此，基於 [26]，擴散算子用於以指數方式增加找到答案的幅度或機率，並由矩陣 G *定義*如下：$G_{i,j}=(2/2^n)$ if $i \ne j$ 且 $G_{i,i}=(-1+(2/2^n))$。**演算法 5 -1** 用於衡量由下列項目產生的答案 分子演算法中的步驟(5a)和(5b) **解獨立集合問題**(Y_0, n, m)。

為了表達方便，我們假設 那 $|y_b{}^1>, |l_k{}^1>, |ok^{1>}, |w_{i+1,j}{}^1>$ 和 $|w_{i+1,i+1}{}^1>$ 為 $1 \le 乙 \le$, $0 \le k \le 米$, $0 \le 我 \le n-1$ 和 $0 \le j \le$ 隨後， i *表示其對應的*量子位元的值為 1。 假設 $|y_b{}^0>$, $|l_k{}^0>$, $|ok_0{}^>$, $|w_{i+1,j}{}^0>$ 且 $|w_{i+1,i+1}{}^0>$ 為 $1 \le 乙 \le$, $0 \le k \le 米$, $0 \le 我 \le n-1$ 和 $0 \le j \le$ 隨後， i 表示它們對應的量子位元的**值為 0** 。 下面在前面的小節。我們使用**演算法 5-1** 中的第一個參數 t *來表示合法答案中頂點集的最大大小*，下一小節**演算法 5 -2** 中步驟(1a)的執行傳遞其值。

演算法 5 -1 (t)：透過讀取最大尺寸的分子解得到數學解任何具有 m *個*邊和 n *個頂點的圖* G 的獨立集合。

(0) 酉算符 $\mathbf{U}_{init}=(H)(\otimes^1_{i=n}\otimes^0_{j=i}I_{2\times2})(\otimes^1_{k=m}I_{2\times2})(I_{2\times2})(\otimes^1_{k=m}I_{2\times2})(H^{\otimes n})$，對初始量子進行運算 狀態向量$(|1>)(\otimes^1_{i=n}\otimes^0_{j=i}|w_{i,j}{}^0>)(\otimes^1_{k=m}|o_k{}^0>)(|o_0{}^1>)(\otimes^1_{k=m}|l_k{}^1>)(\otimes^1_{b=n}|y_b{}^0>)$ 和 $2^n n$ *位*的可能選擇（包含所有可能的獨立集）是

$$|\varphi_{0,0}>=(\frac{1}{\sqrt{2}}(|0>-|1>))\frac{1}{\sqrt{2^n}}(\otimes^1_{i=n}\otimes^0_{j=i}|w_{i,j}{}^0>)(\otimes^1_{k=m}|o_k{}^0>)(|o_0{}^1>)(\otimes^1_{k=m}|l_k{}^1>)(\otimes^1_{b=n}(|y_b{}^0>+|y_b{}^1>))。$$

(1) 為了標記 $2^{n個}$可能的選擇中哪些是合法的獨立集 s，哪些不是答案，圖 5-2 中的量子電路，$(I_{2\times2})(\otimes^1_{i=n}\otimes^0_{j=i}I_{2\times2})$ **(LIS)** ，用於對量子狀態向量進行運算 $|\varphi_{0,0}>$ ，得到如下新的量子態向量

$$|\varphi_{1,0}>=(\frac{1}{\sqrt{2}}(|0>-|1>))\frac{1}{\sqrt{2^n}}(\otimes^1_{i=n}\otimes^0_{j=i}|w_{i,j}{}^0>)$$

$$\sum^{2^n-1}_{y=0}(\otimes^1_{k=m}(|ok_0\oplus lk\bullet ok_1>)(|o_0{}^{1>}-)(\otimes^1_{k=m}|l_k{}^1\oplus 是_{我}\bullet y_j>)(|y>))。$$

(2) 用於實施 ($w_{1,1} \leftarrow 奧姆 \wedge y_1$) 和 ($w_{1,0} \leftarrow 奧姆 \wedge \overline{y_1}$）在方程式（5-3）中，將圖 5-3 中的量子電路 $(I_{2\times2})$ **(CFFV)** 應用於量子狀態向量 $|\varphi_{1,0}>$，則下列新的量子態向量為

$$|\varphi_{2,0}>=(\frac{1}{\sqrt{2}}(|0>-|1>))\frac{1}{\sqrt{2^n}}(\otimes^2_{i=n}\otimes^0_{j=i}|w_{i,j}{}^0>)(\sum^{2^n-1}_{y=0}(|w_{1,1}{}^0\oplus 奧姆\bullet y_1>)(|w_{1,0}{}^0\oplus 奧姆\bullet\overline{y_1}>)(\otimes^1_{k=m}|o_k>)(|o_0{}^1>)(\otimes^1_{k=m}|l_k>)(|y>))。$$

(3)**當** $i=1$ **時** $n-1$

(4)**對於** $j=i$ **降至** 0

(4a) 圖 5-4 中的量子電路 $(I_{2\times2})$ **(CMO)**是確定合法獨立集之間的頂點數，並對量子狀態向量 $(|\varphi_{2+\left(\sum^{i-1}_{\theta_1=0}(\theta_1+1)\right)+(i-j),0}>)$ 進行操作。由於步驟(4a)嵌入在唯一循環中，重複執行圖 5-4 中的量子電路後，$(I_{2\times2})$ **(CMO)** ，計算每個合法獨立集合中頂點數的結果狀態向量為

$|\varphi_{2+\frac{n^2+n-2}{2},0}\rangle$ = ($\frac{1}{\sqrt{2}}$ (|0> −|1>)) $\frac{1}{\sqrt{2^n}}$($\sum_{y=0}^{2^n-1}$($\otimes_{i=n}^{1}\otimes_{j=i}^{0}|w_{i,j}\rangle$)) ($\otimes_{k=m}^{1}|o_k\rangle$) (| $o_0{}^1$>) ($\otimes_{k=m}^{1}|l_k\rangle$) (| y >))。

結束於

結束於

(5) ***CNOT*** 門 ($\frac{|0\rangle-|1\rangle}{\sqrt{2}}\oplus w_{n,t}$) 與目標位 | $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$> 和控制位元| $w_{n,t}$ > 標記合法獨立 設(s) 為量子態向量 (| $\varphi_{2+\frac{n^2+n-2}{2},0}$ >) 中頂點數最大的集合，則新的量子態向量為

$|\varphi_{2+\frac{n^2+n-2}{2}+1,0}\rangle = (\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle))\frac{1}{\sqrt{2^n}}\times(-1)^{w_{n,t}}(\sum_{y=0}^{2^n-1}(\otimes_{i=n}^{1}\otimes_{j=i}^{0}|w_{i,j}\rangle)(\otimes_{k=m}^{1}|o_k\rangle)(|o_0{}^1\rangle)(\otimes_{k=m}^{1}|l_k\rangle)(|y\rangle))$。

、(2)和(1)執行的所有運算反轉可以將輔助量子位元恢復到其初始狀態。

(7) 將擴散算子應用到量子態向量上

步驟(6)中產生。

(8) 反覆 執行步驟(1)到步驟(7)至多 $O(\sqrt{\frac{2^n}{R}})$次，其中 R 的值是解的數量，可以透過量子計數演算法有效地確定 [28, 41]。

(9) 得到答案的成功機率為

測量完成後至少 (1 / 2) 。

結束演算法

引理 5 -4：演算法 5-1 的輸出是透過讀取任何具有 m 條邊和 n *個頂點的圖* G *的*最大尺寸獨立集的分子解而獲得的數學解。

證明：

由於對於任何具有 m *個*邊和 n *個頂點的圖* G *，獨立*集問題有 $2^{n\text{個}}$可能的選擇（包括所有可能的獨立集） ，*因此 n 位*的量子暫存器($\otimes_{b=n}^{1}|y_b\rangle$) 可以表示 $2^{n\text{個}}$選擇，初始值為狀態向量 ($\otimes_{b=n}^{1}|y_b{}^0\rangle$)。任何具有 m *個*邊和 n *個頂點的圖* G *的*獨立集問題需要找到 G 中最大尺寸的獨立集，因此這些輔助量子暫存器是必要的。透過執行步驟(0)，得到初始狀態向量| Ω> = (|1>) ($\otimes_{i=n}^{1}\otimes_{j=i}^{0}|w_{i,j}{}^0\rangle$) ($\otimes_{k=m}^{1}|o_k{}^0\rangle$) (| $o_0{}^1$>) ($\otimes_{k=m}^{1}|l_k{}^1\rangle$) ($\otimes_{b=n}^{1}|y_b{}^0\rangle$) 開始量子計算獨立集問題。酉算符 $\mathbf{U}_{init}$= (H) ($\otimes_{i=n}^{1}\otimes_{j=i}^{0}I_{2\times2}$) ($\otimes_{k=m}^{1}I_{2\times2}$) ($I_{2\times2}$) ($\otimes_{k=m}^{1}I_{2\times2}$) ($H^{\otimes n}$) 對初始狀態向量 | 運算。 Ω>，結果狀態向量變成| $\varphi_{0,0}$> 有 $2^{n\text{個}}$選擇。這表明 $2^{n\text{個}}$可能的分子選擇 由分子演算法中的步驟 s (0a)至(1d)產生的**解決獨立組問題**(Y_0, *n,m) 可以透過***演算法 5-1*** 中的步驟(0)來實現。

演算法 5 -1 中的步驟(1)充當酉算符 **LIS** ，即圖 5 -2 中的量子電路。在執行**演算法 5-1 中的步驟(1)時**，2n 個可能的選擇中的那些*滿足*等式 (5-2) 中的 *簡單*布林電路 ($\wedge_{k=1}^{m}(\,)$)$\overline{y_i\wedge y_j}$的選擇被標記。步驟(1)執行完成後，得到狀態向量| $\varphi_{1,0}$>得到， 包含那些帶有 | 的選項 $o_m{}^1$> 表示它們是合法的獨立集，而帶有 | 的非法選擇 $o_m{}^0$>不符合條件。因此，可以實現由分子演算法中的步驟 s (2a) 至 (2d) 產生的方程式 (5-2) 中的簡單布林電路(($\wedge_{k=1}^{m}$) $\overline{y_i\wedge y_j}$)**求解獨立集問題**(Y_0, *n, m) 透過****演算法 5 -1*** 中的步驟(1) 。

演算法 5 -1 中的步驟(2)充當酉算符 **CFFV** ，對應於圖 5-3 中的量子電路。在執行**演算法 5-1 中的步驟(2)時**，計算每個合法獨立集中第一個頂點影響的個數。步驟(2)執行後，狀態向量| $\varphi_{2,0}$>得到，其中包括 es 那些帶有 | 的合法獨立集。 $w_{1,1}{}^1$> 有*一個*並且包含*第一個*頂點和那些帶有 | 的合法獨立集 $w_{1,0}{}^1$> 具有*零*個且不包含第一個頂點。這意味著簡單的布林電路 ($w_{1,1}\leftarrow$ *奧姆* $\wedge y_1$) 和 ($w_{1,0}\leftarrow$ *奧姆* $\wedge$) 在$\overline{y_1}$*第一個步*驟（ 4a ）和（ 4b ）產生的方程式（5-3）中**求解獨立集問題**(Y_0, *n, m)*中的迭代(0, 0)可以透過**演算法 5-1** 中的步驟(2)來實現。

演算法 5-1 中第一個迴圈中的唯一語句，並且充當酉算符 **CMO，對應於圖** 5-4 中的量子電路。這一步是確定合法獨立集合中 1 的個數（頂點個數）。步驟(3)和步驟(4)各構成一個兩級循環。當索引變數 i 的值等於 1 且索引變數 j *的值*從 1 到 0 時，重複執行步驟(4a)兩次。類似地，當索引變數 i *的值*等於 2 且索引變數 j *的值*從 2 到 0 時，重複執行步驟(4a)3 次。類似地，當索引變數 i 的值等於 (n 1) 索引變數 $-j$ 的值來自於 (n −1) 降到零，步驟(4a) 重複執行 n 次。也就是說，步驟(4a)的執行總數為$(2+3+\ldots n)=(n^2+n-2)/2$. 因為狀態向量| $\varphi_{2,0}$> 由步驟(2)生成，其索引為 2(二)，重複執行步驟(4a)後，我們使用 $2+((n^2+n-2)/2)$ 作為結果狀態和結果狀態向量的索引 | $\varphi_{2+\frac{n^2+n-2}{2},0}$ >獲得 其中計算每個合法獨立集中的頂點數。這顯示簡單的布林電路 ($w_{i+1,j+1}\leftarrow$ *我*$_{+1}\wedge w_{i,j}$) 和 ($w_{i+1,j}\leftarrow\overline{y_{i+1}}\wedge w_{i,j}$) 為 1≤ *我* ≤ n −1 和 0≤ j ≤ **求解獨立集問題**(Y_0, *n, m)*中的同一迭代(i , j *) 中的*步驟 (4 a) 和 (4 b) 中產生的方程式 (5-4)中的 i 可以透過 Step 實現**演算法 5-1** 中的(4a) 。

接下來，一個 **CNOT** 門，（$\frac{|0\rangle-|1\rangle}{\sqrt{2}}\oplus w_{n,t}$) 與目標位 | $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$> 和控制位元| $w_{n,t}$ >，在**演算法 5.1** 的步驟(5)的執行中以階段(1)標示答案−。得到的狀態向量| $\varphi_{2+\frac{n^2+n-2}{2}+1,0}$ > 由答案中階段 (−1) 的部分和階段 (+1) 的另一部分組成。由於量子操作本質上是可逆的，因此執行步驟(6)將反轉步驟(4a) 、步驟(2)和步驟(1)完成的所有操作，從而可以將輔助量子位元恢復到其初始狀態。接下來，在執行**演算法 5-1 中的步驟(7)時**，應用擴散算子來完成增加測量答案的成功機

率的任務。**演算法 5 -1** 的步驟(8)中，重複執行後 步驟(1) 到(7) $O(\sqrt{\frac{2^n}{R}})$次，產生最大成功機率。接下來，透過執行**演算法 5 -1 中的步驟(9)**，獲得測量結果並將答案返回到**演算法 5 -2** 。由於**演算法 5-1** 中每一步產生的結果是有限維希爾伯特空間中的單位向量，因此，我們立即推斷**演算法 5 -1 的輸出是透過讀取**最大值的分子解得到的數學解。 *m 個邊*和 *n 個頂點的*任何圖 *G 的大小獨立集合*。■

G. 求解任意具有 m 個邊和 n 個頂點的圖 G 上的獨立集合問題

以下演算法解決了任何具有 *m 個*邊和 *n 個頂點的圖 G 的*獨立集問題。我們已經使用了前面小節中**演算法 5 -2** 所使用的符號。

演算法 5-2 ：解任意具有 *m 個*邊和 *n 個頂點的*圖 G 的獨立集合問題。

(1)**對於** $t = n$ **至** 1

(1a) 呼叫**演算法 5-1** (t)。

(1b)**如果***第 t 次*執行 Step 得到答案

(1a)**那麼**

(1c)終止**演算法 5-2** 。

結束如果

結束於

結束演算法

引理 5-5 ：**演算法 5-2** 取得任意具有 *m 個*邊和 *n 個頂點的圖 G* 中獨立集問題的最大獨立集合。

證明：

演算法 5 -2 中的步驟(1a)的每次執行中，都會呼叫**演算法 5-1 來**完成兩個主要任務。第一個任務是計算每個合法獨立集中的頂點數。這表明，對於任何具有 *m 個*邊和 *n 個頂點的圖 G，在獨立*集問題中尋找最大尺寸獨立集的分子解的數學解是有限維希爾伯特空間中的單位向量。第二個任務是使用擴散算子，它以指數方式增加從最大獨立集 s 的分子解中測量答案的成功機率。由此我們證明，透過讀取任何具有 *m 個*邊和 *n 個頂點的圖 G 的*最大尺寸獨立集的分子解而獲得的數學解仍然是有限維希爾伯特空間中的單位向量。接下來，在演算法 5-2 中的步驟(1b)的每次執行中，如果從**演算法 5-2** 中的步驟(1a)的*第 t 次執行中*找到答案，則步驟的*第 t 次*執行**演算法 5 -2** 中的(1c)將終止**演算法 5 -2** 。否則，重複執行步驟（1a）至（1c），直到找到任何具有 *m 個*邊和 *n 個頂點的圖 G 的獨立*集問題的答案。因此，立即得出**演算法 5-2 可用於獲得**任何具有 *m 個*邊和 *n 個頂點的圖 G 的獨立*集問題的答案。■

H. Durr-Hoyer 演算法和 Ahuja-Kapoor 演算法以及量子存在性測試無法解決任何具有 m 個邊和 n 個頂點的圖 G 的獨立集問題

許多資訊處理和計算問題都可以追溯到尋找資料庫或成本函數的極值問題。 Durr Hoyer 演算法 [48] 在未排序的資料庫或具有 2^{-n} *個項目*的成本函數中找到滿足任何給定條件的最小值。 Ahuja Kapoor 的演算法 [49] 在未排序的資料庫或具有 2^{-n} *個項目*的成本函數中找到滿足任何給定條件的最大值。為了提高兩種演算法的性能，整合了量子計數和二分搜尋的量子存在測試[31] 可用於在未排序的資料庫或具有 2^n *個項目*的成本函數中尋找滿足任何給定條件的最小值或最大值。任何具有 *m 個*邊和 *n 個頂點的圖 G 的*獨立集問題需要在未排序的資料庫或具有 2^n *個*頂點子集的成本函數中找到具有最大數量頂點的最大獨立集。我們用下面的引理來說明為什麼 Durr –Hoyer 演算法、Ahuja –Kapoor 演算法和量子存在 測試無法解決任何具有 *m 個*邊和 *n 個頂點的圖 G* 的獨立集問題。

引理 5-6 ：Durr –Hoyer 演算法、Ahuja –Kapoor 演算法與量子存在 測試無法解決任何具有 *m 個*邊和 *n 個頂點的圖 G* 的獨立集問題。

證明：

在 Durr –Hoyer 演算法、 –Ahuja Kapoor 演算法和量子存在中 測試演算法，解空間 Y 是一組 2^n *個*可能的選擇， Y 等於 $\{ y_n y_{n-1} \dots y_2 y_1 | \forall 年 \in \{0, 1\}$ 為 $1 \le d \le n \}$。這表示 Y 中每個元素的長度為 *n 位，每個元素代表* 2^n *種可能選擇之一*。為了方便表述，我們假設 $y_d{}^{0表示} y_d$的值為 0， $y_d{}^{1表示} y_d$的值為 1。第*一個*元素 $y_n{}^0 y_{n-1}{}^0 \dots y_2{}^0 y_1{}^0$編碼十進制值 0（零）。*第二*個元素 $y_n{}^0 y_{n-1}{}^0 \dots y_2{}^0 y_1{}^1$編碼十進位值 1（一）。*第三*個元素 $y_n{}^0 y_{n-1}{}^0 \dots y_2{}^1 y_1{}^0$編碼十進位值 2（二）。依此類推，*最後一個*元素 $y_n{}^1 y_{n-1}{}^1 \dots y_2{}^1 y_1{}^1$編碼十進位值 $2^n - 1$. 由於解空間不包含任何頂點子集，因此這三種演算法無法找到最大尺寸的獨立集。因此，從上面的陳述，我們立即得出，Durr –Hoyer 演算法、Ahuja –Kapoor 演算法和量子現有測試不能解決任何具有 *m* 條邊和 *n 個頂點*的圖 *G* 的獨立集問題。■

VI. 性評估

在本節中，我們估計**演算法 5-2 的**時間複雜度和空間複雜度，用於解決任何具有 *m 個*邊和 *n 個頂點的圖 G* 的獨立集問題。隨後，我們證明**演算法 5-2 為解決任何**具有 *m 個*邊和 *n 個頂點的圖 G* 的獨立集問題提供了二次加速，這是*最好的 已知的*加速問題。

A. 演算法 5-2 的時間與空間複雜度

引理 6 - 1 ：**演算法 5-2** 的最佳情況時間複雜度涉及 $((2^{n/2} \times (2 \times n)) + (n + 1))$**阿達瑪**門，$(2^{n/2} \times (2 \times (n^2 + n)))$ **NOT** 門，$(2^{n/2})$ **CNOT** 門，$(2^{n/2} \times (4 \times m + (2 \times (n^2 + n))))$ **CCNOT 閘，** *n 個量子位元的*$(2^{n/2})$ 相移閘和量子測量。

證明：

演算法 5-2 中，步驟(1)是主循環，並且嵌入在該郵件循環中的步驟被執行 *n 次*迭代。因此，步驟(1a)的第一次執行呼叫**演算法 5 -1** 。在**演算法 5-1** 的步驟(0)中，應用(n +1) **個 Hadamard 閘。**接下來，在**演算法 5 -1** 的步驟(1)中，(2× m)應用 **CCNOT 門。接下來，在演算法 5 -1** 的步驟(2)中，應用兩個 **CCNOT** 閘和兩個 **NOT 閘。那麼，演算法 5 -1** 的步驟(4a)是第一個指令中的唯一指令。 循環，**演算法 5-1 的步驟 (4a)**得到 ($n^2 + n - 2$)**非**門和 ($n^2 + n - 2$） **CCNOT** 門。接下來，在**演算法 5-1** 的步驟(5)中，應用一個 **CNOT 閘。**

然後，演算法 5 -1 的步驟(6)將輔助量子位元恢復到原始狀態。因此，**演算法 5 -1** 的步驟 (6)產生 (n^2+n) **NOT** 閘和 ((2× m) + (n^2+n)) **CCNOT** 門。這就是說，步驟（1）到（6）完成了預言機工作，並將答案標記為階段（ −1）。從**演算法 5 -1**的步驟(7)可以清楚看出，執行了一個擴散算子。在**演算法 5 -1** 的步驟(8)中，實作($\sqrt{2^n}$) 預言機工作和($\sqrt{2^n}$) 擴散算子是最壞的情況，因為 R 的值等於 1，而這種情況是最壞的情況。我們假設 n *個量子位元的相移閘* U_{PSG} 的作用如下：

$$\text{巴黎聖日爾曼大學}: \begin{cases} |x> \rightarrow -|x>, & x \neq 0 \\ |0> \rightarrow |0> \end{cases}.$$

由於根據[28, 41]，單一物理操作可以完成 n *個*量子位元 U_{PSG} *的受控相移門*，因此它是基本閘。由[28, 41] 我們可以得到擴散算子的分解， $H^{\otimes n}$ **大學** $_{PSG}$ $H^{\otimes n}$ ，可以實現各個擴散算子。接下來，在**演算法 5-1** 的步驟(9)中，進行測量。因此，在第一次呼叫**演算法 5 -1 後**，得出 (($2^{n/2}$ ×(2× n)) + (n + 1))**阿達瑪**門，($2^{n/2}$ ×(2 ×(n^2+n))) **NOT** 門，($2^{n/2}$) **CNOT** 門，($2^{n/2}$ ×(4× 實現了 m + (2 ×(n^2+n)))) **CCNOT** 閘、 n *個量子位元的* ($2^{n/2}$) 相移閘和量子測量。

演算法 5-1 的第一次呼叫完成後，然後 **在第一次執行演算法 5-2** 中的步驟(1b)時，如果 **演算法 5-2** 中的步驟(1a)的第一次執行返回答案，則**演算法 5-2 中的步驟(1c)的第一次執行時演算法 5 -2** 終止。因此，**演算法 5 -2 的**時間複雜度的最佳情況涉及 (($2^{n/2}$ ×(2× n)) + (n + 1))**阿達瑪**門，($2^{n/2}$ ×(2 ×(n^2+n))) **NOT** 門，($2^{n/2}$) **CNOT** 門，($2^{n/2}$ ×(4× m + (2 ×(n^2+n)))) **CCNOT 閘，** n *個量子位元的* ($2^{n/2}$) 相移閘和量子測量。■

引理 6 - 2 ：演算法 5-2 最壞情況的時間複雜度為(n ×(($2^{n/2}$ ×(2× n)) + (n + 1)))**哈達瑪**門，(n ×($2n^{/2}$ ×(2 ×(n^2+n)))) **非**門，(n ×$2^{n/2}$) **CNOT** 門，(n ×($2n^{/2}$ ×(4× m + (2 ×(n^2+n))))) **CCNOT** 門，(n ×$2^{n/2}$) n 個量子位元的相移閘和 (n) 個量子測量。

證明：

基於**演算法 5 -2** ，對於任何具有 m *個*邊和 n *個頂點的圖* G 求解獨立集問題，最壞的情況是在測量第 n 次執行步驟 (1a) 的結果後找到答案**演算法 5-2** 完成。也就是說，**演算法 5-2 中的步驟(1a)和步驟(1b)**執行 n *次*，**演算法 5-2** 中的步驟(1c)執行一次。因此，**演算法 5-2** 最壞情況的時間複雜度為 (n × (($2^{n/2}$ ×(2× n)) + (n + 1)))**哈達瑪**門，(n × (2n $^{/2}$ ×(2 ×(n^2+n))))**非**門，(n × $2^{n/2}$) **CNOT** 門，(n × (2n $^{/2}$ ×(4× m + (2 ×(n^2+n))))) **CCNOT** 門，(n × $2^{n/2}$) n 個量子位元的相移閘和 (n) 個量子測量。■

引理 6 -3 ：對於任何具有 m *個*邊和 n *個頂點的圖* G *，*解決獨立集問題的最壞和最好情況空間複雜度是相同的：((2× *米*+2× n + 2) + ((n ×(n + 1)) / 2)) 量子位元。

證明：

對於任何具有 m *個*邊和 n *個頂點的圖* G *，*有 $2^{n\text{個}}$可能的選擇（包括所有可能的獨立集）來解決獨立集問題，使用量子暫存器 n *個*量子位元 ($\otimes_{b=n}^{1}|y_b{}^0>$) 編碼 $2^{n\text{個}}$選擇。任何具有 m *個*邊和 n *個頂點的圖* G *的*獨立集問題是找到 G *的最大尺寸獨立集*。這可以透過使用輔助量子暫存器來實現。那些輔助量子暫存器 s 的初始狀態 s 是 (|1>) ($\otimes_{i=n}^{1}\otimes_{j=i}^{0}|w_{i,j}{}^0>$) ($\otimes_{k=m}^{1}|o_k{}^0>$) (|$o_0{}^1$>) ($\otimes_{k=m}^{1}|l_k{}^1>$)。基於**演算法 5 -2，**我們得到**演算法 5-2** 的最佳情況空間複雜度是在實現**演算法 5-1 一次**後找到答案。因此，**演算法 5-2** 的最佳情況空間複雜度涉及 ((2× *米*+2× n + 2) + ((n ×(n + 1)) / 2)) 量子位元。由於量子位元可以重複使用，最糟的情況仍然是((2× *米*+2× n + 2) + ((n ×(n + 1)) / 2)) 量子位元。因此，立即推斷**演算法 5-2 最壞情況和最好情況的空間複雜度**是相同的，並且它們都等於 ((2× *米*+2× n + 2) + ((n ×(n + 1)) / 2)) 量子位元。■

B. 求解任意具有 m 個邊和 n 個頂點的圖 G 的獨立集問題的二次加速證明

引理 6-4 ：**演算法 5-2 給出了解決任何**具有 m *個*邊和 n *個頂點的圖* G 的獨立集問題的二次加速。此加速是針對該問題*已知的最佳*加速。

證明：

貝內特等人。[8] 已經證明經典演算法的二次加速是解決任何 NP 完全問題的*已知最佳*加速。從**引理 6 -2** 可以看出，用於解決具有 m *個*邊和 n *個頂點的*任何圖 G 的獨立集問題的**演算法 5 -2 的最壞情況**與經典演算法的二次加速相符。因此，我們立即得出**演算法 5-2** 給出了二次加速，這是*已知的用於解決*具有 m *個*邊和 n *個頂點的任何圖* G 的獨立集問題的*最佳*加速。■

VII. 任何具有 M 邊和 N 頂點的圖的獨立集問題的分子解的數學表示

對於任何具有 m *個*邊和 n *個頂點的圖* G *，*求解獨立集問題的分子解的數學解是有限維希爾伯特空間中的單位向量。

引理 7 - 1：用來解任意具有 m *條*邊和 n *個頂點的*圖 G *的獨立*集問題的分子解的數學解是有限維希爾伯特空間中的單位向量。

證明：

分子演算法**求解獨立集問題**(Y_0, n, m)的步驟 (0a) 到 (1d) 中，產生由 $2^{n\text{個}DNA\text{序列編碼的}2^{n\text{個}}}$可能的選擇（獨立集） ，並由 n *編碼* Hadamard 閘在**演算法 5-1 的步驟(** 0)中對 n *個*初始量子位元進行操作。這就是說 $2^{n\text{的數學解}}$ 由 $2^{n\text{個}DNA\text{序列編碼的}}$可能選擇（獨立集）是有限維希爾伯特空間。接下來，在每次執行分子演算法中的步驟（2a）到（2d）時，**求解獨立集問題**（ Y_0, n, m ），*合法*選擇（合法獨立集）和*非法*選擇（非法獨立集）$2n^{號}$決定由 $2^{n\,DNA\text{序列編碼的}}$可能選擇。使用**演算法 5-1 中步驟 (** 1) 中的酉算符可以完成相同的任務。這表明由 $2^{n\text{個}DNA\text{序列}}$編碼的 $2^{n\text{個可能選擇中合法選擇和非法選擇的數學解}}$仍然是有限維希爾伯特空間中的單位向量。

分子演算法**求解獨立集問題**(Y_0, n, m) *中*每次執行步驟 (4a) 到 (4b)時，由 $2^{n\text{個}DNA\text{序列}}$編碼的 $2^{n\text{個選擇}}$中的合法選擇為按頂點數分類。使用**演算法 5-1 中**步驟 (2) 和步驟 (4a) 中的酉算符可以完成相同的任務。這意味著由 $2^{n\text{個}DNA\text{序列}}$編碼的

2^{n} *個選擇中分類的*那些合法選擇的數學解仍然是有限維希爾伯特空間中的單位向量。接下來，當分子演算法**求解獨立集問題**(Y_0, n, m)中每次執行步驟 s (5a) 和 (5b)時，由具有 maxi 的 DNA 序列編碼的*最大尺寸*獨立集 s 讀取 mum 個頂點，並且也是在其幅度指數放大後進行測量來讀取它們。也就是說，由最大*頂點數的 DNA 序列編碼的最大*獨立集 s 的數學解仍然是有限維希爾伯特空間中的單位向量。因此，根據上面的陳述，我們立即推導出用於解決具有 *m 個*邊和 *n 個頂點*的任何圖 *G 的獨立*集問題的分子解的數學解是有限維希爾伯特空間中的單位向量。■

VIII. 證明 NP 完全問題之間的約簡是無用的，並且每個 NP 完全問題都有自己的最佳演算法

我們假設有限變數集 $U\{u_1, u_2, \ldots, u_n\}$*上的子句*集合 $C = \{c_1, c_2, \ldots, cm\}$ ，這樣 $|c_x|$ 1 等於 $3 \le x \le$ *米*, 哪裡 $|c_x|$是第 *x 個子*句中的變數數。 3-可滿足性問題 (3-SAT) 是為了找出 *U 是否存在滿足 C* 中所有子句的真值分配。 3-SAT 問題的簡單結構使其成為其他 NP 完全結果中使用最廣泛的問題之一 [7]。庫克-**萊文**定理，也稱為**庫克**定理[50]，指出 3-可滿足性問題（3-SAT）是布林可滿足性問題之一，是 NP 完全的。也就是說，它是在 NP 中，並且 NP 中的任何問題都可以透過作為數位計算機的確定性圖靈機在多項式時間內減少到 3-可滿足性問題（3-SAT）。該定理的一個重要結論是，如果存在解決 3-可滿足性問題 (3-SAT) 的確定性多項式時間演算法，則每個 NP 問題都可以透過確定性多項式時間演算法來解決。我們使用**引理 8-1** 來證明 NP 完全問題之間的約化是*無用的*，並且每個 NP 完全問題都有自己的最佳量子演算法。**引理 8-2** 表明 **所提出的用於解決**具有 *n 個*頂點和 *m 個邊的*圖 *G 中的獨立*集問題的具有二次加速的量子分子演算法不是最好或最優的量子演算法。

引理 8-1 ：NP 完全問題之間的約簡是*無用的*，並且每個 NP 完全問題都有自己的最佳量子演算法。

證明：

我們假設 *U* 是 $\{u_1, u_2, \ldots, u_n\}$ 且 *C* 是 $\{c_1, c_2, \ldots, cm\}$ 。 *U* 和 *C* 是 3-SAT 問題的任何實例。 [7] 使用多項式時間演算法將具有 *m 個子*句和 *n 個布林變數*的 3-SAT 問題轉換為圖 *G 的*獨立集問題，其中 $(3\times n)$ 個頂點和 $((3\times m) + w)$ 邊。 *w 是* u_i 和 $\overline{u_i}$ 出現在不同子句中且 *w* 的值小於 *m 的值的對* $(u_i, \overline{u_i})$ 的數量。這表明，如果應用**演算法 5-2** 和 **演算法 5-1 來解決簡化的 3-SAT 問題，那麼**最好情況的時間複雜度為 $O(2^{(3\times n)/2})$。這意味著，對於解決簡化的 3-SAT 問題，**演算法 5-2** 和**演算法 5-1** 不能給出二次加速，並且 NP 完全問題中的簡化過程不僅不能加快量子演算法的性能，而且相反，會減慢速度。因此，從上面的陳述我們立即推斷出 NP 完全問題之間的約化是*沒有用的*，並且每個 NP 完全問題都有自己的最佳量子演算法。 ■

引理 8-2 ：所提出的用於解決具有 *n 個*頂點和 *m 個邊的*圖 *G 中的獨立*集問題的具有二次加速的量子分子演算法不是最好或最優的量子演算法。

證明：

來自**引理** 根據引理 **6-1 和引理 6-4，所提出的用於解決**具有 *n 個*頂點和 *m 個邊的圖 G* 中的獨立集問題的量子分子演算法的時間複雜度的下限和上限分別為$\Omega(2^{n\times\frac{1}{2}})$查詢和 $O(2^{n\times\frac{1}{2}})$ 查詢 $((2\times 米+2\times n + 2) + ((n\times(n + 1)) / 2))$ 量子位元。所提出的量子分子演算法滿足[8]中的重要結果，該結果表明解決任何 NP 完全問題的二次加速是一個*嚴格的*下界。然而，根據**引理 8-1 的**結論，多項式時間演算法可以將具有 *m 個子*句和 *n 個布林變數的* 3-SAT 問題轉換為圖 *G 的*獨立集合問題，其中 $(3\times n)$ 個頂點和 $((3\times m) + w)$ 邊。這就是說，NP 完全問題之間的約簡使得約簡問題的輸入規模變得比*原始*問題的輸入規模更大。這就是為什麼所提出的用於解決簡化問題的量子分子演算法無法提供任何加速的原因。看來[8]的重要結果違反了庫克定理[50]的重要結論。因此，從上面的陳述中，我們立刻推斷出，所提出的用於解決具有 *n 個*頂點和 *m 個邊的*圖 *G 中的獨立*集問題的具有二次加速比的量子分子演算法不是最好或最優的量子演算法。■

IX. 用於解決輸入 N 位的元素獨特性問題的$\Omega(\,)$查詢$\sqrt{\frac{2^N}{2}}$的量子下界證明

從[33]我們知道輸入 *n* 位的元素不同性問題是確定給定的 2^n 實數是否不同。解決此問題的量子下界是$\Omega(2^{n\times\frac{2}{3}})$ 查詢量子行走演算法 [33]。此問題的正式定義如下：給定一個函數 $\boldsymbol{H}: \{a \mid 0 \le$ *一個* $\le 2n-1\} \rightarrow \{b \mid 0 \le$ *乙* $\le 2^m - 1\}$， *r-元素獨特性*問題是找出 *r*-不同元素 $a_1, a_2, \ldots, a_r \in \{$*一個* $\mid 0 \le$ *一個* $\le 2n-1\}$ 使得 $\boldsymbol{H}(a_1) = \boldsymbol{H}(a_2) = \ldots = \boldsymbol{H}(a_r)$。Childs 和 Eisenberg 在 [51] 中擴展 r -*元素獨特性問題到一個更普遍的問題，即找到滿足任何給定屬性的大小為 r 的*子集的問題。 Childs 和 Eisenberg 在 [51] 中假設存在一個黑盒子函數 $\boldsymbol{O}_F: D \rightarrow R$ ，其中域 *D* 是有限集，範圍 *R* 也是有限集。他們進一步假設域 *D* 等於 $\{X_1, X_2, \ldots \ldots, X_N\}$ 且 $|d|$表示域 *D* 的大小，大小等於 *N*，即問題大小。他們也假設有一個集合 $(D \times R)^r = \{((X_1, \boldsymbol{OF}(X_1)), \ldots, (X_r, \boldsymbol{O}_F(X_r))) \mid Xk \in D$ 和 $\boldsymbol{O}_F(X_k) \in R\}$ 且有一個屬性 $P \subset (D \times R)^r$。 *r 子集查找問題*的更正式定義是找出一些 *r* 子集 $\{X_1, X_2, \ldots, X_r\} \subset D$ 使得 $((X_1, \boldsymbol{O}_F(X_1)), \ldots, (X_r, \boldsymbol{O}_F(X_r))) \in P$，如果不存在則拒絕。我們使用以下引理來表明解決該問題的新Ω量子下界是$(\sqrt{\frac{2^n}{2}})$ 查詢。

引理 9-1 ：為了解決輸入 *n 位的元素獨特性問題，所提出的量子分子演算法*將使用量子行走演算法的Ω量子下界 $\Omega(\,)$ 查詢改進為$2^{n\times\frac{2}{3}}(\sqrt{\frac{2^n}{2}})$ 查詢。

證明：

我們假設 $G = (V, E)$ 是一個圖，其中 *V 是 G* 中的頂點集， *E 是 G* 中的邊集。我們也假設 *V* 是 $\{v_1, \ldots, v_n\}$ 且 *E* 是 $\{(v_a, v_b) \mid v_a$和 v_b分別是 $V\}$ 中的元素。具有 *n 個*頂點和 *m 個邊的*圖 *G 的獨立集合*是子集 $V^1 \subseteq V$ 的頂點使得對所有 $v_a, v_b \in V^1$ ，邊 (v_a, v_b)不在 *E [7, 9]* 。具有 *n 個*頂點和 *m 個邊的圖 G 的*獨立集合問題是找到 *G 中最大尺*

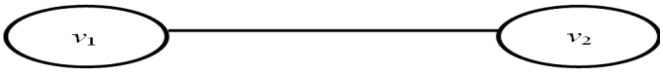
寸的獨立集合。對於具有 n 個頂點和 m 個邊的圖 G ，n 個頂點的集合的子集數為 2^n。我們假設有一個黑盒函數，$\boldsymbol{O}_F: D \to R$ 計算哪些頂點子集是具有最大頂點數的獨立集合。我們也假設域 D 等於 $\{y_n y_{n-1} \ldots y_2 y_1 | \forall 年 \in \{0, 1\}$ 為 $1 \le d \le n\}$。

我們假設 r 個 n 位長度的二進位數 X_1、X_2、……、X_r 都是 D 中的元素。我們也假設有一個集合 $(D \times R)^r = \{((X_1, \boldsymbol{OF}(X_1)), \ldots, (X_r, \boldsymbol{O}_F(X_r))) | Xk \in D$ 和 $\boldsymbol{O}_F(X_k) \in R\}$。我們假設有一個屬性 $P \subset (D \times R)^r$。我們也假設每個 n 位二進制數 X_k，對於 $1 \le k \le r$ 對頂點子集進行編碼，其中黑盒函數 $\boldsymbol{O}_F$ 可以確定最大尺寸的獨立集。這就是說，$((X_1, \boldsymbol{O}_F(X_1)), \ldots, (X_r, \boldsymbol{O}_F(X_r)))$ 是具有 n 個頂點和 m 個邊的圖 G 的獨立集問題的答案，並且 $\{X_1, X_2, \ldots, X_r\} \subset D$ 使得 $((X_1, \boldsymbol{O}_F(X_1)), \ldots, (X_r, \boldsymbol{O}_F(X_r))) \varepsilon$ P。因此，具有 n 個頂點和 m 個邊的圖 G 的獨立集問題是元素唯一性問題的一種類型，並且是 r 子集查找問題的一種類型。

從**引理 6-1** 到**引理 6-2**，為了解決具有 n 個頂點和 m 個邊的圖 G 的獨立集問題，量子下界是 $\Omega(\sqrt{\frac{2^n}{r}})$ 個查詢，量子上限是 $O(\sqrt{\frac{2^n}{r}})$ 個查詢。當 r 的值等於 2 時，量子下界為 $\Omega(2^{n\times\frac{2}{3}})$ 使用量子行走演算法進行查詢 [30]。然而，所提出的 r 值等於 2 的量子分子演算法給出了 $(\,)$ 查詢 $\sqrt{\frac{2^n}{2}}$ 的量子下界 Ω 和 $O(\,)$ 查詢的量子上限 $\sqrt{\frac{2^n}{2}}$。因此，我們立即推斷，為了解決輸入 n 位的元素獨特性問題，所提出的量子分子演算法將使用量子行走演算法的 $\Omega(\,)$ 查詢 $\sqrt{\frac{2^n}{2}}$ 的量子下界增強為 $\Omega(2^{n\times\frac{2}{3}})$ 查詢。■

X. 求二頂點一邊圖中最大獨立集的實驗結果

在圖 10-1 中，圖 G^1 由兩個頂點和一條邊組成。所有獨立組 G^1 中是 $\{v_1\}$、$\{v_2\}$ 和 $\{\}$，它是一個空集合。G^1 的最大獨立集合是 $\{v_1\}$ 和 $\{v_2\}$。我們利用圖 10-2 的量子電路和圖 10-3 的量子電路分別求出答案 $\{v_1\}$ 和答案 $\{v_2\}$。為此，我們使用 **IBM** 量子電腦中具有五個量子位元的後端 *ibmqx4* 來測試我們的理論。



我們問題的圖 G^1。

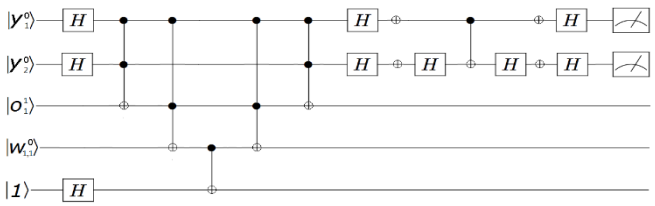


圖 10-2：找出答案 $\{v_1\}$ 的量子電路。

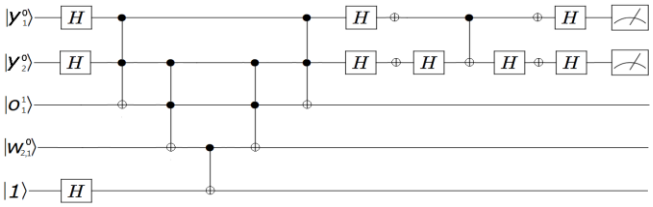


圖 10-3：找出答案 $\{v_2\}$ 的量子電路。

在 **IBM** 的 *ibmqx4 圖形介面*中，可用的閘是 **CNOT**，它是唯一具有兩個量子位元的閘，以及作用於單一量子位元的其他閘。在具有五個量子位元的後端 *ibmqx4 中*，只有六對 **CNOT** 閘。我們將 **CCNOT** 閘分解為 6 個 **CNOT** 閘和一個量子位元的閘，如圖 10-4 [30] 所示。在圖 10-4 中，H 是哈達瑪門，$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\sqrt{-1}\times\frac{\pi}{4}} \end{bmatrix}$ 和 $T^+ = \begin{bmatrix} 1 & 0 \\ 0 & e^{-1\times\sqrt{-1}\times\frac{\pi}{4}} \end{bmatrix}$.在後端 *ibmqx4 中*，我們使用量子位元 q[3]、q[4]、q[2]、q[1] 和 q[0] 分別實現量子位 $|y_1^0\rangle$, $|y_2^0\rangle$, $|o_1^1\rangle$, $|w_{1,1}^0\rangle$ 和 $|1\rangle$ 如圖 10-2 所示。同樣我們也使用量子比特 q[3]、q[4]、q[2]、q[1]和 q[0] 分別實現量子位元 $|y_1^0\rangle$, $|y_2^0\rangle$, $|o_1^1\rangle$, $|w_{2,1}^0\rangle$ 和 $|1\rangle$ 如圖 10-3 所示。由於在後端 *ibmqx4 中*，CNOT 閘無法應用於量子位元 q[3] 和 q[1]，因此圖 10-2 和圖 10-3 中的第二個和第三個 **CCNOT 閘無法由後端** *ibmqx4* 實現。因此，我們利用圖 10-5 的量子電路和圖 10-6 的量子電路分別求出答案 $\{v_1\}$ 和答案 $\{v_2\}$。

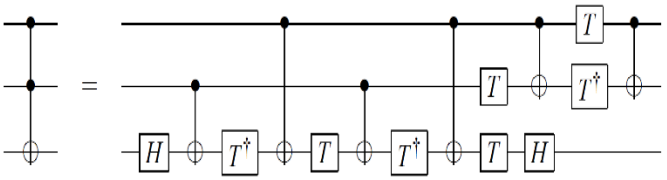


圖 10-4：將 CCNOT 閘分解為 6 個 CNOT 閘和 1 個量子位元的閘。

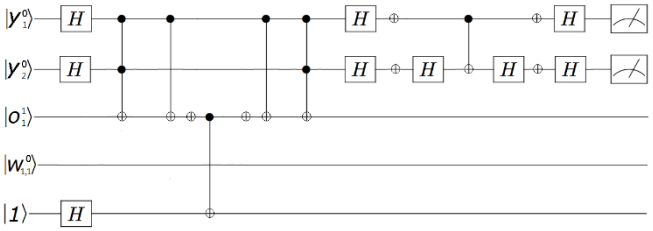


圖 10-5：用於尋找答案 $\{v_1\}$ 的量子電路，適合在後端 *ibmqx4 上執行*。

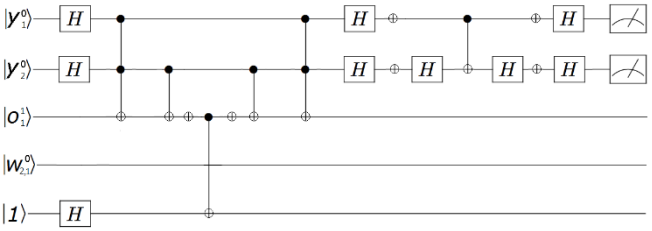


圖 10-6：用於找出答案 $\{v_2\}$ 的量子電路，適合在後端 *ibmqx4 上執行*。

我們以開放量子彙編語言 2.0 版為後端 *ibmqx4 編寫了兩個程序*，以實現圖 10-5 和 10-6 中的兩個量子電路。在第一個程式中，我們使用四個語句 - “OPENQASM 2.0;”、“include “qelib1.inc”;”、“qreg q[5];”和“creg c[5]；” - 宣告五個初始狀態為 |0> 的量子位元和五個初始值為 0 的經典位元。和 “xq[0]；”將 q[2] 和 q[0] 設定為狀態 |1>。然後，三個語句 “ hq[3];”、“hq[4];”、“hq[0];”用於完成圖 10-5 中第一個時隙中的三個 Hadamard 閘。接下來，*十五*句話—— “hq[2]；”，“cx q[4]，q[2]；”，“tdg q[2]；”，“cx q[3]，q[2]；”，“tq[2] ;”, “cx q[4],q[2];”, “tdg q[2];”, “cx q[3],q[2];”, “tq[4];”, “tq [2];”, “cx q[3],q[4];”, “hq[2];”, “tq[3];”, “tdg q[4];”和“cx q[3]，q[4]；” – 用於在圖 10-5 的*第二個時隙中完成第一個* ***CCNOT*** 閘。然後，兩個語句“cx q[3],q[2];”和“xq[2]；”實現 ***CNOT*** 門和 ***NOT*** 閘，用於標記圖 10-5 中第三個時隙到第四個時隙的答案。

接下來，在第一個程式中，語句“cx q[2],q[0];”用於在圖 10-5 中的第 5 個時隙中以(－1)來標記目標狀態的振幅。之後，用 17 條語句完成圖 10-5 中第 6 個時隙到第 8 個時隙的反轉運算。這*十七個*語句是“xq[2];”、“cx q[3],q[2];”、“cx q[3],q[4];”、“tdg q[4];”、「 tq[3]; 」, “hq[2];”, “cx q[3],q[4];”, “tq[2];”, “tq[4];”, “cx q [3]],q[2];”, “tdg q[2];”, “cx q[4],q[2];”, “tq[2];”, “cx q[3], q[2];”, “tdg q[2];”, “cx q[4],q[2];”和“hq[2];”。

接下來，*十一個*語句「hq[3];」、「hq[4];」、「xq[3];」、「xq[4];」、「hq[4];」、「cx q [3”],q[4];”, “hq[4];”, “xq[4];”, “xq[3];”, “hq[4];”和“總部[3]；”用於完成圖 10-5 中*第 9* 時隙到*第 15* 時隙答案幅度的放大。最後，兩個語句“ measure q[3] -> c[3];”和“測量 q[4] -> c[4]； ”完成圖 10-5 中*第 16 個時隙*的答案的測量。類似地，在*第二個*程式中，使用以下語句來找出答案 { v_2 }："OPENQASM 2.0;包括“qelib1.inc”； qreg q[5];克雷格 c[5]； xq[2]; xq[0];總部[3]；總部[4]；總部[0]；總部[2]； CX q[4],q[2]; tdg q[2]； CX q[3],q[2]; tq[2]; CX q[4],q[2]; tdg q[2]； CX q[3],q[2]; tq[4]； tq[2]; CX q[3],q[4]; 總部[2]； tq[3]; tdg q[4]； CX q[3],q[4]; CX q[4],q[2]; xq[2]; CX q[2],q[0]; xq[2]; CX q[4],q[2]; CX q[3],q[4]; tdg q[4]； tq[3]；總部[2]； CX q[3],q[4]; tq[2]; tq[4]; CX q[3],q[2]; tdg q[2]；CX q[4],q[2]; tq[2]; CX q[3],q[2]; tdg q[2]；CX q[4],q[2]; 總部[2]；總部[4]； xq[3]; xq[4];總部[4]； CX q[3],q[4];總部[4]； xq[4]; xq[3];總部[4]；總部[3]；測量 q[3] -> c[3]；測量 q[4] -> c[4]； 」。

圖 10-7 和圖 10-8 分別給出了兩個程式對應的電路。圖 10-7 在後端 *ibmqx4* 上，我們使用量子位元 q[3]、q[4]、q[2]、q[1] 和 q[0] 分別實現量子位 $|y_1^0\rangle$, $|y_2^0\rangle$, $|o_1^1\rangle$, $|w_{1,1}^{\ 0}\rangle$ 和 $|1\rangle$ 如圖 10-5 所示。同樣，在圖 10-8 中，在後端 *ibmqx4 上*，我們也使用量子位 q[3]、q[4]、q[2]、q[1]和 q[0]分別實現量子位 $|y_1^0\rangle$, $|y_2^0\rangle$, $|o_1^1\rangle$, $|w_{2,1}^{\ 0}\rangle$ 和 $|1\rangle$ 如圖 10-6 所示。

圖 10-7：第一個程式尋找答案{v 1 }的對應電路。

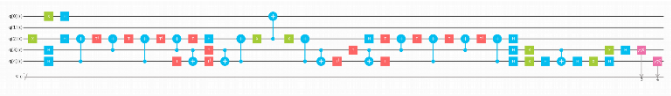
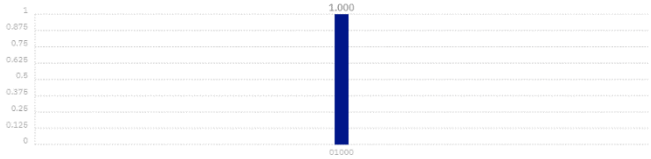


圖 10-8：第二個程式找答案{v 2 }的對應電路。

我們使用「simulate」指令在目標裝置上執行圖 10-7 和圖 10-8 中的兩個電路，即後端*模擬器*。圖 10-9 和圖 10-10 分別顯示了兩次測量結果。在圖 10-9 中，我們得到狀態 01000 的機率為 1.000。因為 q[4] 的值為 0，q[3] 的值為 1，所以我們得到第一個答案 { v_1 } 的機率為 1.000。類似地，在圖 10-10 中，我們得到狀態 10000 的機率為 1.000。 q[4] 的值為 1，q[3] 的值為 0，因此我們得到第二個答案 { v_2 } 的機率為 1.000。



在後端模擬器上求答案{v 1 }的測量結果。

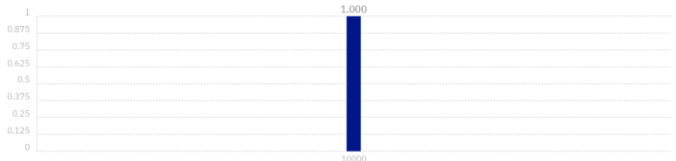


圖 10-10：在後端模擬器上求答案{v 2 }的測量結果。

我們使用「Run」指令在後端 *ibmqx4 的真實處理器上執行圖 10-7 和 10-8 中的兩個電路*。 圖 10-11 和圖 10-12 分別顯示了兩次測量結果。在圖 10-11 中，我們以 0. 541 的機率得到狀態 01000 ，或以 0. 154 的機率得到狀態 00000 ，或以 0. 112 的機率得到狀態 10000 ，或以 0. 217 的機率得到狀態 11000 。因為對於機率為 0. 541 的狀態 01000， q[4] 的值為 0，q[3] 的值為 1，所以我們得到第一個答案 { v_1 } 的機率為 0. 541。如圖 10-12 所示，我們以 0. 258 的機率得到狀態 10000 ，或以 0. 163 的機率得到狀態 00000 ，或以 0. 244 的機率得到狀態 01000 ，或以 0. 359 的機率得到狀態 11000 。儘管狀態 11000 具有較高的機率 0.359，但它編碼的集合{ v_2, v_1 }不是獨立集合。因此，我們不選擇它作為答案。對於機率為 0. 258 的狀態 10000，q[4] 的值為 1，q[3] 的值為 0，因此我們以 0. 258 的機率得到第二個答案 { v_2 }。

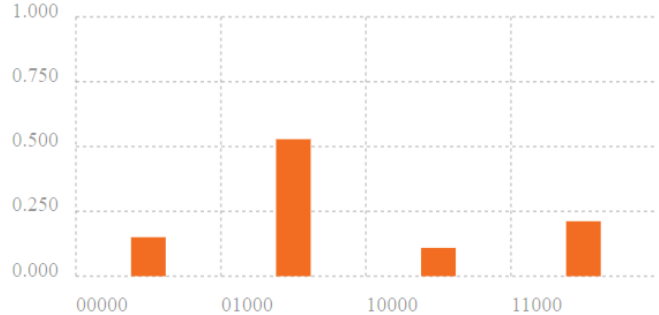


圖 10-11：在後端 ibmqx4 的真實處理器上尋找答案{v1}的

測量結果。

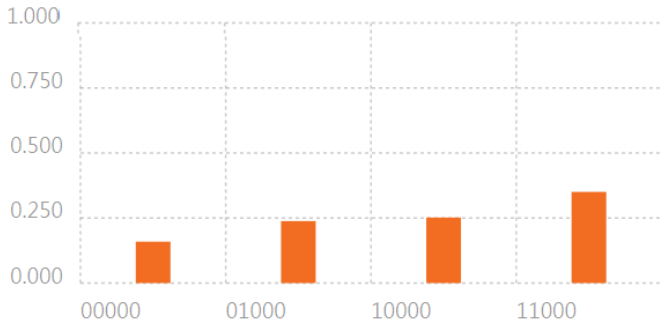


圖 10-12：在後端 ibmqx4 的真實處理器上尋找答案{v2}的測量結果。

XI. 求三頂點二邊圖中最大獨立集的實驗結果

在圖 10-13 中，圖 G^2 由三個頂點和兩條邊組成。 G^2 中的獨立集合為{ v_2, v_3 }, { v_1 }, { v_2 }, { v_3 } 和{}，這是一個空集合。 G^2 的最大獨立集合是{ v 2 , v 3 }。我們用開放量子組合語言 2.0 版寫第三個程式來找出圖 G^2 的*最大*獨立集 { v_2, v_3 }。圖 10-14 是相應的量子電路。

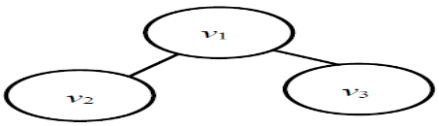


圖 10-13：具有三個頂點和兩條邊的圖 G^2。

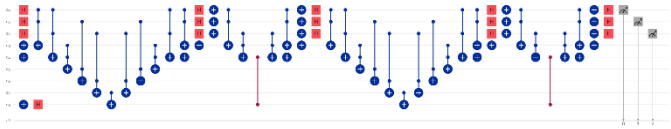


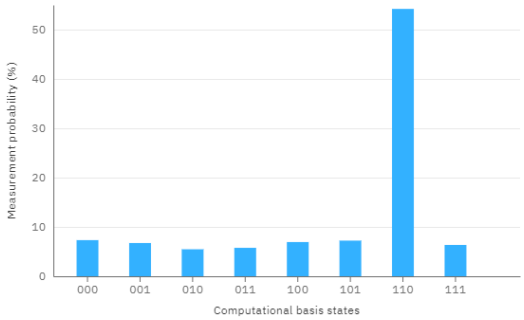
圖 10-14：第三個程式找出答案{ v_2, v_3 }對應的量子電路。

第三個程序以（－1）標示答案的幅度，並將答案的幅度放大兩倍。它指定了四個語句「OPENQASM 2.0；包括“qelib1.inc”； qreg q[9]； creg c[3];”*聲明*三個具有初始狀態 |0> 的 9 個量子位元和具有初始值 0 的經典位元。]對頂點 v_1 進行編碼。接下來，我們使用七個語句“hq[0];”總部[1]；總部[2]；xq[8];總部[8]； xq[3]； xq[4];”產生所有可能的解決方案並設定這些輔助量子位元的初始狀態。

然後，我們使用十一個語句「ccx q[0], q[1], q[3]; ccx q[0],q[2],q[4]; ccx q[3],q[4],q[5]; ccx q[1],q[5],q[6]; ccx q[2],q[6],q[7]; CX q[7],q[8]; ccx q[2],q[6],q[7]; ccx q[1],q[5],q[6]; ccx q[3],q[4],q[5]; ccx q[0],q[2],q[4]; ccx q[0],q[1],q[3];”用(－1)標記答案的幅度。接下來，我們使用語句「hq[0];總部[1]；總部[2]； xq[0]; xq[1]; xq[2]; xq[3]; xq[4]； ccx q[0],q[1],q[3]; ccx q[3],q[2],q[4]; q[4],q[8]； ccx q[3],q[2],q[4]; ccx q[0],q[1],q[3]; xq[0]; xq[1]; xq[2]; xq[3]; xq[4];總部[0]；總部[1]；總部[2]；”執行答案幅度的放大。

接下來，我們使用十一個語句「ccx q[0], q[1], q[3]; ccx q[0],q[2],q[4]; ccx q[3],q[4],q[5]; ccx q[1],q[5],q[6]; ccx q[2],q[6],q[7]; CX q[7],q[8]; ccx q[2],q[6],q[7]; ccx q[1],q[5],q[6]; ccx q[3],q[4],q[5]; ccx q[0],q[2],q[4]; ccx q[0],q[1],q[3];”用(－1)標記答案的幅度。然後，我們使用語句“hq[0];總部[1]；總部[2]； xq[0]; xq[1]; xq[2]; xq[3]; xq[4]; ccx q[0],q[1],q[3]; ccx q[3],q[2],q[4]; q[4],q[8] ； ccx q[3],q[2],q[4]; ccx q[0],q[1],q[3]; xq[0]; xq[1]; xq[2]； xq[3]; xq[4];總部[0]；總部[1]；總部[2]；”執行答案幅度的放大。最後，我們使用三個語句「measure q[0] -> c[0];測量 q[1] -> c[1]；測量 q[2] -> c[2]；”完成答案的測量。

我們使用「simulate」指令在目標裝置上執行圖 10-14 的量子電路，也就是後端*模擬器*。圖 10-15 顯示了第三個程序的測量結果。在圖 10-15 中，我們以最高機率 0.55 獲得狀態 110。因為 q[2]的值為 1，q[1]的值為 1，q[0]的值為 0，所以我們以 0.55 的機率得到答案{ v_2, v_3 }。



在後端模擬器上求答案{ v_2, v_3 }的測量結果。

XII. 結論

許多資訊處理和計算問題都可以追溯到尋找資料庫或成本函數的極端值。 Durr –Hoyer 演算法和 Ahuja –Kapoor 演算法是量子搜尋演算法的相當有用的擴展，旨在找到未排序資料庫或成本函數的最小/最大點。它表明在 [31]許多著名的用於尋找未排序資料庫或成本函數的最小/最大點的量子演算法在期望值方面有效地提供了極值；因此，無法給出所需基本步驟數量的合理上限。為了提高 Durr Hoyer 和 Ahuja –Kapoor 演算法的性能，整合了量子計數和二分搜尋的量子存在測試[31]– 被提議。任何具有 m *個*邊和 n *個頂點的圖 G 的*獨立集問題是在未排序的資料庫或具有 2^n *個*頂點子集的成本函數中找到具有最大頂點數的最大獨立集。然而，在**引理 5-6 中**，我們表明 Durr –Hoyer 演算法、Ahuja –Kapoor 演算法和量子現有測試不能解決任何具有 m 條邊和 n *個*頂點的圖 G 的獨立集問題。

引理 4-1 到**引理 4-2** 顯示任何具有 n *個*頂點和 m *個邊的圖 G* 的獨立集問題可以透過分子演算法求解**獨立集問題**(Y_0, n, m) with O($n^2 + m$)生物操作，O(2^n) DNA 鏈，O(n) 管和最長的 DNA 鏈，O(n)。**引理 5-1** 到**引理 5-5** 顯示相同的問題可以透過二次加速來解決 透過**演算法 5-2** 和**演算法 5-1**，它們實現了從分子演算法**求解獨立集問題**(Y_0, n, m) 產生的簡單布林電路。在 **引理** 從**引理 6-1** 到**引理 6-4** ，我們證明**演算法 5-2** 和**演算法 5-1 給出了**最好*的*二次加速比因處理任何具有 n *個*頂點和 m *個邊的圖 G 的獨立集問題*而*聞名的*加速。為了解決同樣的問題，已知的最佳經典演算法[52]最壞情況的時間複雜度仍然是 O(2^n)。據我們所知，

這是解決相同問題的當前可用最佳方法的替代方法。

此外，在**引理 7-1 中**，我們證明了解決相同問題的分子解的數學解是有限維希爾伯特空間中的單位向量。在 **引理 8-1**，我們證明了 NP 完全問題的約簡過程不僅無法加快量子演算法的效能，反而會減慢它的速度。因此， NP 完全問題之間的約簡是*沒有用的*，每個 NP 完全問題都有自己最好的量子演算法。這意味著使用標準約簡和所提出的量子分子演算法並不一定會產生針對 NP 問題的最佳量子演算法。此外，在**引理 8-2 中，我們表明，所提出的用於解決**具有 *n 個*頂點和 *m 個邊的*圖 *G 中的獨立*集問題的具有二次加速的量子分子演算法不是最好或最優的量子演算法。在**引理 9-1** 中，我們證明了為了解決輸入 *n 位的元素獨特性問題，所提出的量子分子演算法*將使用量子遊走演算法的 ( ) 查詢的量子下界改進為$\Omega(\sqrt{\frac{2^n}{2}})$Ω查詢$2^{n\times\frac{2}{3}}$。

致謝

中華民國國家科學基金會 MOST 105-2221-E-151-040 - 的支持。作者要感謝第四篇參考文獻 [45] 的作者 Amos 博士，感謝他為第四節標題為「*生物分子操作的介紹和實施」的 B 小節提供了寶貴的資訊*。作者也要感謝齊宇博士、彭教授、馮教授和李教授對減少計算每個合法獨立集合中頂點數量的量子電路提出的寶貴意見。

參考

[1] Feynman ， RP 的最小化。在吉爾伯特， DH 編輯。 ，萊因霍爾德出版公司，紐約，第 282-296 頁， 1961 年。

[2] L. Adleman，「組合問題解的分子計算」。*科學*，卷。 266，第 1021-1024 頁，1994 年。

[3] Feynman , R. 用電腦模擬物理。*國際理論物理學雜誌*， *21* (6/7) ，第 467-488 頁， 1982 年。

[4] 圖靈， AM 關於可計算數，及其在 Entscheidungs 問題中的應用。*倫敦數學會會刊，第* 2 - *42 (1)條*，第 230-265 頁， 1937 年。

[5] Benioff , P.不耗能圖靈機的量子力學模型。*物理評論信，48* ，第 1581-1585 頁， 1982 年。

[6] Deutsch , D. 量子理論、丘奇圖靈原理與通用量子電腦。*倫敦皇家學會會刊。A.數學和物理科學*， *400* ， 第 97-117 頁， 1985 年。

[7] Garey , MR 和 Johnson ，DS *計算機和難處理性：NP 完備性理論指南*。WH 弗里曼公司，紐約，1979 年。

[8] Bennett, CH、Bernstein, E.、Brassard, G. 和 Vazirani，量子計算的 UV 優點和缺點。 *SIAM 計算雜誌*， *26* (5) ，第 1510-1523 頁，1997 年。

[9] Karp , R.關於組合問題的計算複雜度。*網路, 5* ，第 45-68 頁，1975 年。

[10] Boneh , D.、Dunworth , C. 和 Lipton , RJ 使用分子電腦破解 DES。第*一屆DIMACS 基於 DNA 的電腦研討會*的*會議記錄*（普林斯頓大學），美國數學會。摘自《離散數學與理論計算機科學 DIMACS 系列》，第 27 卷，第 37-66 頁， 1996 年。

[11] 阿德曼， L.M . ，羅瑟蒙德， P 。 W.、Roweis , S.和 Winfree , E. 將分子計算應用於資料加密標準。*第二屆DNA 計算年度研討會*（普林斯頓大學），美國數學會。在《離散數學與理論計算機科學DIMACS系列》中，第 31-44 頁， 1999 年。

[12]張文， -L 。 ,何明和郭， M 。 用於基於 DNA 的計算的快速平行分子演算法： 因式分解整數。 *IEEE 奈米生物科學彙刊*，4 (2) ，第 149-163 頁， 2005 年。

[13] 張， W.-L 。 基於 DNA 的快速平行分子計算演算法：二次同餘和整數因式分解。 *IEEE 會刊奈米生物科學*，第 11 卷，第 1 期，第 62-69 頁，2012 年。

[14] Lipton , R. 硬計算問題的 DNA 解決方案。 *《科學》* ，268 ，第 542-545 頁， 1995 年。

[15] Adleman ，LM 論建構分子電腦。在 Lipton , R. 和 Baum, E.編。 *以 DNA 為基礎的計算機*，美國數學會。摘自《離散數學與理論計算機科學 DIMACS 系列》 ，第 1-21 頁， 1996 年。

[16]葉志偉，朱志鵬。和吳， K.-R 。基於 DNA 計算的二元整數規劃問題的分子解。*生物系統*， *83* (1) ，第 56-66 頁， 2006 年。

[17] Ho, M. 基於 DNA 的超級計算的快速並行分子解決方案：子集積問題。*生物系統*，80 (3) ，第 233–250 頁， 2005 年。

[18] 漢高， C . V. ，貝克， T 。 ，郭， J 。 N. ，羅森伯格， G 。和斯班克，H . P.背包問題解的 DNA 計算。*生物系統*，88 (1-2) ，第 156-162 頁，2007 年。

[19]張， W.-L 。基於快速平行 DNA 的分子計算演算法：集合劃分問題。*IEEE 奈米生物科學彙刊*， 第 6 卷，第 1 期，第 346-353 頁， 2007 年。

[20] 葉C.和Chu, C. 基於 DNA 計算的基於規則的系統的分子驗證。 *IEEE 知識與資料工程彙刊*， *20* (7) ，第 965-975 頁， 2008 年。

[21] 張， W.-L 。和 Vasilakos ，在生物計算機上實現生物分子資料庫的 AV DNA 演算法。 *IEEE 奈米生物科學彙刊*，第 14 卷，第 1 期，第 104-111 頁， 2015 年。

[22] Chang ， W.-L.，Vasilakos ， AV 和 Ho ， MS-H。 在生物計算機上實現複雜向量算術運算的基於 DNA 的演算法。*IEEE 奈米生物科學彙刊*，第 14 卷，第 8 期，第 1-8 頁， 2015 年。

[23] Woods D.、Doty D.、Myhrvold C.、Hui J.、Zhou F.、Yin P. 和 Winfree E. 使用可重編程 DNA 自組裝的多樣化且穩健的分子演算法。*《自然》*，第 567 *卷*，第 366-372 頁，2019 年 3 月 21 日。

[24] 任 XM，王 XM，王 ZC 和吳 TH 基於廣義旅行商問題的仿生計算模型的平行 DNA 演算法。 *《國際計算智慧系統期刊》* ，第 14 卷第 1 期，第 228-237 頁，2021 年。

[25] 徐傑等。基於探針圖的圖頂點著色問題的 DNA 計算模型。*工程學*，卷。 4、沒有。 1，2018 年，第 61-77 頁。

[26] 王 ZC 等。一種解決基於 Adleman-Lipton 模型的有能力車輛路徑問題的新型生物啟發式計算演算法。*生物系統*，卷。 184，2019 年，第 184 頁。 103997。

[27] Deutsch , D. and Jozsa , R. 透過量子運算快速解決問題。*倫敦皇家學會會刊。A.數學與物理科學*，439，第 553-558 頁， 1992 年。

[28]肖爾， PW 量子計算演算法：離散對數和因式分解演算法。第 35 屆 IEEE 電腦科學*基礎研討會論文集*（美國新墨西哥州聖塔菲），第 124-134 頁， 1994 年。

[29] Grover, LK用於資料庫搜尋的快速量子機械演算法。第二十八屆 ACM 計算*理論年度研討會論文集*（美國賓州費城），AC M， 第 212-219 頁，1996 年。

[30] Nielsen ， MA 和 Cchang ， IL *量子計算和量子資訊*。劍橋大學出版社，紐約，紐約， 2000 年。

[31] Imre , S. and Balazs , F.*量子計算和通訊*： *工程方法*。約翰威利父子有限公司，英國， 2007 年。

[32] Lipton RJ 和 Regan KW *線性代數數子演算法：入門*。麻省理工學院出版社，2014 年， **ISBN** 978-0-262-02839-4，2014 年。

[33] Aaronson , S. and Shi, Y. 碰撞與元素獨特性問題的量子下界。 *ACM 雜誌*，51：第 595-605 頁，2004 年。

[34]霍文.是的。和朗， G 。 L。固態電路中的糾纏與擠壓。 *新物理學雜誌*，10，第 1-11 頁， 2008 年。

[35] 楊文林，魏華，陳成雲。和 Feng, M.使用捕獲的超冷離子實現多量子位 Grover 搜尋。*雜誌 美國光學學會 B* ， 25 (10)，第 1720-1727 頁， 2008 年。

[36] Long , GL 和 Xiao, L. 7-qubit NMR Liouville 空間計算機中獲取演算法的實驗實現。 *化學物理雜誌，119 ，第* 8473-8481 頁， 2003 年。

[37] Boneh , D. 和 Lipton, RJ 隱藏線性函數的量子密碼分析。在 *CRYPTO '95* ，電腦科學講義，Springer-Verlag ，第 424-437 頁， 1995 年。

[38] Lukac, M. and Perkowski , M.量子符號邏輯綜合的演化方法。 2008 年 IEEE 進化計算大會，2008 年 IEEE 世界計算智能大會，（中國香港），IEEE，第 3374-3380 頁，2008 年。

[39] DJ Moylett 等人。有界度圖旅行商問題的量子加速。*物理評論A*，卷。95，沒有。 2017 年 3 月，第 14 頁。 32323。

[40] Chang，W.-L.，等。解決最大集團問題的量子加速。*物理評論A*，卷。97，沒有。 2018年第3頁， 32344。

[41] Pelofske，E.，等。解決量子退火器上的大型最小頂點覆蓋問題。*第16屆 ACM 國際計算前沿會議論文集*，2019 年，第 76-84 頁。

[42] Arute F.、Arya K.、Babbush R.、Bacon D.、Bardin JC、Barends R.、Biswas R.、Boixo S.、Fernando G.、Brandao SL、Buell DA、Burkett B.、Chen Y ., Chen Z., Chiaro B., Collins R., Courtney W., Dunsworth A., Farhi E., Foxen B., Fowler A., Gidney C., Giustina M., Graff R., Guerin K ., Habegger S.、Harrigan MP、Hartmann MJ、Ho A.、Hoffmann M.、Huang T.、Humble TS、Isakov SV、Jeffrey E.、Jiang Z.、Kafri D.、Kechedzhi K.、Kelly J.、Klimov PV 、Knysh S.、Korotkov A.、Kostritsa F.、Landhuis D.、Lindmark M.、Lucero E.、Lyakh D.、Mandrà S.、McClean JR、McEwen M.、Megrant A.、Mi X、Michielsen K.、Mohseni M.、Mutus J.、Naaman O.、Neeley M.、Neill C.、Niu MY、Ostby E.、Petukhov A.、Platt JC、Quintana C.、Rieffe EG、Roushan P.、Rubin NC 、Sank D.、Satzinger KJ、Smelyanskiy V.、Sung KJ、Trevithick MD、Vainsencher A.、Villalonga B.、White T.、Yao ZJ、Yeh P.、Zalcman A.、Neven H. 和Martinis JM 使用可程式設計的量子霸權超導處理器。 *《自然》，第*574 卷，第 505-510 頁，2019 年 10 月 23 日。

[43] Silva V.*針對開發人員的實用量子運算：使用 Python、量子彙編語言和 IBM Q Experience 在雲端中對量子設備進行程式設計*。 Apress，2018 年 12 月 13 日，ISBN-10：1484242173 和 ISBN-13：978-1484242179，2018。

[44] Johnston, ER、Harrigan N. 和 Gimeno-Segovia M.*量子電腦程式設計：基本演算法和程式碼範例*。奧萊利媒體公司，ISBN- 13：978-1492039686，ISBN- 10：1492039683，2019 。

[45] Amos, M.*理論與實驗 DNA 計算*。施普林格， ISBN - 13：978-3540657736 ， ISBN - 10：3540657738，2006年 4 月。

[46] 張， W.-L 。和 Vasilakos ， AV *分子計算：面向複雜問題解決的新型計算架構*，Springer，ISBN-13：978-3319051215，ISBN-10：3319051210，2014 年 6 月，

[47] Boyer, M.、Brassard , G.、Hoeyer , P. 和 Tapp, A. 量子搜尋的嚴格界限。*福爾奇。 《物理學》* ，46，第 493-506 頁， 1998 年。

[48] Durr C. 和 Hoyer P. 尋找最小值的量子演算法。 arXiv: quant-ph/9607014, 1996

[49] Ahuja A. 和 Kapoor S. 尋找最大值的量子演算法。 arXiv：Quant-ph/9911082，1999。

[50] Cook, S. 定理證明程序的複雜性。第三屆 ACM 計算理論年度研討會論文集，第 151-158 頁， 1971 年。

[51] Childs AM 和 Eisenberg JM 用於子集查找的量子演算法。 *《量子資訊與計算期刊》* ，第 5 卷第 7 期，第 593-604 頁，2005 年。

[52]Xiao M.和 Nagamochi, H.最大獨立集的精確演算法。*資訊與計算*，第 255 卷，第 126–146 頁，2017 年。