

آلگوریتم های ساده کوانتومی برای محاسبه

وحیدکریمی پور- دانشکده فیزیک - دانشگاه صنعتی شریف

۱۲ آذر ۱۳۹۳

۱ مقدمه

کامپیوتر کوانتومی چیست؟ چگونه می توان از مکانیک کوانتومی برای محاسبه استفاده کرد؟ آیا یک کامپیوتر کوانتومی قدرتی فراتر از کامپیوترهای کلاسیک دارد؟ اگر چنین است، کدام خصلت های مکانیک کوانتومی این توانایی را بوجود می آورند؟ هدف ما در این درس فراهم آوردن پاسخ هایی جزئی برای این سوال هاست. نخست به سوال اول می پردازیم. می دانیم که در کامپیوترهای کلاسیک هر نوع داده ای در بیت های کلاسیک ذخیره می شود. یک بیت کلاسیک صرف نظر از نوع سخت افزاری که برای آن به کار رفته است، یک سیستم فیزیکی است که می تواند در دو حالت 0 یا 1 قرار گیرد. از اصطلاح حافظه کلاسیک^۱ برای مجموعه ای از بیت ها که می تواند داده های گسترده تری را در خود ذخیره کند. این که حافظه معمولاً به صورت دسته های هشت تایی به نام بایت یا بزرگتر از آن منظم شده است و یا این که نوع آن چگونه است (حافظه کوتاه مدت *Random Access Memory (RAM)* یا دائمی) فعلاً مورد توجه ما نیست. هر نوع پردازشی در نهایت به این معناست که ما بتوانیم توابع دلخواهی را برای داده هایی که در این حافظه ها ذخیره شده اند محاسبه کنیم. در درس گذشته دیدیم که چگونه می توان با استفاده از مدارهای منطقی کلاسیک، چه از نوع برگشت پذیر چه از نوع بازگشت ناپذیر آن، داده هایی را که در این

^۱ Classical Register

حافظه‌ها ذخیره می شود پردازش کرد.

مفهوم مرکزی در کامپیوتر کوانتومی بیت کوانتومی یا کیوبیت^۲ است. یک کیوبیت به یک سیستم کوانتومی گفته می شود که فضای هیلبرت آن دو بعدی است. این فضای هیلبرت را با C^2 نشان می دهیم.

$$C^2 := \left\{ \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \mid \alpha, \beta \in C \right\}. \quad (۱)$$

معمول است که بردارهای پایه‌ی این فضا را با $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ و $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ نشان می دهند. یک کیوبیت بر خلاف بیت کلاسیک می تواند در ترکیبی از حالت های پایه فوق قرارگیرد. حالت یک کیوبیت با برداری در این فضای هیلبرت نشان داده می شود:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \quad (۲)$$

یک حافظه کوانتومی n کیوبیتی^۳ عبارت است از مجموعه‌ای متشکل از n کیوبیت. فضای هیلبرت این حافظه عبارت است از $(C^2)^{\otimes n}$:

$$(C^2)^{\otimes n} = \left\{ \sum_{s_0, s_1, \dots, s_{n-1} \in \{0,1\}} \alpha_{s_0, s_1, \dots, s_{n-1}} |s_0, s_1, \dots, s_{n-1}\rangle \right\}, \quad (۳)$$

و بعد از آن برابر است با 2^n . دقت کنید که این حافظه می بایست چنان باشد که تمام بردارهای فضای هیلبرت آن قابل دسترسی باشند. بنابراین اگر دو کیوبیت را تنها کنارهم بگذاریم به این معنی نیست که یک حافظه دو کیوبیتی ساخته‌ایم. زیرا بردارهای فضای هیلبرت دو کیوبیت جداگانه به صورت زیر هستند:

$$|\phi\rangle \otimes |\phi'\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha'|0\rangle + \beta'|1\rangle), \quad (۴)$$

^۲Qubit

^۳n-qubit Quantum Register

یعنی به صورت ضرب تانسوری دو بردار از فضاهاى هرکدام از کیوبیت ها نوشته می شوند و حال آنکه در فضای هیلبرت دو کیوبیت یعنی $(C^2)^{\otimes 2}$ بردارهایی وجود دارند که به صورت فوق قابل نوشتن نیستند، مثل بردار $|\phi\rangle = \frac{1}{2}(|00\rangle + |11\rangle)$. این نوع بردارها را اصطلاحاً بردارهای درهم تنیده می نامیم. برای آنکه یک بردار کلی به صورت

$$a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle \quad (5)$$

به صورت حاصل ضرب تانسوری دو بردار نوشته شود می بایست شرط $ad - bc = 0$ برقرار شود. بنابراین بردارهایی که به صورت ضرب تانسوری هستند مجموعه‌ای با اندازه صفر را در فضای هیلبرت دو کیوبیت تشکیل می دهند.

توازی کوانتومی: اولین تفاوت مهم کامپیوتر کوانتومی با کامپیوتر کلاسیک این است که یک حافظه کوانتومی می تواند در آن واحد در تمام حالت های بالقوه خود قرار بگیرد. این خصلت ناشی از برهم نهی حالت های کوانتومی است و اصطلاحاً توازی کوانتومی^۴ خوانده می شود. هرگاه هر حالت $|s_0, s_1, \dots, s_{n-1}\rangle$ را برای کد کردن عدد دوتایی $s = (s_0, s_1, \dots, s_n)$ به کار ببریم حالتی مثل

$$|\phi\rangle = \sum_{s=0}^{2^n-1} \phi_s |s\rangle, \quad (6)$$

حالت کلی یک حافظه n بیتی است. بنابراین هرگاه حافظه را در حالت $|\psi\rangle$ قرار دهیم مثل این است که همزمان آن را در تمام حالت های $|s\rangle$ قرار داده‌ایم. البته واضح است که هرگاه حافظه را در پایه‌ی محاسباتی یعنی همان پایه 0 ها و 1 ها اندازه گیری کنیم تنها یکی از مقادیر s با احتمال $|\phi_s|^2$ بدست خواهند آمد. بنابراین توازی کوانتومی اگر چه یک خاصیت مهم حافظه است ولی این خاصیت را می بایست با ظرافت مورد بهره برداری قرار داد.

گیت کوانتومی: اگر اطلاعات را در کیوبیت ها یعنی حالت های کوانتومی ذخیره کنیم، به ناچار پردازش اطلاعات می بایست یا با عملگرهای یکانی که تحول را نشان می دهند و یا با اندازه گیری ها انجام بگیرند. معمولاً اصطلاح گیت کوانتومی برای عملگر یکانی بکار برده می شود. گیت کوانتومی هرگاه روی یک کیوبیت اثر کند آن را گیت تک کیوبیتی^۵ و هرگاه روی n تا کیوبیت اثر کند، گیت n کیوبیتی^۶ خوانده می شود. در درس های آینده راجع به گیت های چند کیوبیتی و طرز ساختن آنها به طور کلی به تفصیل بحث خواهیم کرد.

^۴Quantum Parallellism

^۵One qubit gate

^۶n-Qubit Gate

آلگوریتم کوانتومی: آلگوریتم کوانتومی در ساده ترین شکل آن به مجموعه‌ای از گیت های کوانتومی متوالی گفته می شود که روی یک حالت معین اولیه اثر می کنند و چنان تنظیم شده‌اند که حالت نهایی چنان باشد که پس از اندازه گیری های سنجیده‌ای روی آن جواب یک مسئله معین را با احتمال بسیار خوب در بر داشته باشد.

هدف ما در این درس این است که با چند آلگوریتم ساده کوانتومی آشنا شویم و ببینیم که چگونه کامپیوترهای کوانتومی و آلگوریتم های کوانتومی می توانند بعضی از مسائل را سریعتر از کامپیوترها و آلگوریتم های کلاسیک حل کنند.

۲ جعبه سیاه یا اوراکل

فرض کنید که جعبه سیاهی تابع f را به روی ورودی خود اعمال می کند. ما تنها می توانیم به این جعبه سیاه ورودی های مختلف بدهیم و خروجی های آن را ثبت کنیم. به این کار فراخوانی تابع می گویند. در اصطلاح علم کامپیوتر به چنین جعبه سیاهی یک اوراکل (*Oracle*) می گویند. در ساده ترین مثال فرض کنید که این تابع یک تابع یک بیتی است، یعنی $f: B_1 \rightarrow B_1$. ولی در حالت های کلی تر این تابع می تواند از n بیت به یک بیت یا از n بیت به m بیت باشد. اصطلاح جعبه سیاه را به این دلیل به کار می بریم که بسیاری از مسائل رایانش را می توان با استفاده از آن صورت بندی کرد. در این نوع مسائل هدف همواره این است که با دادن ورودی های مختلف به این جعبه سیاه یا تابع و مشاهده خروجی های آن بتوانیم خاصیتی از تابع را پیدا کنیم. در این نوع مسائل معمولاً نمی خواهیم نوع تابع را به طور کامل بفهمیم بلکه فقط یک خاصیت مشخص از تابع مورد نظر است. در زیر چند نوع از این مسائل را طرح می کنیم و نشان می دهیم که آلگوریتم های کوانتومی می توانند بهتر از آلگوریتم های کلاسیک این مسائل را حل کنند. این مسائل به ترتیب از حالت خیلی ساده شروع شده و سپس به حالت های پیچیده تر ختم می شوند.

می خواهیم ببینیم چند بار این تابع را فراخوانی کنیم، تا بتوانیم تابع f را تعیین کنیم. می دانیم که این تابع یکی از چهار تابعی است که در جدول 3 نشان داده شده است. براحتی معلوم می شود برای تعیین تابع f می بایست دوبار تابع فراخوانی شود، یک بار با ورودی 0 و بار دیگر با ورودی 1 تا بتوان نوع تابع را تشخیص داد. می توانیم مسئله را ساده تر کنیم و بگوییم که دقیقاً نوع تابع را نمی خواهیم بلکه می خواهیم بفهمیم که آیا این تابع یک تابع ثابت است و یا یک تابع متوازن؟ منظور از

تابع ثابت، تابعی است که خروجی اش همواره یک مقدار ثابت و مستقل از ورودی است (مثل تابع f_1 یا f_4) و منظور از تابع متوازن، تابعی است که خروجی اش به ازای نیمی از ورودی ها برابر با 0 و به ازای نیمی دیگر برابر با 1 است (مثل تابع f_2 و f_3). حال می پرسیم که با چند بار فراخوانی این تابع می توان به ثابت بودن یا متوازن بودن آن پی برد. بازهم کمی فکر کردن نشان می دهد که دو بار می بایست تابع فراخوانده شود.

برای اولین بار دویچ نشان داد که این مسئله را می توان با الگوریتم کوانتومی تنها با یک بار فراخوانی حل کرد. این کار دویچ سرآغاز پژوهش های جدی در باره الگوریتم های کوانتومی بوده است. نکته اصلی که در این الگوریتم و همه الگوریتم های کوانتومی وجود دارد استفاده هوشمندانه از خاصیت برهم نهی حالت ها ست. هم چنین چیزی که در بسیاری از این الگوریتم ها مشترک است استفاده از این نکته است که جعبه سیاه به دلیل آنکه تابع را به صورت برگشت پذیر محاسبه می کند دارای دو ورودی و دو خروجی به ترتیب زیر است:

$$|x, y\rangle \longrightarrow |x, y \oplus f(x)\rangle, \quad (7)$$

۳ الگوریتم کوانتومی برای مسئله دویچ

فرض کنید که جعبه سیاهی تابع f را به روی ورودی خود اعمال می کند. ما تنها می توانیم به این جعبه سیاه ورودی های مختلف بدهیم و خروجی های آن را ثبت کنیم. به این کار فراخوانی تابع می گویند. در اصطلاح علم کامپیوتر به چنین جعبه سیاهی یک اوراکل (*Oracle*) می گویند. در ساده ترین مثال فرض کنید که این تابع یک تابع یک بیتی است، یعنی $f: B_1 \rightarrow B_1$. می خواهیم ببینیم چند بار این تابع را فراخوانی کنیم، تا بتوانیم تابع f را تعیین کنیم. می دانیم که این تابع یکی از چهار تابعی است که در جدول 3 نشان داده شده است. براحتی معلوم می شود برای تعیین تابع f می بایست دوبار تابع فراخوانی شود، یک بار با ورودی 0 و بار دیگر با ورودی 1 تا بتوان نوع تابع را تشخیص داد. می توانیم مسئله را ساده تر کنیم و بگوییم که دقیقاً نوع تابع را نمی خواهیم بلکه می خواهیم بفهمیم که آیا این تابع یک تابع ثابت است و یا یک تابع متوازن؟ منظور از تابع ثابت، تابعی است که خروجی اش همواره یک مقدار ثابت و مستقل از ورودی است (مثل تابع f_1 یا f_4) و منظور از تابع متوازن، تابعی است که خروجی اش به ازای نیمی از ورودی ها برابر با 0 و به ازای نیمی دیگر برابر با 1 است (مثل تابع

f_2 و f_3). حال می پرسیم که با چند بار فراخوانی این تابع می توان به ثابت بودن یا متوازن بودن آن پی برد. بازهم کمی فکر کردن نشان می دهد که دو بار می بایست تابع فراخوانده شود.

برای اولین بار دویچ نشان داد که این مسئله را می توان با الگوریتم کوانتومی تنها با یک بار فراخوانی حل کرد. این کار دویچ سرآغاز پژوهش های جدی در باره الگوریتم های کوانتومی بوده است. نکته اصلی که در این الگوریتم و همه الگوریتم های کوانتومی وجود دارد استفاده هوشمندانه از خاصیت برهم نهی حالت ها است. هم چنین چیزی که در بسیاری از این الگوریتم ها مشترک است استفاده از این نکته است که جعبه سیاه به دلیل آنکه تابع را به صورت برگشت پذیر محاسبه می کند دارای دو ورودی و دو خروجی به ترتیب زیر است:

$$|x, y\rangle \longrightarrow |x, y \oplus f(x)\rangle, \quad (8)$$

که در نتیجه آن

$$|x, -\rangle \equiv \frac{1}{\sqrt{2}}(|x, 0\rangle - |x, 1\rangle) \longrightarrow \frac{1}{\sqrt{2}}(|x, f(x)\rangle - |x, f(x) + 1\rangle) \equiv (-1)^{f(x)}|x\rangle|-\rangle. \quad (9)$$

بنابراین وقتی که ورودی دوم را در حالت $|-\rangle$ نگه می داریم آنچه که در ورودی اول ظاهر می شود فازی است که توسط مقدار تابع $f(x)$ تعیین می شود بدون اینکه در ورودی دوم هیچ تغییری ایجاد شود. حال اگر ورودی اول را نیز در یک حالت برهم نهی یکنواخت تهیه کنیم آنگاه حالت خروجی نیز یک برهم نهی خواهد بود به این ترتیب که:

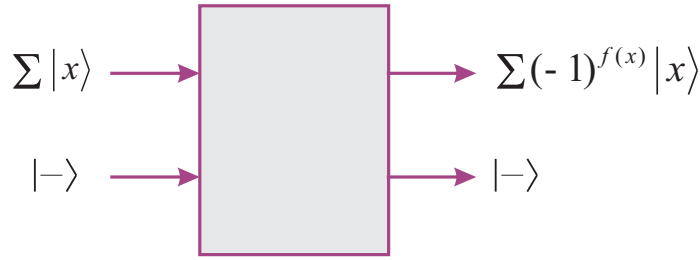
$$\sum_x |x, -\rangle \longrightarrow \sum_x (-1)^{f(x)}|x\rangle|-\rangle. \quad (10)$$

توجه به این نکته کلید درک بسیاری از الگوریتم هایی است که در پی می آید. به طور خلاصه می توان این عمل را در دیگرام زیر نشان داد:

این نکته را یادآوری می کنیم که برای تهیه یک برهم نهی یکنواخت از تمامی حالت ها می بایست از عملگرهای هادامارد استفاده کنیم.

■ تمرین: نشان دهید که

$$H^{\otimes n}|\mathbf{x}\rangle = \frac{1}{\sqrt{N}} \sum_{\mathbf{y}=0}^{N-1} (-1)^{\mathbf{x} \cdot \mathbf{y}} |\mathbf{y}\rangle, \quad (11)$$



شکل ۱: استفاده از توازی کوانتومی در الگوریتم های کوانتومی

که در آن $N = 2^n$ است.

الگوریتم کوانتومی دویچ را با مدارش توضیح می دهیم. از روی مدار شکل ؟؟ می توان فهمید که در مراحل متوالی بردار حالت ورودی چگونه تحول می یابد: داریم

$$|\Psi_0\rangle = |s\rangle \otimes |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{2}(|00\rangle + |10\rangle - |01\rangle - |11\rangle)$$

$$|\Psi_1\rangle = \frac{1}{2}(|0, f(0)\rangle + |1, \overline{f(1)}\rangle - |0, \overline{f(0)}\rangle - |1, \overline{f(1)}\rangle) \quad (12)$$

حال دقت می کنیم که اگر تابع f ثابت باشد دراین صورت $f(0) = f(1) = f$ و $\overline{f(0)} = \overline{f(1)} = \overline{f}$. درنتیجه خواهیم داشت

$$|\Psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|f\rangle - |\overline{f}\rangle) \quad (13)$$

و

$$|\Psi_3\rangle = |0\rangle \otimes \frac{1}{\sqrt{2}}(|f\rangle - |\overline{f}\rangle). \quad (14)$$

اما اگر تابع f متوازن باشد دراین صورت $f(0) = \overline{f(1)} = f$ و $\overline{f(0)} = f(1) = \overline{f}$. درنتیجه خواهیم داشت

$$|\Psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \otimes \frac{1}{\sqrt{2}}(|f\rangle - |\overline{f}\rangle), \quad (15)$$

f_4		f_3		f_2		f_1	
۱	0	۱	0	۰	0	۰	0
۱	1	۰	1	۱	1	۰	1

جدول ۱: انواع توابع یک بیتی

$$|\Psi_3\rangle = |1\rangle \frac{1}{\sqrt{2}}(|f\rangle - |\bar{f}\rangle). \quad (۱۶)$$

بنابراین وقتی که کیوبیت بالا را در مدار در (پایه محاسباتی) اندازه گیری کنیم، برای تابع ثابت مقدار 0 و برای تابع متوازن مقدار 1 را بدست می آوریم. به این ترتیب با یک بار خواندن تابع می توانیم به ثابت بودن و یا متوازن بودن تابع پی برد.

■ تمرین: فرض کنید که به جای کیوبیت ها از کیوتریت ها استفاده می کنیم. کلی ترین حالت یک کیوتریت به صورت $|\psi\rangle = a|0\rangle + b|1\rangle + c|2\rangle$ است. مسئله دویچ را به این صورت تعمیم می دهیم. یک جعبه سیاه تابع زیر را محاسبه می کند:

$$f : \{0, 1, 2\} \longrightarrow \{0, 1, 2\}. \quad (۱۷)$$

گفته شده است که این تابع یا متوازن است یا ثابت. در حالت کلاسیک چند بار می بایست تابع را فراخوانی کنیم تا بفهمیم که آیا تابع ثابت است یا متوازن؟

هرگاه بجای کیوبیت ها کیوتریت (سیستم های اسپینی سه حالت) در اختیار داشته باشید یک الگوریتم کوانتومی طراحی کنید که بتواند این مسئله را موثرتر از مسئله کلاسیک حل کند.

راهنمایی: در این تمرین و در تمرین های بعدی از عملگر هادامارد با تعریف

$$H = \frac{1}{\sqrt{d}} \sum_{i,j} \omega^{ij} |i\rangle \langle j| \quad (۱۸)$$

استفاده کنید.

■ تمرین: مسئله دویچ را به این صورت تعمیم می دهیم. یک جعبه سیاه تابع زیر را محاسبه می کند:

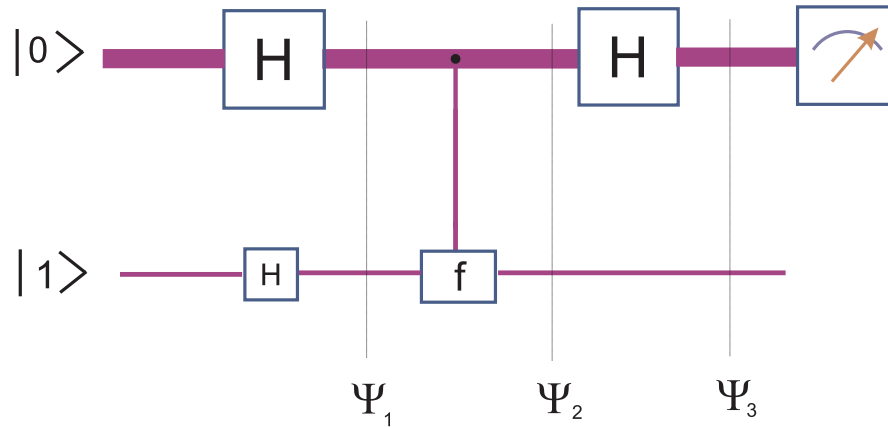
$$f : \{0, 1, 2, \dots, d-1\} \longrightarrow \{0, 1, 2, \dots, d-1\}. \quad (۱۹)$$

گفته شده است که این تابع یا متوازن است یا ثابت. در حالت کلاسیک چند بار می بایست تابع را فراخوانی کنیم تا بفهمیم که آیا تابع ثابت است یا متوازن؟

هرگاه بجای کیویت ها کیودیت (سیستم های اسپینی d حالت) در اختیار داشته باشید یک آگوریتم کوانتومی طراحی کنید که بتواند این مسئله را موثر تر از مسئله کلاسیک حل کند.

۴ آگوریتم کوانتومی برای مسئله دویچ - جوزا

مسئله دویچ جوزا (Deutsch - Jozsa) تعمیمی است از مسئله دویچ. فرض کنید که تابع $f : B_n \longrightarrow B_1$ یک تابع n بیتی است و به ما گفته شده است که این تابع متوازن یا ثابت است یعنی مقادیر خروجی آن همه یا با هم مساوی هستند و یا نیمی از مقادیر خروجی اش صفر و نیمی صفر هستند. می دانیم که تعداد توابع ثابت برابر است با 2 . اما تعداد توابع متوازن به صورت نمایی زیاد است و این تفاوت مهم این مسئله با مسئله دویچ است. برای آنکه تعداد توابع متوازن را بشماریم دقت می کنیم که تعداد ورودی های مختلف این تابع و در نتیجه خروجی های آن برابر است با 2^n . تعداد توابع متوازن برابر است با تعداد طرعی که می توان نیمی از 2^n خروجی را برابر با صفر و نیمی دیگر را برابر با 1 گرفت و این تعداد برابر است با $\frac{2^n!}{(2^{n-1})^2}$ و این عددی است که رابطه اش با n یک رابطه نمایی است. بنابراین تعداد کل توابع ممکن که می توانند ثابت یا متوازن باشند برابر است با $\frac{2^n!}{(2^{n-1})^2} + 2$. سوال این است که چند بار می بایست تابع را فراخوانیم تا بتوانیم با اطمینان بگوییم که تابع ثابت است یا متوازن. اگر بخواهیم یک آگوریتم تعینی (یعنی آگوریتمی که با قطعیت پاسخ را مشخص کند) برای این مسئله به کار ببریم باید به



شکل ۲: مداری که برای الگوریتم دویچ - جوزا به کار می رود.

ترتیب زیر عمل کنیم. اعداد n بیتی $x := (x_{n-1}x_{n-2} \cdots x_1x_0)$ را یک به یک می دهیم (تابع را فرامی خوانیم تا $f(x)$ را محاسبه کند).

هرگاه به ازای دو مقدار مختلف x مقادیر متفاوتی برای $f(x)$ بدست بیاوریم الگوریتم را متوقف کرده و حکم می کنیم که تابع متوازن است. اما اگر چنین نبود الگوریتم را ادامه می دهیم تا نیمی از اعداد بعلاوه یک یعنی $2^{n-1} + 1$ عدد خوانده شوند. اگر برای این تعداد ورودی نیز تابع مقدار یکسان داشته باشد حتماً تابع ثابت است.

حال نشان می دهیم که با استفاده از یک الگوریتم کوانتومی می توان تنها با یک بار فراخوانی تابع، متوازن بودن یا ثابت بودن آن را با قطعیت تعیین کرد. برای این کار از مدار شکل ۲ استفاده می کنیم که دقیقاً همان مدار شکل ۳ است، با این تفاوت که در خط بالایی $|0\rangle$ نشان دهنده یک حالت صفر n تایی یعنی $|000 \cdots 0\rangle$ است و H نیز نشان دهنده یک عملگر هادامارد n تایی است که روی n کیوبیت عمل می کند. به عبارت دیگر عمل هادامارد به صورت $H^{\otimes n}$ عمل می کند. حال با توجه به این شکل، بردار حالت را در مراحل مختلف می نویسیم. داریم

$$|\Psi_0\rangle = |s\rangle \otimes |-\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x, 0\rangle - |x, 1\rangle$$

$$|\Psi_1\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (|x, f(x)\rangle - |x, \overline{f(x)}\rangle) = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle \otimes (|0\rangle - |1\rangle). \quad (20)$$

بنابراین n کیویت اول در حالت $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle$ هستند. حال اگر تابع f یک تابع ثابت باشد، آنگاه این حالت برابر است با $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle$ و عمل گیت های هادامارد در سمت راست مدار، روی این حالت حالت $|000\dots 0\rangle$ را تولید خواهد کرد. اما اگر تابع f یک تابع متوازن باشد، ثابت می کنیم که عمل گیت های هادامارد، حالتی را تولید خواهد کرد که اندازه گیری اش به هیچ وجه حالت $|000\dots 0\rangle$ را تولید نخواهد کرد. برای اینکه این نکته را بفهمیم باید ثابت کنیم که برای یک تابع متوازن

$$\langle 000\dots 0 | H^{\otimes n} | \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle = 0. \quad (21)$$

برای اثبات این رابطه $H^{\otimes n}$ را روی حالت سمت چپ اثر می دهیم. در نتیجه سمت چپ عبارت فوق برابر می شود با

$$\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} = 0. \quad (22)$$

بنابراین ثابت کرده ایم که برای یک تابع ثابت، n کیویت بالا در مدار شکل 2، در حالت $|000\dots 0\rangle$ قرار می گیرند و برای یک تابع متوازن این n کیویت در هر حالتی غیر از صفر قرار خواهند گرفت. بنابراین با یک بار اندازه گیری این حافظه ی n کیویتی می توانیم ثابت بودن یا متوازن بودن یک تابع را تشخیص دهیم.

■ تمرین: نشان دهید که با استفاده از الگوریتم های تصادفی می توان مسئله دویچ-جوزا را در زمان چندجمله ای حل کرد. برای این کار k بار تابع را فراخوانی کنید. اگر در همه موارد مقدار تابع یکسان بود اعلام کنید که تابع ثابت است و در غیر این صورت اعلام کنید که تابع متوازن است. برای آنکه خطای شما در این الگوریتم تصادفی به طور متوسط از ϵ کمتر باشد، مقدار کمینه ی k را پیدا کنید.

■ تمرین: مسئله دویچ-جوزا را به این صورت تعمیم می دهیم. یک جعبه سیاه تابع زیر را محاسبه می کند:

$$f : \{0, 1, 2, \dots, d-1\}^{\times n} \longrightarrow \{0, 1, 2, \dots, d-1\}. \quad (23)$$

یعنی این تابع از n تا دیت (بجای بیت) به یک بیت تعریف شده است. گفته شده است که این تابع یا متوازن است یا ثابت. در حالت کلاسیک چند بار می بایست تابع را فراخوانی کنیم تا بفهمیم که آیا تابع ثابت است یا متوازن؟ هرگاه بجای کیوبیت ها کیودیت (سیستم های اسپینی d حالت) در اختیار داشته باشید یک الگوریتم کوانتومی طراحی کنید که بتواند این مسئله را موثرتر از مسئله کلاسیک حل کند. یک الگوریتم تصادفی و کلاسیک با چندبار فراخوانی تابع می تواند این مسئله را حل کند به شرطی که احتمال پاسخ اشتباه از ϵ کمتر باشد؟

۵ الگوریتم کوانتومی برای مسئله برنشتاین - وزیرانی

نخست مسئله برنشتاین - وزیرانی را تعریف می کنیم. فرض کنید که تابع $f: B_n \rightarrow B_1$ داده شده است. این تابع به شکل $f(x) := a \cdot x$ تعریف شده است که در آن a یک عدد n بیتی است. ضرب این دو عدد به شکل زیر تعریف می شود:

$$a \cdot x = \sum_{k=0}^{n-1} a_k x_k \mod 2. \quad (24)$$

سوال این است که با چند بار فراخواندن تابع می توان مقدار a را تعیین کرد. با الگوریتم های کلاسیک واضح است که می بایست n بار تابع را فراخوانی کرد تا a تعیین شود. نشان می دهیم که با یک الگوریتم کوانتومی می توان تنها با یک بار فراخوانی مقدار a را مشخص کرد. برای این الگوریتم باز هم از همان مدار شکل 2 استفاده می کنیم. می دانیم که در این مدار حالت زیر تولید می شود.

$$|\Psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{a \cdot x} |x\rangle. \quad (25)$$

دقت کنید که برای سادگی حالت کیوبیت پایینی یعنی $(|0\rangle - |1\rangle)/\sqrt{2}$ را ننوشتیم، زیرا در بقیه بحث ما دخالتی ندارد. حال اثر عملگرهای هادامارد این حالت را تبدیل به حالت زیر می کند:

$$|\Psi_3\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{x \cdot (a+y)} |x\rangle. \quad (26)$$

اما می دانیم که

$$\sum_{x=0}^{2^n-1} (-1)^{(a+y) \cdot x} = 2^n \delta_{a,y}. \quad (27)$$

بنابراین خواهیم داشت $|\Psi_3\rangle = |a\rangle$ ، یعنی در حقیقت مقدار a تنها با یک بار اثر کردن تابع در n کیوبیت بالا نوشته خواهد شد.

■ تمرین: فرض کنید که به جای کیوبیت ها از کیودیت ها استفاده می کنیم. مسئله برنشتاین - وزیرانی را به صورت زیر تعمیم می دهیم. یک جعبه سیاه تابع زیر را محاسبه می کند:

$$f : \{0, 1, 2, \dots, d-1\}^{\times n} \longrightarrow \{0, 1, 2, \dots, d-1\}, \quad f(x) := x \cdot a \equiv x_1 a_1 + x_2 a_2 + \dots + x_n a_n \pmod{d}. \quad (28)$$

در حالت کلاسیک چند بار می بایست تابع را فراخوانی کنیم تا مقدار a را بفهمیم.

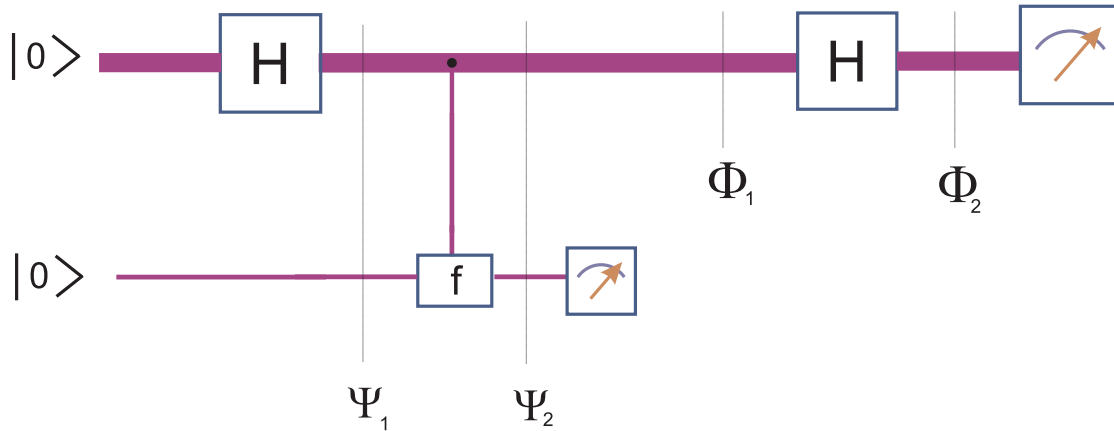
یک الگوریتم کوانتومی طراحی کنید تا بتواند با یک بار خواندن مقدار a را پیدا کند.

۶ الگوریتم کوانتومی برای مسئله سیمن

فرض کنید که تابع $f : B_n \longrightarrow B_n$ دارای این خاصیت است که یک و فقط یک $a \in B_n$ وجود دارد به قسمی که $f(x) = f(x \oplus a) \quad \forall x \in B_n$. مسئله سیمن *Simon* عبارت است از یافتن مقدار a با استفاده از فراخوانی های تابع. حال سوال این است که با چند بار خواندن تابع می توانیم مقدار a را تعیین کنیم. برای این مسئله می بایست از یک مدار کوانتومی متفاوت استفاده کنیم. این مدار در شکل 3 نشان داده شده است.

مراحل مختلف این الگوریتم را با نوشتن حالت هایی که در روی شکل مربوط به آن نوشته شده است بیان می کنیم. داریم

$$|\Psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, 0\rangle \quad (29)$$



شکل ۳: مداری که برای الگوریتم سیمون به کار می رود.

$$|\Psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, f(x)\rangle \quad (30)$$

مرحله بعدی آن است که حافظه دوم یعنی آن قسمتی که $f(x)$ را در بردارد اندازه گیری کنیم. فرض کنید که مقدار $f(x_0)$ بدست آید. دراین صورت حافظه اول به حالت زیر تقلیل پیدا می کند:

$$|\Phi_1\rangle = \frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus a\rangle). \quad (31)$$

در مرحله بعدی روی این حالت گیت های هادامارد $H^{\otimes n}$ را اثر می دهیم. حالتی که بدست می آید برابر است با:

$$|\Phi_2\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} [(-1)^{x_0 \cdot a}] |y\rangle. \quad (32)$$

باکمی محاسبه خواننده می تواند نشان دهد که این حالت برابر است با:

$$|\Phi_2\rangle = \frac{1}{\sqrt{2^{n-1}}} \sum_{a \cdot y=0} [(-1)^{x_0 \cdot y}] |y\rangle. \quad (33)$$

این امر به این معناست که هرگاه ما حالت $|\Phi_2\rangle$ را اندازه بگیریم، عددی مثل y بدست می آید که در رابطه ای $a \cdot y = 0$ صدق می کند. با حدود n بار اندازه گیری می توانیم تعداد n معادله مستقل برای a پیدا کنیم که حل کردن آنها عدد a را بدست می دهد.

باید دقت کنیم که تا کنون هیچ کدام از الگوریتم های معرفی شده کلاس حل پذیری مسئله را تغییر نداده اند. یعنی نتوانسته اند مسئله ای را از کلاس نمایی به کلاس چند جمله ای تبدیل کنند. اما این الگوریتم ها مقدمه ای خواهند بود برای الگوریتم بسیار

مهم شُر^۷ که در درس های آینده با آن آشنا خواهیم شد. (در یکی از درس های آینده که به نظریه محاسبه می پردازیم کلاس پیچیدگی را به طور دقیق تعریف می کنیم، ولی به یک معنای خیلی کیفی می توان گفت که کلاس نمایی شامل مسائلی است که زمان حل آنها نسبت به ساین مسئله ها رابطه نمایی دارد. این نوع مسائل را مسائل سخت می نامیم. کلاس چند جمله ای شامل مسائلی است که زمان حل آنها با ساین مسئله ها رابطه توانی دارد و بنابراین می توان آنها را مسائل آسان نامید). هم چنین درس بعدی را نظر به اهمیتی که مسئله جستجو در مجموعه های نامنظم دارد به الگوریتم کوانتومی جستجو اختصاص خواهیم داد.

■ تمرین: فرض کنید که به جای کیویت ها از کیودیت ها استفاده می کنیم. مسئله سیمون را به صورت زیر تعمیم می دهیم. یک جعبه سیاه تابع زیر را محاسبه می کند:

$$f : \{0, 1, 2, \dots, d-1\}^{\times n} \longrightarrow X \quad (34)$$

که در آن X یک مجموعه دلخواه است و این شرط اضافه که

$$f(\mathbf{x}) = f(\mathbf{y}) \quad \text{if and only if} \quad \mathbf{y} = \mathbf{x} + \mathbf{a} \pmod{d} \quad (35)$$

که در آن منظور از طرف راست این است که

$$y_i = x_i + a_i \pmod{d} \quad \forall i = 1, \dots, n. \quad (36)$$

در حالت کلاسیک چند بار می بایست تابع را فراخوانی کنیم تا مقدار a را بفهمیم.

یک الگوریتم کوانتومی طراحی کنید تا بتواند با یک بار خواندن مقدار a را پیدا کند.

۷ الگوریتم های کوانتومی بیشتر

باید توجه کنیم که به جز الگوریتم هایی که در بالا نام بردیم تعداد زیادی الگوریتم های دیگر در سالهای اخیر ابداع شده اند که همگی نشان دهنده توان بالای کامپیوترهای کوانتومی هستند. به تدریج که پیش می رویم و شهود کوانتومی پژوهشگران تقویت می شود ممکن است مجموعه الگوریتم های کوانتومی نیز روز به روز گسترده تر شود. در بخش بعدی تنها زیر مجموعه کوچکی از این مسئله ها را که بیان ساده ای دارند و برای آنها الگوریتم های کوانتومی ابداع شده است معرفی می کنیم. خواننده برای یافتن الگوریتم های بیشتر و هم چنین مراجع اولیه ای که این الگوریتم ها در آن ها معرفی شده اند می تواند به وبگاه زیر رجوع کند:

<http://math.nist.gov/quantum/zoo/>,

Stephen Jordan Applied and Computational Mathematics Division, NIST

در این وبگاه مجموعه ۵۲ نوع الگوریتم کوانتومی فهرست شده اند البته برای شرح هر الگوریتم خواننده می بایست به مراجع درج شده در این وبگاه نگاه کند. تاکید می کنیم که بجز الگوریتم جستجو و الگوریتم تجزیه اعداد که در فصل های آینده آن ها را به تفصیل معرفی می کنیم در بقیه موارد خواننده علاقمند برای فهم این الگوریتم ها می بایست به مقالات اصلی رجوع کند.

در مقایسه الگوریتم ها معمولا از یک نماد گذاری استفاده می کنیم که در زیر این نماد گذاری را توضیح می دهیم. این نماد گذاری را معمولا به این معنا به کار می بریم که وابستگی دو تابع از متغیرهایشان از یک مرتبه است و می نویسیم $f = O(g)$. اما بهتر است که به طور دقیق تر این نمادگذاری را توضیح دهیم.

■ یک نکته در باره نماد گذاری: دو تابع $f, g : \mathcal{N} \rightarrow \mathcal{N}$ از اعداد صحیح $\mathcal{N}$ به اعداد صحیح را در نظر بگیرید. می گوئیم که

$$f = O(g) \quad (37)$$

هرگاه یک عدد صحیح مثبت مثل N و یک عدد مثبت حقیقی مثل c وجود داشته باشد به قسمی که

$$f(n) \leq c g(n), \quad \forall n > N. \quad (38)$$

هرگاه داشته باشیم

$$f = O(g), \quad g = O(f)$$

می نویسیم

$$f \sim g.$$

■ تمرین: درستی روابط زیر را نشان دهید:

$$\begin{aligned} 2n^3 + 3n + 6 &= O(n^3), \\ n^3 &= O(n^4 + 6n) \\ n^3 &\sim 2n^3 + 6n \\ 2n^2 &= O(2^n). \end{aligned} \quad (39)$$

■ تمرین: کدام یک از روابط زیر درست نیستند:

$$\begin{aligned} 2n^3 + 3n + \frac{1}{n} &= O(n^3), \\ n^5 + n^3 &= O\left(\frac{1}{10^6}n^4 + 6n\right) \\ n^3 &\sim 2n^3 + 6n \\ 2^n &= O(n^{0.00001 n}). \end{aligned} \quad (40)$$

بعد از این توضیح به ذکر چند تا از الگوریتم های کوانتومی مهم می پردازیم. در هر مورد تنها نشان می دهیم که چگونه زمان اجرای این الگوریتم ها در حالت کلاسیک و کوانتومی باهم تفاوت دارد. در مورد مفهوم دقیق تر زمان اجرای الگوریتم ها در درس نظریه محاسبه به طور دقیق تر بحث خواهیم کرد.

۱.۷ جستجو در یک مجموعه نامنظم

■ توصیف مسئله:

یک مجموعه N عضوی S داده شده است که در آن یکی از اعضا یک خاصیت معین دارد. این مجموعه هیچ نوع نظم خاصی ندارد. می خواهیم آن عضو خاص را پیدا کنیم. به زبان دیگر، یک تابع از آن مجموعه به مجموعه دو عضوی $\{0, 1\}$ داده شده با این شرط که

$$f(x) = 0, \quad \forall x \in S, \text{ except } w \text{ for which } f(w) = 1. \quad (41)$$

می خواهیم w را پیدا کنیم. چندبار تابع را می بایست فراخوانی کنیم؟

■ زمان لازم برای الگوریتم کلاسیک:

$$T_{cl}(N) \sim (N) \quad (42)$$

■ زمان لازم برای الگوریتم کوانتومی:

$$T_q(N) \sim (\sqrt{N}). \quad (43)$$

■ تمرین: برای جستجوی کلاسیک یکی از این دو روش را به کار می بریم.

روش اول: متغیرها را به طور کاملاً تصادفی به تابع می دهیم تا مقدار تابع برابر با 1 شود و متغیر مورد نظر را پیدا کنیم.

روش دوم: متغیرها را یک به یک و به طور منظم به تابع می دهیم تا مقدار تابع برابر با یک شود و متغیر مورد نظر را پیدا کنیم.

در هر دو مورد متوسط تعداد دفعاتی که می بایست تابع را فراخوانی کنیم تا متغیر مورد نظر را پیدا کنیم به طور دقیق حساب کنید .

۲.۷ تجزیه یک عدد

■ توصیف مسئله:

یک عدد n رقمی داده شده است. هدف یافتن یکی از عامل های این عدد است.

■ زمان لازم برای الگوریتم کلاسیک:

$$T_{cl}(n) \sim e^{1.9n^{\frac{1}{3}}(\log n)^{\frac{2}{3}}} \quad (44)$$

■ زمان لازم برای الگوریتم کوانتومی:

$$T_q(n) \sim n^3 \quad (45)$$

■ تمرین: با استفاده از یک برنامه ساده مثل *Maple* یا *Mathematica* یک مجموعه از جفت اعداد اول (p, q) را در هم ضرب کرده و سپس حاصل را تجزیه کنید. سعی کنید که تعداد ارقام این دو عدد در گام های ۲ تا ۴ از ۲۰ تا ۲۰ رقم تغییر کند. زمان های لازم برای ضرب دو عدد و سپس تجزیه آنها را روی یک نمودار رسم کنید.

■ تمرین: فرض کنید که روی یک کامپیوتر، تجزیه یک عدد ۱۰۰ رقمی با الگوریتم کلاسیک یک ماه طول می کشد. زمان لازم برای تجزیه اعداد ۲۰۰ رقمی، ۳۰۰ رقمی، و ۴۰۰ رقمی را روی همان کامپیوتر پیدا کنید. حال فرض کنید که روی یک کامپیوتر کوانتومی زمان لازم برای تجزیه آن عدد ۱۰۰ رقمی با الگوریتم کوانتومی یک ماه طول بکشد. زمان لازم را برای تجزیه اعداد فوق را روی این کامپیوتر کوانتومی پیدا کنید.

۳.۷ لگاریتم گسسته

■ توصیف مسئله:

سه عدد صحیح a, b, N داده شده با این قول که

$$b = a^s \pmod{N}. \quad (۴۶)$$

هدف پیدا کردن s است. با توجه به این که

$$s = \log_a(b), \pmod{N}, \quad (۴۷)$$

این مسئله لگاریتم گسسته خوانده می شود.

■ زمان لازم برای الگوریتم کلاسیک:

هیچ الگوریتم کلاسیک خوبی برای این مسئله وجود ندارد. تنها می توان a را به توان های اعداد متوالی رساند تا این که نتیجه برابر با b شود. بنابراین

$$T_{cl}(N) \sim e^{O(\log N)} \quad (۴۸)$$

■ زمان لازم برای الگوریتم کوانتومی:

$$T_q(N) \sim (\log N)^3 \quad (۴۹)$$

۴.۷ معادله پل

■ توصیف مسئله:

معادله پل^۸ به این صورت است:

$$x^2 - dy^2 = 1, \quad (50)$$

که در آن d یک عدد صحیح مثبت است که مربع کامل نیست. هدف پیدا کردن حلی از این معادله به صورت یک جفت عدد صحیح (x_1, y_1) است که $x_1 + \sqrt{d}y_1$ را کمینه کند.

■ تمرین: سه تا از حل های معادله زیر را پیدا کنید

$$x^2 - 6y^2 = 1$$

و از بین آنها حلی را که $x + \sqrt{6}y$ را کمینه می کند پیدا کنید.

■ زمان لازم برای الگوریتم کلاسیک:

هیچ الگوریتم کلاسیکی که بتواند این مسئله را در زمان چندجمله ای حل کند در دست نیست. مسئله تجزیه اعداد را می توان به این مسئله مرتبط کرد. حل این مسئله برای شکستن کدهای رمزنگاری که مبتنی بر خواص منحنی های بیضوی هستند کاربرد دارد.

■ زمان لازم برای الگوریتم کوانتومی:

اگر d یک عدد n رقمی باشد، آنگاه

$$T_q(n) \sim Poly(n). \quad (51)$$

۵.۷ تصدیق ضرب ماتریسی

■ توصیف مسئله:

سه ماتریس $N \times N$ با درایه های حقیقی مثل A , B , و C داده شده است. می خواهیم ببینیم که آیا $AB = C$ هست یا نه؟

^۸Pell's equation

■ زمان لازم برای الگوریتم کلاسیک:

$$T_{cl}(N) \sim N^2. \quad (52)$$

■ زمان لازم برای الگوریتم کوانتومی:

$$T_q(N) \sim N^{\frac{5}{3}}. \quad (53)$$

۶.۷ مسئله جمع زیرمجموعه ها

■ توصیف مسئله:

مجموعه ای از اعداد صحیح مثل $\{x_1, x_2, \dots, x_n\}$ و یک عدد صحیح s در نظر بگیرید. می خواهیم ببینیم که آیا یک زیر مجموعه از این اعداد وجود دارد به قسمی که مجموع آنها برابر با s شود؟ این مسئله یک مسئله $NP - Complete$ است به قسمی که هر مسئله سخت دیگری در زمان چندجمله ای به آن تبدیل می شود.

■ زمان لازم برای الگوریتم کلاسیک:

سخت ترین شرایط وقتی است که اعداد داخل مجموعه از مرتبه 2^n باشند. در این شرایط بهترین الگوریتم کلاسیک زمان زیر را دارد:

$$T_{cl}(n) \sim 2^{0.291 n} \quad (54)$$

■ زمان لازم برای الگوریتم کوانتومی:

$$T_q(n) \sim 2^{0.241 n}. \quad (55)$$

۷.۷ مسئله زیرگروه های پنهان آبدلی

■ توصیف مسئله:

این مسئله تعمیمی از مسئله سیمون یا مسئله شریا هر نوع مسئله *Period Finding* است. فرض کنید که G یک گروه آبدلی متناهی و H یک زیرگروه آن باشد. دو عضو g_1 و g_2 را هم ارز می گوئیم و می نویسیم $g_2 \sim g_1$ هرگاه یک عضو از زیر گروه H مثل h وجود داشته باشد به نحوی که

$$g_2 = g_1 h. \quad (56)$$

رابطه ای که در بالا تعریف کرده ایم یک رابطه هم ارزی است و گروه G را افراز می کند. یک کلاس هم ارزی را برای یک عضو g با $[g]$ نشان می دهیم و مجموعه تمام کلاس های هم ارزی را با G/H نشان می دهیم. حال فرض کنید که یک تابع $f: G \rightarrow K$ وجود دارد که در آن K یک مجموعه دلخواه است. به ما گفته شده است که تابع f دارای این خاصیت است که

$$f(g_2) = f(g_1) \quad \text{if and only if } g_2 \sim g_1. \quad (57)$$

به عبارت دیگر مقدار تابع f روی دو عضو مساوی است اگر و فقط اگر آنها متعلق به یک کلاس هم ارزی باشند. یک کلاس هم ارزی را یک هم-مجموعه یا *Coset* می نامیم. مسئله این است که چندبار می بایست تابع f را فراخوانی کنیم تا بتوانیم زیر گروه H را بیابیم.

■ زمان لازم برای الگوریتم کلاسیک:

$$T_{cl}(G) \sim |G| \quad (58)$$

که در آن $|G|$ تعداد اعضای گروه است.

■ زمان لازم برای الگوریتم کوانتومی:

$$T_q(G) \sim \log(|G|) \quad (59)$$

■ تمرین: نشان دهید که مسئله های دویچ، و سیمون حالت های خاصی از مسئله زیر گروه پنهان آبدلی هستند.

۸.۷ مسئله زیرگروه های پنهان غیرآبلی

- توصیف مسئله: همان مسئله بالاست با این تفاوت که گروه G غیر آبلی است. در حالت کلی هنوز هیچ الگوریتم کوانتومی کلی برای این مسئله یافته نشده است. تنها برای بعضی از گروه های خاص این مسئله قابل حل است.
- زمان لازم برای الگوریتم کلاسیک:

$$T_{cl}(G) \sim |G|, \quad (60)$$

- زمان لازم برای الگوریتم کوانتومی:

$$T_q(G) \sim \log(|G|), \quad \text{only for some special groups.} \quad (61)$$

۹.۷ حل دستگاه معادلات خطی

- توصیف مسئله:

یک ماتریس مربعی A که $N \times N$ است و یک بردار N بعدی b و معادله زیر را در نظر بگیرید:

$$Ax = b. \quad (62)$$

در بسیاری از مسایل بیش از آنکه نیازمند دانستن خود x باشیم می خواهیم متوسط کمیت های مختلفی مثل M را روی این بردار به صورت

$$x^T M x \quad (63)$$

حساب کنیم. روی ماتریس A نیز شرایط متعارفی وجود دارد به این معنا که : یک : ماتریس A خلوت^۹ است به این معنا که تعداد عناصر غیر صفر در هر سطر آن از مرتبه $O(\log N)$ است. و

^۹ Sparse

دو: ویژه مقدارهای آن در شرط زیر صدق می کنند:

$$\frac{1}{\kappa} \leq \lambda \leq 1. \quad (64)$$

برای آنکه این رابطه را بفهمیم ، دقت می کنیم که همواره می توان با تقسیم یک ماتریس بر بزرگترین ویژه مقدارش ماتریس جدیدی بدست آورد که بزرگترین ویژه مقدار آن برابر با 1 است و هیچ تغییری نیز در حل مسئله خطی بوجود نخواهد آمد. نسبت بزرگترین ویژه مقدار به کوچکترین ویژه مقدار یک ماتریس را با پارامتر κ نشان می دهیم. هر چه که این پارامتر بزرگتر باشد، معکوس کردن ماتریس دشوارتر می شود زیرا به این معناست که کوچکترین ویژه مقدار به صفر نزدیک تر شده و در نتیجه ماتریس به یک ماتریس وارون ناپذیر نزدیک تر می شود. به همین دلیل است که در همه آگوریتم ها علاوه بر بعد ماتریس، این پارامتر نیز اهمیت دارد.

■ زمان لازم برای آگوریتم کلاسیک:

$$T_{cl}(N) \sim \sqrt{N\kappa} \quad (65)$$

■ زمان لازم برای آگوریتم کوانتومی:

$$T_q(N) \sim \sqrt{(\log N)\kappa}. \quad (66)$$

۱۰.۷ جستجو در داده های مرتب

■ توصیف مسئله:

■ زمان لازم برای آگوریتم کلاسیک:

■ زمان لازم برای آگوریتم کوانتومی:

۱۱.۷ جابجایی ماتریس ها

■ توصیف مسئله: تعداد K ماتریس از مجموعه

$$S = \{X_1, X_2, \dots, X_K\} \quad (۶۷)$$

در نظر بگیرید. این ماتریس ها $n \times n$ هستند. تابعی داریم که به ازای هر $i, j \in \{1, 2, \dots, n\}$ و هر $k \in \{1, 2, \dots, K\}$ عنصر زیر را حساب می کند:

$$F : (i, j, k) \longrightarrow (X_k)_{i,j}. \quad (۶۸)$$

سوال این است که چند بار می بایست این تابع فراخوانی شود تا بفهمیم آیا ماتریس های مجموعه S با هم جابجا می شوند یا نه؟

■ زمان لازم برای الگوریتم کلاسیک:

$$T_{cl}(n, K) \sim Kn^2 \quad (۶۹)$$

■ زمان لازم برای الگوریتم کوانتومی:

$$T_q(n, K) \sim K^{\frac{4}{5}} n^{\frac{9}{5}}. \quad (۷۰)$$

۱۲.۷ جابجایی گروه ها

■ توصیف مسئله:

یک گروه G و مجموعه مولدهای آن به صورت

$$S = \{g_1, g_2, \dots, g_K\} \quad (۷۱)$$

داده شده است. معنای این حرف این است که با ضرب کردن این عناصر و وارون های آنها به تمام ترتیبات ممکن می توان تمام عناصر گروه را تولید کرد. یک تابع F داریم که عمل ضرب گروه را انجام می دهد به این معنا که :

$$F : (g, g') = gg'. \quad (72)$$

حال سوال این است که چندبار می بایست تابع F را فراخوانی کنیم تا بتوانیم تشخیص دهید آیا گروه G آبلی هست یا نه؟ در نگاه اول به نظر می رسد که می بایست $2K^2$ بار این تابع را فراخوانی کرد . اما ظاهراً الگوریتم های کلاسیک بهتری نیز وجود دارد.

■ زمان لازم برای الگوریتم کلاسیک:

$$T_{cl}(K) \sim K \quad (73)$$

■ زمان لازم برای الگوریتم کوانتومی:

$$T_q(K) \sim K^{\frac{2}{3}}. \quad (74)$$