

پروژه محاسبات کوانتومی

درس طراحی الگوریتم

امیر محمد درپوش

سید صالح اعتمادی

دانشگاه علم صنعت ۱۴۰۲-۱۴۰۳

توضیحات کلی پروژه

این پروژه شامل 6 تمرین برنامه نویسی در زبان برنامه نویسی کوانتومی **q#** است. برای حل این تمرین ها ابتدا باید افزونه برنامه نویسی **Azure Quantum Development kit** را به نسخه **vs code** خود اضافه کرده و سپس کد های داده شده را کامل کنید. هر تمرین شامل یک تابع تست (Test) و یک تابع **Qn** ناقص است که در آن باید منطق کامپیوتر کوانتومی پیاده سازی شود.

برای ساده سازی محیط توسعه می‌توانید از [این](#) لینک استفاده کنید و بعد از پیدا کردن مدار درست، آن را در **q#** پیاده سازی کنید.

منابع پیشنهادی برای یادگیری

هدف این پروژه آشنایی شما با مفاهیم پایه ای و دایره لغات الگوریتم های کوانتومی است. به این منظور سوالات طراحی شده نسبتاً ساده هستند و تنها با استفاده از گیت ها **H** و **CNOT** حل می‌شوند. به این منظور برای آشنا شدن با این مفاهیم این سری بلاگ ها از جاناتان هوپی را مطالعه کنید.

1. [QC — What is a Quantum Computer?. Hype drives curiosity but also causes... | by Jonathan Hui | Medium](#)
2. [QC — What are Qubits in Quantum computing? | by Jonathan Hui | Medium](#)
3. [QC — Control quantum computing with unitary operators, interference & entanglement | by Jonathan Hui | Medium](#)
4. [QC — Programming with Quantum Gates \(Single Qubits\) | by Jonathan Hui | Medium](#)
5. [QC — Programming with Quantum Gates \(2-Qubit operator\) | by Jonathan Hui | Medium](#)
6. [QC — Quantum Algorithm with an example | by Jonathan Hui | Medium](#)
7. Quantum Computing **for Computer Scientists** ([slides](#), [video](#))

این منابع برای حل سوالات خواسته شده کافی است ولی در صورت علاقه به مطالعه بیشتر می‌توانید از منابع زیر استفاده کنید.

- The [QuantumCasts](#) Series by Google.
- **Infinite Series** on Quantum Computing
 - **The Mathematics** of Quantum Computers Infinite Series ([YouTube](#))
 - How to **Break Cryptography** Infinite Series ([YouTube](#))
 - Hacking at Quantum Speed with **Shor's Algorithm** Infinite Series ([YouTube](#))
- **Shor's Algorithm** Explained ([YouTube](#))
- **Microsoft** Resources for Quantum Computing ([link](#))
- **IBM** Resources for Quantum Computing ([link](#))
- **Prof. Salman Beigi's** [lecture notes](#) on Quantum Computing.
- UC Berkeley's [Quantum Lab](#)
- [List](#) of universities with Quantum Computing research groups.
- University of Waterloo [Institute for Quantum Computing](#)
- Notable Quantum Computing Researchers
 - [Aram Harrow](#)
 - [Peter Shor](#)
- Comprehensive list of [Quantum Algorithms](#)

1 ایجاد برهم نهی (superposition) یکنواخت تمام حالت های پایه

به شما N کیوبیت در حالت 0 داده می شود. $|0...0\rangle$

شما باید یک برهم نهی یکسان از تمام 2^N حالت موجود بردار های پایه بسازید.

$$|S\rangle = \frac{1}{\sqrt{2^N}} (|0...0\rangle + \dots + |1...1\rangle)$$

به عنوان مثال به ازای $N = 2$ حالت مورد نظر برابر است با:

$$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

2 ساخت حالت GHZ

به شما N کیوبیت در حالت 0 داده می شود. شما باید حالت Greenberger-Horne-Zeilinger (GHZ) را در آن ها ایجاد کنید.

$$|GHZ\rangle = \frac{1}{\sqrt{2}} (|0...0\rangle + |1...1\rangle)$$

3 ساخت برهم نهی حالت صفر و یکی از حالت های پایه

به شما N کیوبیت و یک رشته باینری داده شده است. این رشته یکی از حالت های پایه غیر صفر را توصیف می کند. شما باید یک برهم نهی مساوی از حالت 0 و حالت داده شده ایجاد کنید.

$$|S\rangle = \frac{1}{\sqrt{2}} (|0...0\rangle + |\psi\rangle)$$

- اولین بیت ψ همیشه 1 است.

4 درهم تنیده کردن دو کیوبیت

در این سوال به شما دو کیوبیت داده می شود. شما باید این دو کیوبیت را با هم درهم تنیده کرده (Entangle) و در نهایت به صورت کامنت ویژگی های این حالت را توضیح دهید.

5 تشخیص حالت صفر

به شما N کیوبیت داده می‌شود. شما باید تشخیص دهید که این کیوبیت در حالت صفر است یا حالت W .

حالت پایه: $|0...0\rangle$

$$\text{حالت } W: |W\rangle = \frac{1}{\sqrt{N}}(|100...0\rangle + |010...0\rangle + \dots + |00...01\rangle)$$

6 آشنایی با الگوریتم Deutsch-Jozsa

الگوریتم Deutsch-Jozsa یک الگوریتم ساده است که اجرای آن روی کامپیوتر کلاسیک با پیچیدگی نمایی امکان پذیر است. اگر بتوانیم آن را روی کامپیوتر کوانتومی اجرا کنیم میتوانیم در زمان خطی آن را حل کنیم. فرض کنید تابعی به نام f داریم که یک بیت 0 یا 1 بعنوان ورودی گرفته و یک بیت 0 یا 1 بعنوان خروجی میدهد. اگر این تابع به ازای هر مقدار ورودی (0 یا 1) همواره یک خروجی دهد آن را $constant$ می‌نامیم و اگر خروجی آن به ازای ورودی های مختلف بین 0 و 1 تغییر کند آن را $balanced$ می‌نامیم.

هدف از الگوریتم Deutsch-Jozsa این است که تشخیص دهد تابع f از کدامیک از انواع گفته شده است. اگر بخواهیم برای ورودی تک بیتی این سوال را بصورت کلاسیک حل کنیم نیاز است که ۲ بار با دادن ۲ حالت مختلف ورودی (۰ یا ۱) خروجی را بسنجیم. اگر ورودی را ۲ بیتی در نظر بگیریم لازم است ۴ بار خروجی تابع را بسنجیم. در راه حل کوانتومی میتوانیم با در نظر گرفتن حالات مختلف ورودی بصورت $superposition$ ای از $qbit$ ها آن را در زمان خطی حل کنیم

راهنمای پیاده سازی الگوریتم Deutsch-Jozsa

در این سوال به شما یک اوراکل کوانتومی از N کیوبیت داده شده است. وظیفه شما تشخیص نوع این اوراکل است. تابع پیاده سازی شده توسط شما یک Boolean خروجی می‌دهد. در صورتی که این اوراکل $Constant$ است مقدار $false$ و در صورتی که این اوراکل $balanced$ است مقدار $true$ را برگردانید.