Basic quantum algorithms: Deutsch-Jozsa, Bernstein-Vazirani

Yipeng Huang

Rutgers University

October 10, 2022

Promise algorithms vs. unstructured search

Quantum algorithms offer exponential speedup in "promise" problems

A progression of related algorithms:

1. Deutsch's
2. Deutsch-Jozsa
3. Bernstein-Vazirani
4. Simon's
5. Shor's

Table of contents

Deutsch-Jozsa algorithm: simplest quantum algorithm showing advantage vs. classical

A Heist

- You break into a bank vault. The bank vault has 2^n bars. Three possibilities: all are gold, half are gold and half are fake, or all are fake.
- Even if you steal just one gold bar, it is enough to fund your escape from the country, forever evading law enforcement.
- You do not want to risk stealing from a bank vault with only fake bars.
- You have access to an oracle $f(x)$ that tells you if gold bar x is real.
- Using the oracle sounds the alarm, so you only get to use it once.

Deutsch-Jozsa algorithm: simplest quantum algorithm showing advantage vs. classical

More formal description

- The 2^n bars are either fake or gold. $f : \{0,1\}^n \to \{0,1\}$.
- Three possibilities:
  1. All are fake. f is constant. $f(x) = 0$ for all $x \in \{0,1\}^n$.
  2. All are gold. f is constant. $f(x) = 1$ for all $x \in \{0,1\}^n$.
  3. Half fake half gold. f is balanced.
     $$\left|\{x \in \{0,1\}^n : f(x) = 0\}\right| = \left|\{x \in \{0,1\}^n : f(x) = 1\}\right| = 2^{n-1}$$
- The oracle U works as follows: $U|c\rangle|t\rangle = |c\rangle|t \oplus f(c)\rangle$
- Try deciding if f is constant or balanced using oracle U only once.

What is in the oracle

For $n = 1$, four possibilities

	f_0	f_1	f_2	f_3
f(0)	0	0	1	1
f(1)	0	1	0	1
	f is constant 0	f is balanced	f is balanced	f is constant 1

Demonstration of Deutsch-Jozsa for the $n = 1$ case

Output of circuit is $c = 0$ iff f is constant

1. Initial state: $|c\rangle \otimes |t\rangle = |0\rangle \otimes |1\rangle = |0\rangle |1\rangle = |01\rangle$
2. After first set of Hadamards: $H \otimes H\Big(|0\rangle \otimes |1\rangle \Big) = H|0\rangle \otimes H|1\rangle = |+\rangle \otimes |-\rangle =$

$$\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{2}\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

From here, let's take an aside via matrix-vector multiplication to build intuition with interference and phase kickback.

Demonstration of Deutsch-Jozsa for the $n = 1$ case

Output of circuit is $c = 0$ iff f is constant

1. Initial state: $|c\rangle \otimes |t\rangle = |0\rangle \otimes |1\rangle = |0\rangle\,|1\rangle = |01\rangle$
2. After first set of Hadamards: $H \otimes H\Big(\,|0\rangle \otimes |1\rangle\,\Big) = |+\rangle\,|-\rangle =$
   $$\Big(\tfrac{1}{\sqrt{2}}|0\rangle + \tfrac{1}{\sqrt{2}}|1\rangle\Big)\Big(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle\Big) = \tfrac{1}{2}\Big(\,|0\rangle\,(\,|0\rangle - |1\rangle\,) + |1\rangle\,(\,|0\rangle - |1\rangle\,)\Big)$$
3. After applying oracle U:
   $$U\tfrac{1}{2}\Big(\,|0\rangle\,(\,|0\rangle - |1\rangle\,) + |1\rangle\,(\,|0\rangle - |1\rangle\,)\Big) = \tfrac{1}{2}\Big(\,|0\rangle\,(\,|f(0) \oplus 0\rangle - |f(0) \oplus 1\rangle\,) +$$
   $$|1\rangle\,(\,|f(1) \oplus 0\rangle - |f(1) \oplus 1\rangle\,)\Big) = \tfrac{1}{2}\Big(\,|0\rangle\,(\,|f(0)\rangle - |f(\bar{0})\rangle\,) + |1\rangle\,(\,|f(1)\rangle - |f(\bar{1})\rangle\,)\Big)$$

Demonstration of Deutsch-Jozsa for the $n = 1$ case

Output of circuit is $c = 0$ iff f is constant

1. Initial state: $|c\rangle \otimes |t\rangle = |0\rangle \otimes |1\rangle = |0\rangle|1\rangle = |01\rangle$
2. After first set of Hadamards: $\frac{1}{2}\Big(|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle) \Big)$
3. After applying oracle U: $U\frac{1}{2}\Big(|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle) \Big) =$
   $\frac{1}{2}\Big(|0\rangle\big(|f(0)\rangle - |\overline{f(0)}\rangle\big) + |1\rangle\big(|f(1)\rangle - |\overline{f(1)}\rangle\big) \Big)$
4. This last expression can be factored depending on f:
   $U\frac{1}{2}\Big(|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle) \Big) =$
   $\begin{cases} \frac{1}{2}\big(|0\rangle + |1\rangle\big)\big(|f(0)\rangle - |\overline{f(0)}\rangle\big) \text{ if } f(0) = f(1) \\ \frac{1}{2}\big(|0\rangle - |1\rangle\big)\big(|f(0)\rangle - |\overline{f(0)}\rangle\big) \text{ if } f(0) \neq f(1) \end{cases} = \begin{cases} |+\rangle|-\rangle & \text{if } f(0) = f(1) \\ |-\rangle|-\rangle & \text{if } f(0) \neq f(1) \end{cases}$

The trick where oracle's output on $|t\rangle$ affects phase of $|c\rangle$ is called phase kickback.

Demonstration of Deutsch-Jozsa for the $n = 1$ case

Output of circuit is $c = 0$ iff f is constant

1. Initial state: $|c\rangle \otimes |t\rangle = |0\rangle \otimes |1\rangle = |0\rangle |1\rangle = |01\rangle$
2. After first set of Hadamards: $\frac{1}{2}\Big(|0\rangle (|0\rangle - |1\rangle) + |1\rangle (|0\rangle - |1\rangle)\Big)$
3. After applying oracle U:
$$U\frac{1}{2}\Big(|0\rangle (|0\rangle - |1\rangle) + |1\rangle (|0\rangle - |1\rangle)\Big) = \begin{cases} |+\rangle |-\rangle & \text{if } f(0) = f(1) \\ |-\rangle |-\rangle & \text{if } f(0) \neq f(1) \end{cases}$$
4. After applying second H on top qubit:
$$\begin{cases} H \otimes I\big(|+\rangle |-\rangle \big) = |0\rangle |-\rangle & \text{if } f(0) = f(1) \\ H \otimes I\big(|-\rangle |-\rangle \big) = |1\rangle |-\rangle & \text{if } f(0) \neq f(1) \end{cases}$$

Deutsch-Jozsa programs and systems

Algorithm

David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. 1992.

Programs

Google Cirq programming example.

Implementation

- Mach-Zehnder interferometer implementation.
  `https://www.st-andrews.ac.uk/physics/quvis/simulations_html5/sims/SinglePhotonLab/SinglePhotonLab.html`
- Ion trap implementation. Gulde et al. Implementation of the Deutsch–Jozsa algorithm on an ion-trap quantum computer. Letters to Nature. 2003.

Mach-Zehnder interferometer implementation of Deutsch's algorithm

$$|0\rangle \xrightarrow{H} |+\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{cases} \xrightarrow{I} |+\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} & \xrightarrow{H} |0\rangle \\ \xrightarrow{Z} |-\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{bmatrix} & \xrightarrow{H} |1\rangle \\ \xrightarrow{-Z} -|-\rangle = \begin{bmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} & \xrightarrow{H} -|1\rangle \\ \xrightarrow{-ZZ=-I} -|+\rangle = \begin{bmatrix} \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{bmatrix} & \xrightarrow{H} -|0\rangle \end{cases}$$

Table of contents

Deutsch-Jozsa algorithm: Deutsch's algorithm for the $n > 1$ case

The state after the first set of Hadamards

1. Initial state: $|c\rangle \otimes |t\rangle = |0\rangle^{\otimes n} \otimes |1\rangle = |0\ldots0\rangle\,|1\rangle = |0\ldots01\rangle$
2. After first set of Hadamards: $|+\rangle^{\otimes n} \otimes |-\rangle = \frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} |c\rangle \otimes |-\rangle$

Deutsch's algorithm: Deutsch-Jozsa for the $n = 1$ case

The state after applying oracle U

1. Initial state: $|c\rangle \otimes |t\rangle = |0\rangle^{\otimes n} \otimes |1\rangle = |0\dots0\rangle\,|1\rangle = |0\dots01\rangle$
2. After first set of Hadamards: $|+\rangle^{\otimes n} \otimes |-\rangle = \frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} |c\rangle \otimes |-\rangle$
3. After applying oracle U:

$$
\begin{aligned}
U\Big(|+\rangle^{\otimes n} \otimes |-\rangle\Big) &= \frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} |c\rangle \otimes \left(\frac{|f(c)\rangle - |\overline{f(c)}\rangle}{\sqrt{2}}\right) \\
&= \frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} (-1)^{f(c)}\,|c\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)
\end{aligned}
$$

Lemma: the Hadamard transform

$H^{\otimes n}|c\rangle = \frac{1}{2^{n/2}}\sum_{m=0}^{2^n-1}(-1)^{c\cdot m}|m\rangle$

- $$\begin{aligned} &H^{\otimes n}|c\rangle \\ &= H|c_0\rangle \otimes H|c_1\rangle \otimes \ldots \otimes H|c_{n-1}\rangle \\ &= \frac{1}{\sqrt{2}}\Big(|0\rangle + (-1)^{c_0}|1\rangle\Big) \otimes \frac{1}{\sqrt{2}}\Big(|0\rangle + (-1)^{c_1}|1\rangle\Big) \otimes \ldots \otimes \frac{1}{\sqrt{2}}\Big(|0\rangle + (-1)^{c_{n-1}}|1\rangle\Big) \\ &= \frac{1}{2^{n/2}}\sum_{m=0}^{2^n-1}(-1)^{c_0m_0+c_1m_1+\ldots+c_{n-1}m_{n-1} \mod 2}|m\rangle \end{aligned}$$

- Try it out for $n = 1$: $H^{\otimes 1}|c\rangle = \frac{1}{2^{1/2}}\sum_{m=0}^{2^1-1}(-1)^{c\cdot m}|m\rangle =$
$\frac{1}{\sqrt{2}}(-1)^0|0\rangle + \frac{1}{\sqrt{2}}(-1)^c|1\rangle = \begin{cases} \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = |+\rangle & \text{if } |c\rangle = |0\rangle \\ \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = |-\rangle & \text{if } |c\rangle = |1\rangle \end{cases}$

Deutsch-Jozsa algorithm: Deutsch's algorithm for the $n > 1$ case

The state after applying oracle U

1. Initial state: $|c\rangle \otimes |t\rangle = |0\rangle^{\otimes n} \otimes |1\rangle = |0\ldots0\rangle\,|1\rangle = |0\ldots01\rangle$
2. After first set of Hadamards: $|+\rangle^{\otimes n} \otimes |-\rangle = \frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} |c\rangle \otimes |-\rangle$
3. After applying oracle U: $U\Big(|+\rangle^{\otimes n} \otimes |-\rangle\Big) = \frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} (-1)^{f(c)} |c\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$
4. After final set of Hadamards:

$$(H^{\otimes n} \otimes I)\left(\frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} (-1)^{f(c)} |c\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\right)$$

$$= \frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} (-1)^{f(c)} \left(\frac{1}{2^{n/2}} \sum_{m=0}^{2^n-1} (-1)^{c\cdot m} |m\rangle\right) \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$= \frac{1}{2^n} \sum_{c=0}^{2^n-1} \sum_{m=0}^{2^n-1} (-1)^{f(c)+c\cdot m} |m\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Deutsch-Jozsa algorithm: Deutsch's algorithm for the $n > 1$ case

Output of circuit is 0 iff f is constant

1. Initial state: $|c\rangle \otimes |t\rangle = |0\rangle^{\otimes n} \otimes |1\rangle = |0\ldots0\rangle\,|1\rangle = |0\ldots01\rangle$
2. After first set of Hadamards: $|+\rangle^{\otimes n} \otimes |-\rangle = \frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} |c\rangle \otimes |-\rangle$
3. After applying oracle U: $U\Big(|+\rangle^{\otimes n} \otimes |-\rangle\Big) = \frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} (-1)^{f(c)} |c\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$
4. After final set of Hadamards: $(H^{\otimes n} \otimes I)\left(\frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} (-1)^{f(c)} |c\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\right) =$
   $\frac{1}{2^n} \sum_{c=0}^{2^n-1} \sum_{m=0}^{2^n-1} (-1)^{f(c)+c\cdot m} |m\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$
5. Amplitude of upper register being $|m\rangle = |0\rangle$:

$$\frac{1}{2^n} \sum_{c=0}^{2^n-1} (-1)^{f(c)}$$

Deutsch-Jozsa algorithm: Deutsch's algorithm for the $n > 1$ case

Output of circuit is 0 iff f is constant

1. Initial state: $|c\rangle \otimes |t\rangle = |0\rangle^{\otimes n} \otimes |1\rangle = |0\ldots 0\rangle\,|1\rangle = |0\ldots 01\rangle$
2. After first set of Hadamards: $|+\rangle^{\otimes n} \otimes |-\rangle = \frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} |c\rangle \otimes |-\rangle$
3. After applying oracle U: $U\Big(|+\rangle^{\otimes n} \otimes |-\rangle\Big) = \frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} (-1)^{f(c)} |c\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$
4. After final set of Hadamards: $(H^{\otimes n} \otimes I)\left(\frac{1}{2^{n/2}} \sum_{c=0}^{2^n-1} (-1)^{f(c)} |c\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\right) =$
   $\frac{1}{2^n} \sum_{c=0}^{2^n-1} \sum_{m=0}^{2^n-1} (-1)^{f(c)+c\cdot m} |m\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$
5. Amplitude of upper register being $|m\rangle = |0\rangle$: $\frac{1}{2^n} \sum_{c=0}^{2^n-1} (-1)^{f(c)}$
6. Probability of measuring upper register to get $m = 0$:

$$\left| \frac{1}{2^n} \sum_{c=0}^{2^n-1} (-1)^{f(c)} \right|^2 = \begin{cases} \left|(-1)^{f(c)}\right|^2 = 1 & \text{if } f \text{ is constant} \\ 0 & \text{if } f \text{ is balanced} \end{cases}$$

Table of contents

Table of contents

The factoring problem

One way functions for cryptography

1. Multiplying two b-bit numbers: on order of b^2 time.
2. Best known classical algorithm to factor a b-bit number: on order of about $2^{\sqrt[3]{b}}$ time.

- Makes multiplying large primes a candidate one-way function.
- It's an open question of mathematics to prove whether one way functions exist.

Public key cryptography

Numberphile YouTube channel explanation of RSA public key cryptography:
`https://www.youtube.com/watch?v=M7kEpw1tn50`

The factoring problem

One way functions for cryptography

1. Multiplying two b-bit numbers: on order of b^2 time.
2. Best known classical algorithm to factor a b-bit number: on order of about $2^{\sqrt[3]{b}}$ time.

Quantum integer factoring algorithm

- Quantum algorithm to factor a b-bit number: b^3.
- Peter Shor, 1994.
- Important example of quantum algorithm offering exponential speedup.

Table of contents

The classical part: converting factoring to order finding / period finding

General strategy for the classical part

1. Factoring
2. Modular square root
3. Discrete logarithm
4. Order finding
5. Period finding

The fact that a quantum algorithm can support all these primitives leads to additional ways that future quantum computing can be useful / threatening to existing cryptography.

Factoring

$$N = pq$$

$$N = 15 = 3 \times 5$$

Modular square root

Finding the modular square root

$$s^2 \mod N = 1$$

$$s = \sqrt{1} \mod N$$

Trivial roots would be $s = \pm 1$.

- Are there other (nontrivial) square roots?
- For $N = 15$, $s = \pm 4$, $s = \pm 11$, $s = \pm 14$ are all nontrivial square roots. (Show this).
- Later in these slides, we will see how nontrivial square roots are useful for factoring.

Discrete log

1. Pick a that is relatively prime with N.
2. Efficient to test if relatively prime by finding GCD using Euclid's algorithm.
   For example, a=6 and n=15.

Exercise: list the possible a's for $N = 15$.