



WIKIPEDIA
The Free Encyclopedia

[Main page](#)

[Contents](#)

[Featured content](#)

[Current events](#)

[Random article](#)

[Donate to Wikipedia](#)

[Wikipedia store](#)

Interaction

[Help](#)

[About Wikipedia](#)

[Community portal](#)

[Recent changes](#)

[Contact page](#)

Tools

[What links here](#)

[Related changes](#)

[Upload file](#)

[Special pages](#)

[Permanent link](#)

[Page information](#)

[Wikidata item](#)

[Cite this page](#)

Print/export

[Create a book](#)

[Download as PDF](#)

[Printable version](#)

Languages

[Deutsch](#)

[Español](#)

[Français](#)

[עברית](#)

[Lietuvių](#)

[Polski](#)

[Português](#)

[Русский](#)

[Suomi](#)

[Українська](#)

[Tiếng Việt](#)

[Edit links](#)

Article [Talk](#)

[Read](#)

[Edit](#)

[View history](#)

Deutsch–Jozsa algorithm

From Wikipedia, the free encyclopedia

(Redirected from [Deutsch-Jozsa algorithm](#))

The **Deutsch–Jozsa algorithm** is a [quantum algorithm](#), proposed by [David Deutsch](#) and [Richard Jozsa](#) in 1992^[1] with improvements by [Richard Cleve](#), [Artur Ekert](#), Chiara Macchiavello, and [Michele Mosca](#) in 1998.^[2] Although of little practical use, it is one of the first examples of a quantum algorithm that is exponentially faster than any possible deterministic classical algorithm. It is also a [deterministic algorithm](#), meaning that it always produces an answer, and that answer is always correct.

Contents [\[hide\]](#)

[1 Problem statement](#)

[2 Motivation](#)

[3 Classical solution](#)

[4 History](#)

[5 Decoherence](#)

[6 Deutsch's Algorithm](#)

[7 References](#)

[8 External links](#)

Problem statement [\[edit\]](#)

In the Deutsch–Jozsa problem, we are given a black box quantum computer known as an [oracle](#) that implements the function $f : \{0, 1\}^n \rightarrow \{0, 1\}$. In layman's terms, it takes n-digit binary values as input and produces either a 0 or a 1 as output for each such value. We are [promised](#) that the function is either [constant](#) (0 on all inputs or 1 on all inputs) or *balanced*^[3] (returns 1 for half of the input [domain](#) and 0 for the other half); the task then is to determine if *f* is constant or balanced by using the oracle.

Motivation [\[edit\]](#)

The Deutsch–Jozsa problem is specifically designed to be easy for a quantum algorithm and hard for any deterministic classical algorithm. The motivation is to show a black box problem that can be solved efficiently by a quantum computer with no error, whereas a deterministic classical computer would need exponentially many queries to the black box to solve the problem. More formally, it yields an oracle relative to which [EQP](#), the class of problems that can be solved exactly in polynomial time on a quantum computer, and **P** are different.

Since the problem is easy to solve on a probabilistic classical computer, it does not yield an oracle separation with [BPP](#), the class of problems that can be solved with bounded error in polynomial time on a probabilistic classical computer. [Simon's problem](#) is an example of a problem that yields a separation between **BQP** and **BPP**.

Classical solution [\[edit\]](#)

For a conventional [deterministic](#) algorithm where *n* is number of bits, $2^{n-1} + 1$ evaluations of *f* will be required in the worst case. To prove that *f* is constant, just over half the set of inputs must be evaluated and their outputs found to be identical (remembering that the function is guaranteed to be either balanced or constant, not somewhere in between). The best case occurs where the function is balanced and the first two output values that happen to be selected are different. For a conventional [randomized algorithm](#), a constant *k* evaluations of the function suffices to produce the correct answer with a high probability (failing with probability $\epsilon \leq 1/2^{k-1}$). However, $k = 2^{n-1} + 1$ evaluations are still required if we want an answer that is always correct. The Deutsch–Jozsa quantum algorithm produces an answer that is always correct with a single evaluation of *f*.

History [\[edit\]](#)

The Deutsch–Jozsa Algorithm generalizes earlier (1985) work by David Deutsch, which provided a solution for the simple case.

Specifically we were given a boolean function whose input is 1 bit, $f : \{0,1\} \rightarrow \{0,1\}$ and asked if it is constant.^[4]

The algorithm as Deutsch had originally proposed it was not, in fact, deterministic. The algorithm was successful with a probability of one half. In 1992, Deutsch and Jozsa produced a deterministic algorithm which was generalized to a function which takes n bits for its input. Unlike Deutsch's Algorithm, this algorithm required two function evaluations instead of only one.

Further improvements to the Deutsch–Jozsa algorithm were made by Cleve et al.,^[2] resulting in an algorithm that is both deterministic and requires only a single query of f . This algorithm is still referred to as Deutsch–Jozsa algorithm in honour of the groundbreaking techniques they employed.^[2]

The Deutsch–Jozsa algorithm provided inspiration for [Shor's algorithm](#) and [Grover's algorithm](#), two of the most revolutionary quantum algorithms.^{[5][6]}

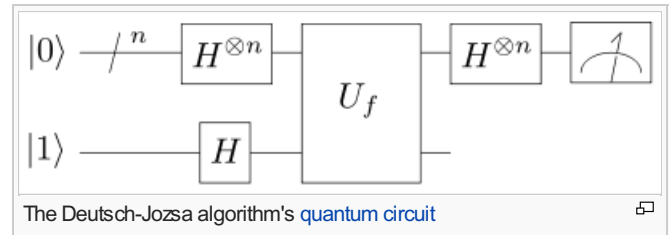
Decoherence [\[edit\]](#)

For the Deutsch–Jozsa algorithm to work, the oracle computing $f(x)$ from x has to be a quantum oracle which doesn't decohere x . It also mustn't leave any copy of x lying around at the end of the oracle call.

The algorithm begins with the $n+1$ bit state $|0\rangle^{\otimes n}|1\rangle$. That is, the first n bits are each in the state $|0\rangle$ and the final bit is $|1\rangle$. A

[Hadamard transformation](#) is applied to each bit to obtain the state

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



We have the function f implemented as quantum oracle. The oracle maps the state $|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$, where \oplus is addition modulo 2 (see below for details of implementation). Applying the quantum oracle gives

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle(|f(x)\rangle - |1 \oplus f(x)\rangle).$$

For each x , $f(x)$ is either 0 or 1. A quick check of these two possibilities yields

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle(|0\rangle - |1\rangle).$$

At this point the last qubit may be ignored. We apply a [Hadamard transformation](#) to each qubit to obtain

$$\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \left[\sum_{y=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle \right] = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \left[\sum_{x=0}^{2^n-1} (-1)^{f(x)} (-1)^{x \cdot y} \right] |y\rangle$$

where $x \cdot y = x_0y_0 \oplus x_1y_1 \oplus \dots \oplus x_{n-1}y_{n-1}$ is the sum of the bitwise product.

Finally we examine the probability of measuring $|0\rangle^{\otimes n}$,

$$\left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2$$

which evaluates to 1 if $f(x)$ is constant ([constructive interference](#)) and 0 if $f(x)$ is balanced ([destructive interference](#)).

Deutsch's Algorithm [\[edit\]](#)

Deutsch's algorithm is a special case of the general Deutsch–Jozsa algorithm. We need to check the condition $f(0) = f(1)$. It is equivalent to check $f(0) \oplus f(1)$ (where \oplus is addition modulo 2, which can also be viewed as a quantum [XOR gate](#) implemented as a [Controlled NOT gate](#)), if zero, then f is constant, otherwise f is not constant.

We begin with the two-qubit state $|0\rangle|1\rangle$ and apply a [Hadamard transform](#) to each qubit. This yields

$$\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle).$$

We are given a quantum implementation of the function f that maps $|x\rangle|y\rangle$ to $|x\rangle|f(x) \oplus y\rangle$. Applying this function to our current state we obtain

$$\begin{aligned}
& \frac{1}{2}(|0\rangle(|f(0) \oplus 0\rangle - |f(0) \oplus 1\rangle) + |1\rangle(|f(1) \oplus 0\rangle - |f(1) \oplus 1\rangle)) \\
&= \frac{1}{2}((-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle) + (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle)) \\
&= (-1)^{f(0)}\frac{1}{2}(|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle)(|0\rangle - |1\rangle).
\end{aligned}$$

We ignore the last bit and the global phase and therefore have the state

$$\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle).$$

Applying a Hadamard transform to this state we have

$$\begin{aligned}
& \frac{1}{2}(|0\rangle + |1\rangle + (-1)^{f(0)\oplus f(1)}|0\rangle - (-1)^{f(0)\oplus f(1)}|1\rangle) \\
&= \frac{1}{2}((1 + (-1)^{f(0)\oplus f(1)})|0\rangle + (1 - (-1)^{f(0)\oplus f(1)})|1\rangle).
\end{aligned}$$

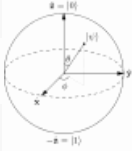
Obviously $f(0) \oplus f(1) = 0$ if and only if we measure a zero and $f(0) \oplus f(1) = 1$ if and only if we measure a one. So with certainty we know whether $f(x)$ is constant or balanced.

References [\[edit\]](#)

- ↑ David Deutsch and Richard Jozsa (1992). "Rapid solutions of problems by quantum computation". *Proceedings of the Royal Society of London A* **439**: 553. Bibcode:1992RSPSA.439..553D . doi:10.1098/rspa.1992.0167 .
- ↑ ^a ^b ^c R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca (1998). "Quantum algorithms revisited". *Proceedings of the Royal Society of London A* **454**: 339–354. arXiv:quant-ph/9708016 . Bibcode:1998RSPSA.454..339C . doi:10.1098/rspa.1998.0164 .
- ↑ Certainty from Uncertainty
- ↑ David Deutsch (1985). "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer" (PDF). *Proceedings of the Royal Society of London A* **400**: 97. Bibcode:1985RSPSA.400...97D . doi:10.1098/rspa.1985.0070 .
- ↑ Lov K. Grover (1996). *A fast quantum mechanical algorithm for database search*. Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. pp. 212–219. arXiv:quant-ph/9605043 . doi:10.1145/237814.237866 .
- ↑ Peter W. Shor (1994). *Algorithms for quantum computation: discrete logarithms and factoring* (PDF). Proceedings of the 35th IEEE Symposium on Foundations of Computer Science. pp. 124–134. doi:10.1109/SFCS.1994.365700 .

External links [\[edit\]](#)

- Deutsch's lecture about Deutsch algorithm
- Implementation of the Deutsch-Jozsa algorithm in the Scala programming language

v · t · e	Quantum information science		[hide]
General	Quantum computer · Qubit · Quantum information · Quantum programming · Timeline of quantum computing		
Quantum communication	Quantum capacity · Classical capacity · Entanglement-assisted classical capacity · Quantum channel (Quantum network) · Quantum cryptography (Quantum key distribution) · Quantum energy teleportation · Quantum teleportation · Superdense coding · LOCC · Entanglement distillation		
Quantum algorithms	Universal quantum simulator · Deutsch–Jozsa algorithm · Grover's algorithm · Quantum Fourier transform · Shor's algorithm · Simon's problem · Quantum phase estimation algorithm · Quantum annealing · Algorithmic cooling		
Quantum complexity theory	Quantum Turing machine · BQP · QMA · PostBQP		
Quantum computing models	Quantum circuit (Quantum gate) · One-way quantum computer (cluster state) · Adiabatic quantum computation · Topological quantum computer		
Decoherence prevention	Quantum error correction · Stabilizer codes · Entanglement-Assisted Quantum Error Correction · Quantum convolutional codes		
	Quantum optics	Cavity QED · Circuit QED · Linear optical quantum computing	
	Ultracold atoms	Trapped ion quantum computer · Optical lattice	

Physical implementations	Spin-based	Nuclear magnetic resonance QC · Kane QC · Loss–DiVincenzo QC · Nitrogen-vacancy center
	Superconducting quantum computing	Charge qubit · Flux qubit · Phase qubit

Categories: Quantum algorithms

This page was last modified on 26 August 2015, at 01:31.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

