Quantum computing for cryptographers IV

Gover's algorithm

Tanja Lange
idea and design by Daniel J. Bernstein

Eindhoven University of Technology

SAC – Post-quantum cryptography

Grover's algorithm

Grover's algorithm is often described as a search in unstructured data, but it does need a function that can capture this search.

Let

$$f : \{0,1\}^n \to \{0,1\}.$$

This need not be a function from $\mathbf{F}_2^n$.

Assume: unique $s \in \{0,1\}^n$ has $f(s) = 0$.

Traditional algorithm to find s:
compute f for many inputs, hope to find output 0.

Success probability is very low until #inputs approaches 2^n.

Grover's algorithm

Grover's algorithm is often described as a search in unstructured data, but it does need a function that can capture this search.

Let

$$f : \{0,1\}^n \to \{0,1\}.$$

This need not be a function from $\mathbf{F}_2^n$.

Assume: unique $s \in \{0,1\}^n$ has $f(s) = 0$.

Traditional algorithm to find s:
compute f for many inputs, hope to find output 0.

Success probability is very low until #inputs approaches 2^n.

Grover's algorithm takes only $2^{n/2}$ reversible computations of f.
Typically: reversibility overhead is small enough that this easily beats traditional algorithm.

Grover's algorithm in action

Start from uniform superposition
over all n-bit strings u.

Grover's algorithm in action

Start from uniform superposition
over all n-bit strings u.

Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Grover's algorithm in action

Start from uniform superposition
over all n-bit strings u.

Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Grover's algorithm in action

Start from uniform superposition
over all n-bit strings u.

Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2
about $0.58 \cdot 2^{0.5n}$ times.

Grover's algorithm in action

Start from uniform superposition
over all n-bit strings u.

Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2
about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

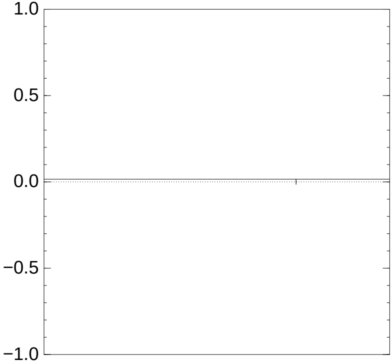
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after Step 1:



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

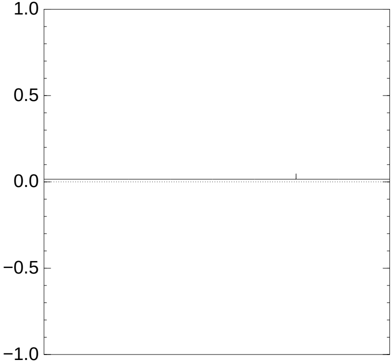
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after Step 1 + Step 2:



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

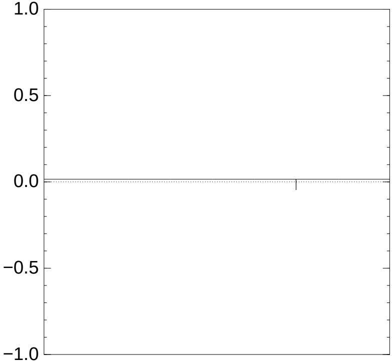
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after Step 1 + Step 2 + Step 1:



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

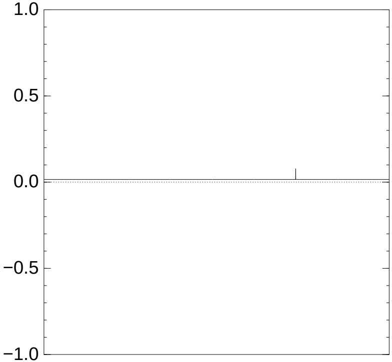
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $2 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

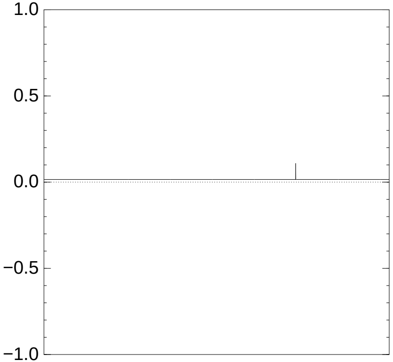
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $3 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

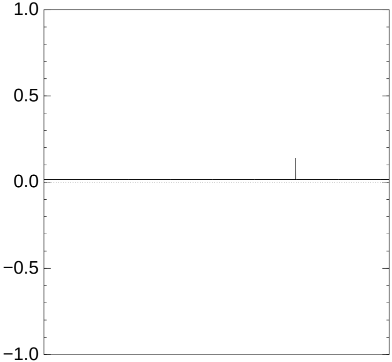
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $4 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

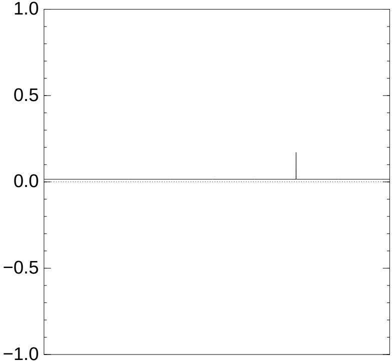
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $5 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

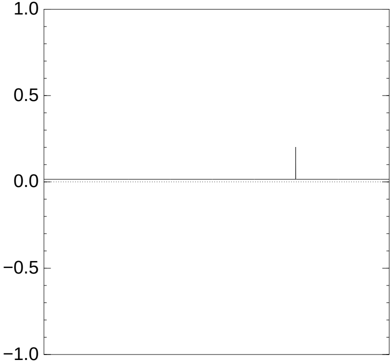
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $6 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

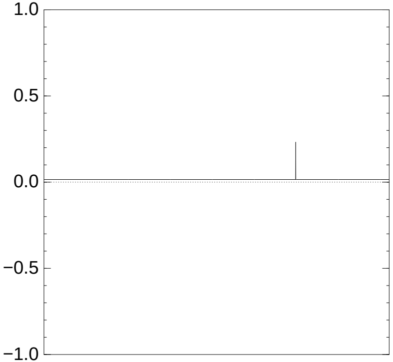
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $7 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

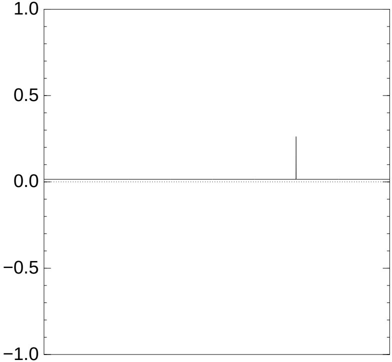
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $8 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

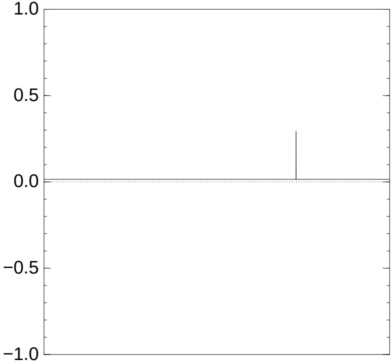
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $9 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

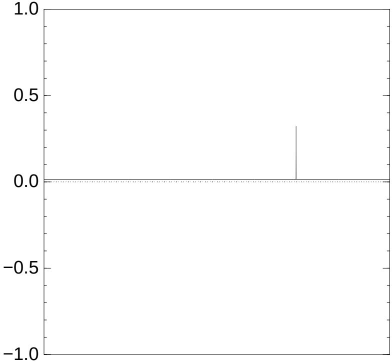
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $10 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

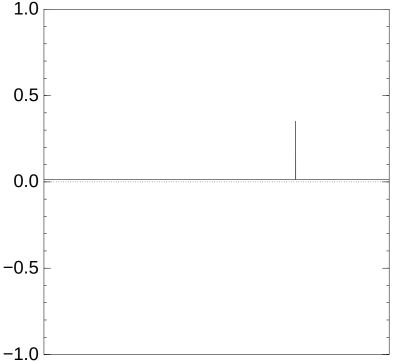
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $11 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

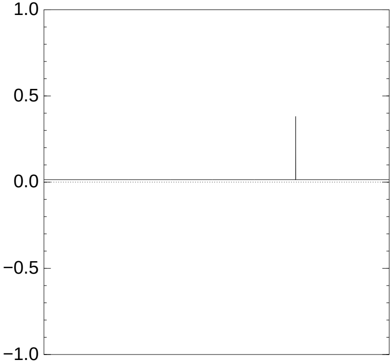
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $12 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

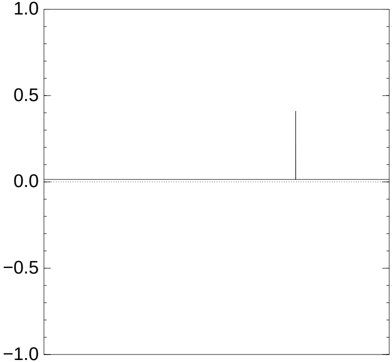
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $13 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

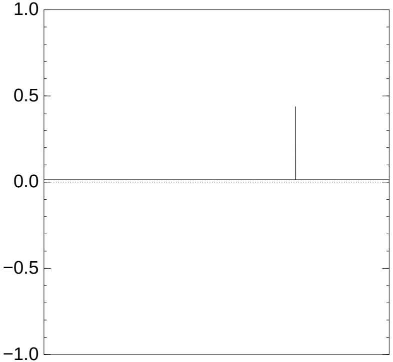
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $14 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

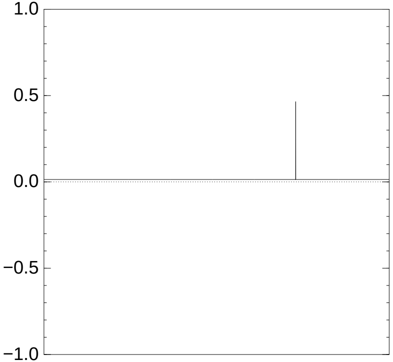
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $15 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

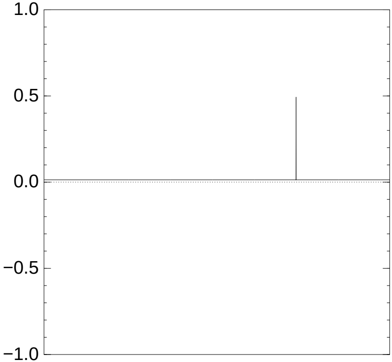
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $16 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

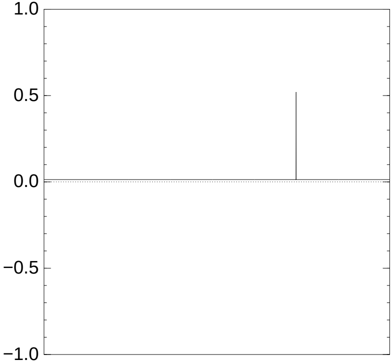
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2
about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $17 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

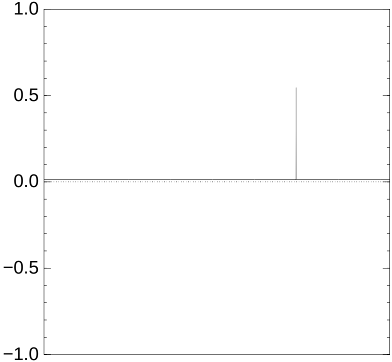
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $18 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

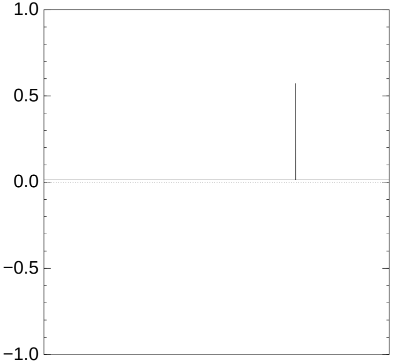
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $19 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

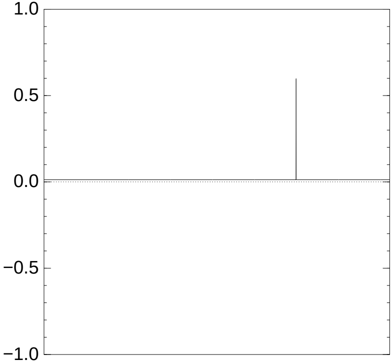
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2
about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$
for an example with $n = 12$
after $20 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

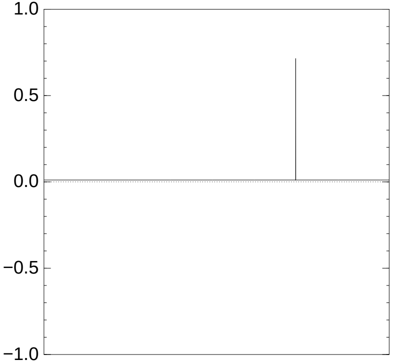
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $25 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

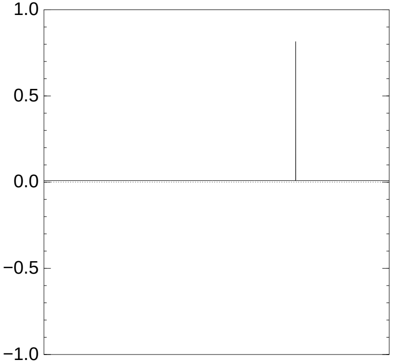
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $30 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

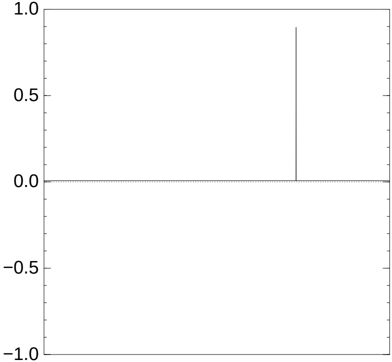
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $35 \times$ (Step 1 + Step 2):



Good moment to stop, measure.

Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

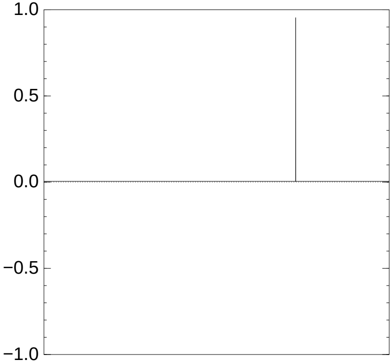
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $40 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

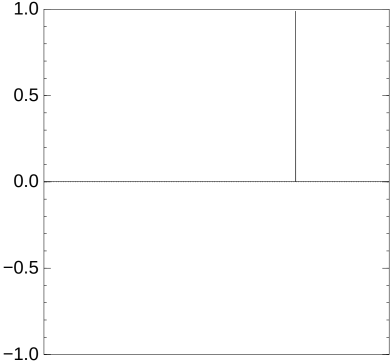
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $45 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

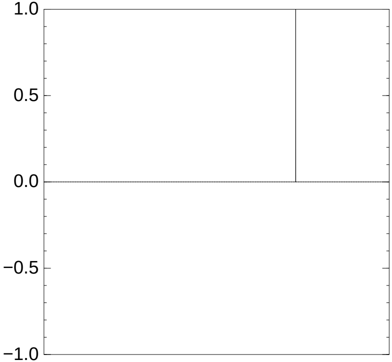
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $50 \times$ (Step 1 + Step 2):



Traditional moment to stop measure.

Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

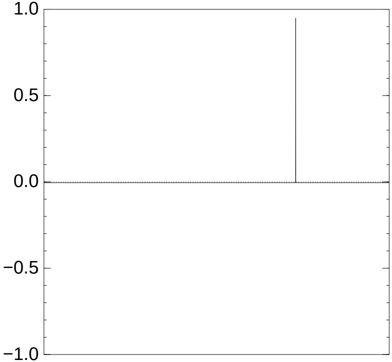
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $60 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

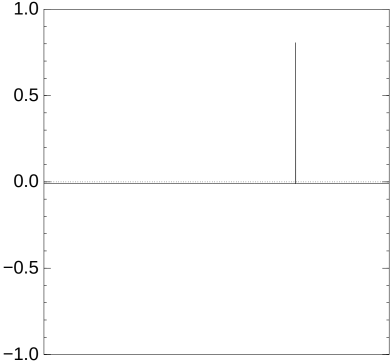
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $70 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

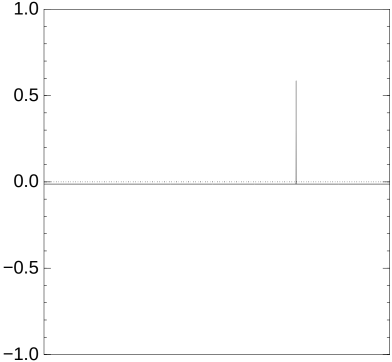
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $80 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

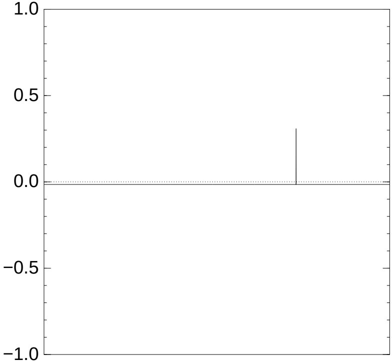
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $90 \times$ (Step 1 + Step 2):



Grover's algorithm in action

Start from uniform superposition over all n-bit strings u.

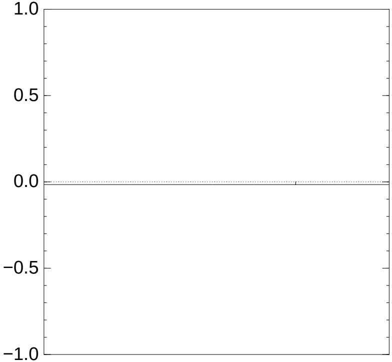
Step 1: Set $a \leftarrow b$ where
$b_u = -a_u$ if $f(u) = 0$,
$b_u = a_u$ otherwise.
This is fast.

Step 2: "Grover diffusion".
Negate a around its average.
This is also fast.

Repeat Step 1 + Step 2 about $0.58 \cdot 2^{0.5n}$ times.

Measure the n qubits.
With high probability this finds s.

Normalized graph of $u \mapsto a_u$ for an example with $n = 12$ after $100 \times$ (Step 1 + Step 2):



Very bad stopping point.