CS 599 P1: Introduction to Quantum Computation | Boston University, Fall 2025
Instructor: Alexander Poremba | **Scribe:** Logan Grout

LECTURE #13: GROVER'S ALGORITHM

Grover's algorithm is a celebrated quantum algorithm, discovered by Lov Grover in 1996. It provides a *quadratic speed-up* over any classical algorithm for searching an unstructured database or solving a general unstructured search problem. Although the improvement from $O(N)$ to $O(\sqrt{N})$ might appear modest compared to the exponential speed-up of Shor's algorithm, Grover's method is widely applicable. For example, suppose that a classical search problem requires 1 billion seconds ($\approx$ 31 years) to solve. Then, Grover's quantum algorithm would only require about 9 hours—a dramatic improvement!

Grover's algorithm forms the basis of numerous quantum subroutines (for example, amplitude amplification) and provides the best possible quantum speed-up for black-box search problems.

1 Problem Definition

Grover's algorithm addresses one of the most fundamental computational tasks: searching for a marked item in an otherwise unstructured database. This is called the *unstructured search problem*.

Definition: Unstructured Search Problem

Given (oracle) access to a Boolean function

$$f : \{0, 1, \dots, N-1\} \to \{0, 1\},$$

with the promise that there exists a unique element $x^* \in \{0, 1, \dots, N-1\}$ satisfying $f(x^*) = 1$, and $f(x) = 0$, otherwise, the goal is to *find* the marked item x^*.

Classical vs Quantum Complexity:

- **Classical:** In the worst case, a classical algorithm must query $f(x)$ for nearly all N possible inputs before finding x^*. Therefore, the expected number of oracle calls required is $\Omega(N)$.
- **Quantum:** Grover's algorithm achieves a quadratic speedup, requiring only $O(\sqrt{N})$ queries to the quantum oracle U_f. Each query evaluates $f(x)$ coherently across all N possibilities simultaneously by exploiting quantum superposition and interference.

Question: *Where does the oracle f come from?*

In most realistic settings, f is not an externally provided black box, but rather a function we can implement ourselves. Typically, $f(x)$ encodes a computational task we wish to solve, such as checking whether a candidate solution satisfies certain constraints. Because both classical and quantum algorithms must evaluate $f(x)$, the total runtime of Grover's algorithm is meaningful only when evaluating f can be done efficiently

— ideally in $\text{poly}(\log N)$ or $\text{poly}(n)$ time, where $N = 2^n$. Importantly, just because we know how to implement f efficiently, does not mean it is easy to find a desired marked item! See the following example:

Example: Boolean Satisfiability (SAT)

Consider a Boolean formula

$$\varphi(x_1, x_2, \ldots, x_n)$$

over n input variables. We wish to find an assignment $(x_1^*, \ldots, x_n^*) \in \{0,1\}^n$ that makes the formula evaluate to true.
This can be viewed as an instance of the unstructured search problem by defining the oracle

$$f(x) = \begin{cases} 1, & \text{if } \varphi(x) = \text{True}, \\ 0, & \text{otherwise.} \end{cases}$$

The search space has size $N = 2^n$. Assuming that evaluating $\varphi(x)$ can be done in polynomial time (in n), a classical exhaustive search requires $O(2^n)$ evaluations, while Grover's algorithm finds a satisfying assignment in expected time

$$O(\text{poly}(n)\sqrt{2^n}) = O(\text{poly}(n)\, 2^{n/2}).$$

Thus, Grover's algorithm provides a quadratic improvement over brute-force search, which can be significant for moderately large problem instances.

2 Phase Oracle

Before introducing Grover's algorithm, we must understand how a quantum computer can evaluate a classical Boolean function in superposition by computing it into the phase via a so-called *phase oracle*.

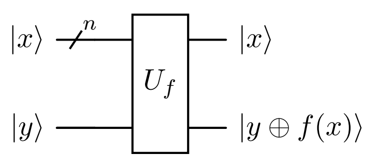


Figure 1: Standard oracle for f

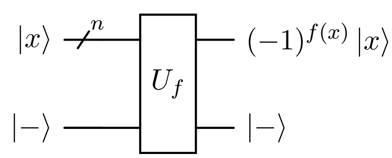


Figure 2: Phase oracle for f

In Grover's algorithm, it is convenient to use the *phase oracle*, which we can obtain from the standard oracle U_f by initializing the auxiliary qubit in the state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Then, applying U_f gives:

$$U_f |x\rangle |-\rangle = (-1)^{f(x)} |x\rangle |-\rangle .$$

The auxiliary qubit is unchanged and can be discarded, resulting in an effective operation

$$|x\rangle \mapsto (-1)^{f(x)} |x\rangle .$$

For the rest of the lecture, we will assume $N = 2^n$ and use U_f to simply denote the *phase oracle* and discard the ancillary $|-\rangle$ qubit. We therefore abuse notation and write

$$U_f : |x\rangle \mapsto (-1)^{f(x)} |x\rangle .$$

3 Grover's Algorithm

Grover's algorithm finds the unique marked element x^* such that $f(x^*) = 1$ using repeated applications of two reflections:

1. A reflection about the hyperplane defined by the oracle U_f, which flips the phase of $|x^*\rangle$.
2. A reflection about the uniform superposition state, implemented by the *diffusion operator* D.

Each iteration amplifies the amplitude of $|x^*\rangle$ while reducing that of all other basis states.

Grover's Algorithm

1. **Initialization:** Prepare the n-qubit uniform superposition

$$|\psi_0\rangle = |+^n\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle .$$

2. **Oracle Reflection:** Apply the phase oracle U_f:

$$U_f |x\rangle = \begin{cases} -|x\rangle , & \text{if } x = x^*, \\ \ \ |x\rangle , & \text{otherwise.} \end{cases}$$

   This inverts the sign of the amplitude corresponding to the marked item.

3. **Diffusion Reflection:** Apply the so-called diffusion operator D with

$$D = 2 |+^n\rangle\langle +^n| - I$$

   which reflects all amplitudes about their mean, thereby increasing the marked state's amplitude and slightly decreasing all others. It performs the ampltidude transformation

$$\alpha_x \mapsto 2\overline{\alpha} - \alpha_x, \qquad \text{where } \overline{\alpha} = \frac{1}{N} \sum_{x=0}^{N-1} \alpha_x .$$

4. **Iteration:** Repeat steps 2 and 3 approximately

$$t \approx \frac{\pi}{4}\sqrt{N}$$

   times. After these iterations, measuring the state yields x^* with high probability.

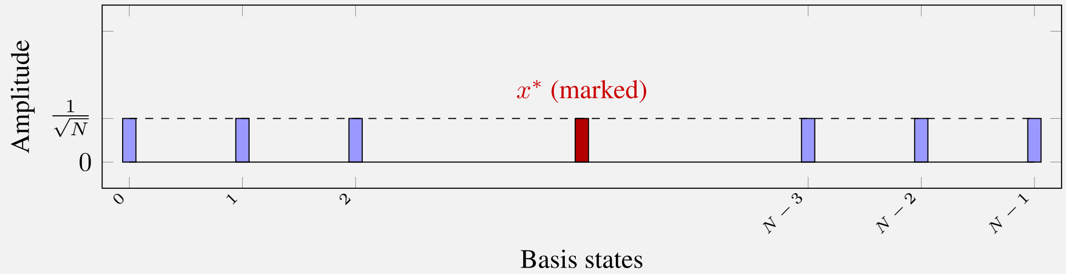
Let us now take a closer look at the Grover iteration works.

3.1 Visual illustration of a single Grover iteration

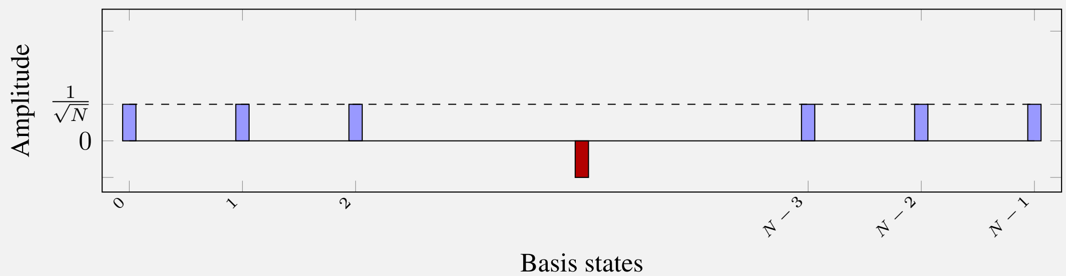
To visualize Grover's iteration, consider the following progression of amplitudes:

Illustration of the Grover Iteration

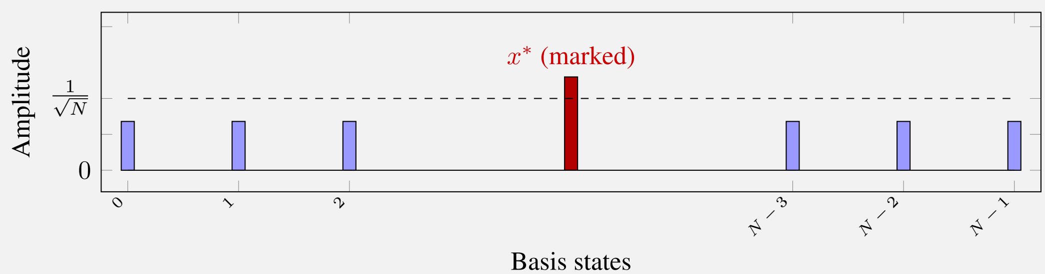
1. **Start with uniform superposition:** $|+^n\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$.



2. **Apply the phase oracle U_f:** The marked amplitude flips sign.



3. **Apply the diffusion operator D:** Reflection about the mean increases the marked amplitude.



4. **Repeat:** Each iteration of (U_f, D) further increases the amplitude of $|x^*\rangle$. After t iterations:

$$\alpha_{x^*} \approx \frac{2t+1}{\sqrt{N}}, \qquad \Pr[\text{measure } x^*] \approx \frac{(2t+1)^2}{N}.$$

Choosing $t = O(\sqrt{N})$ yields a measurement success probability close to 1.

3.2 The Diffusion Operator

The *diffusion operator* acts as a reflection about the uniform superposition:

$$D = 2\left|+^n\right\rangle\!\left\langle +^n\right| - I.$$

In the computational basis $\{|0\rangle, \ldots, |N-1\rangle\}$, we can express it as

$$D = \begin{pmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \cdots & \frac{2}{N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} - 1 \end{pmatrix}.$$

It performs the transformation

$$\alpha_x \mapsto 2\overline{\alpha} - \alpha_x, \qquad \text{where} \quad \overline{\alpha} = \frac{1}{N}\sum_{x=0}^{N-1} \alpha_x.$$

Thus, D *flips* every amplitude about their average value, amplifying those below the mean (including the marked one after phase inversion). The diffusion operator D can be implemented using the simple unitary operator $A = 2\left|0\right\rangle\!\left\langle 0\right| - I$ (which is later interleaved with Hadamards), as shown below.

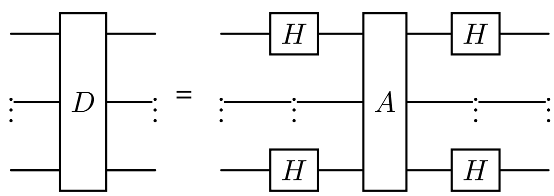


Figure 3: Implementation of the diffusion operator D.

3.3 Quantum Circuit for Grover's Algorithm

The full quantum circuit for Grover's algorithm is a direct translation of the iterative procedure we have already described conceptually. We begin with n qubits initialized to $|0\rangle^{\otimes n}$ and apply Hadamard gates to create the uniform superposition $|+^n\rangle$. Then, each *Grover iteration* consists of two main components:

- The **phase oracle** U_f, which flips the sign of the amplitude corresponding to the marked state x^*.
- The **diffusion operator** D, which reflects all amplitudes about their average, thereby amplifying the marked state's amplitude.

After roughly $O(\sqrt{N})$ repetitions of this sequence, the probability of measuring the marked state approaches 1. The following circuit illustrates this process:
The repeated (U_f, D) pairs form the **Grover iteration block**. The number of iterations required depends on the size of the search space N and the number of marked elements. For a single marked item, approximately $\lfloor \frac{\pi}{4}\sqrt{N} \rfloor$ iterations maximize the probability of measuring the correct result. Finally, a measurement in the computational basis yields the marked element x^* with high probability.

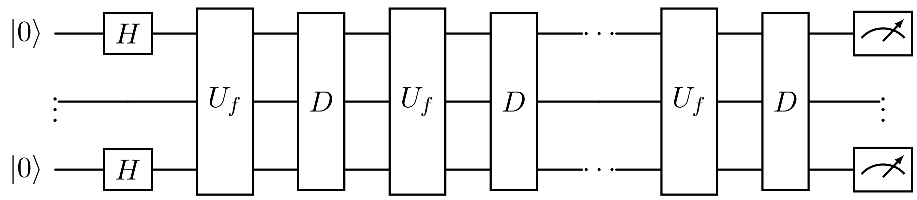


Figure 4: Quantum circuit for Grover's algorithm. Each iteration applies the oracle U_f followed by the diffusion operator D.

3.4 Geometric Interpretation

Grover's algorithm can be elegantly understood as a sequence of geometric rotations in a two-dimensional subspace. Although the search space consists of $N = 2^n$ computational basis states, all the action of the algorithm takes place within the plane spanned by the two orthogonal vectors:

$$|x^*\rangle\,, \qquad |\text{unmarked}\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq x^*} |x\rangle\,.$$

By construction, $\langle x^* | \text{unmarked}\rangle = 0$. The initial uniform superposition can therefore be expressed as

$$|+^n\rangle = \frac{1}{\sqrt{N}}\,|x^*\rangle + \sqrt{1 - \frac{1}{N}}\,|\text{unmarked}\rangle\,.$$

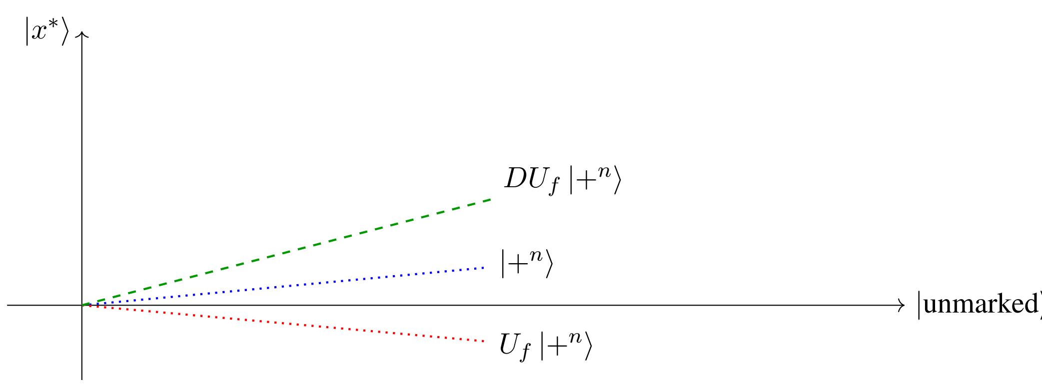


Figure 5: Geometric view of Grover's iteration as successive reflections about $|+^n\rangle$ and $|x^*\rangle$.

Each Grover iteration performs two reflections:

1. U_f reflects the current state about the $|\text{unmarked}\rangle$ axis (by flipping the phase of $|x^*\rangle$),
2. D reflects the result about the initial state $|+^n\rangle$.

The composition of two reflections yields a *rotation* by an angle 2θ within the plane $\text{span}\{|x^*\rangle\,, |\text{unmarked}\rangle\}$, where

$$\sin\theta = \frac{1}{\sqrt{N}}.$$

After t iterations, the state vector has rotated by an angle $(2t+1)\theta$ toward $|x^*\rangle$, and thus the probability of measuring the marked state is

$$\Pr[\text{measure } x^*] = \left| \langle x^*|(DU_f)^t|+^n\rangle \right|^2 = \sin^2\bigl((2t+1)\theta\bigr).$$

To maximize this probability, we choose

$$t \approx \frac{\pi}{4}\sqrt{N},$$

which brings the state vector very close to $|x^*\rangle$, ensuring that measurement yields the correct solution with high probability.

3.5 Multiple marked elements

Grover's algorithm also extends naturally to the case where there are K marked items:

$$\bigl|\{\, x \in \{0,1,\ldots,N-1\} : f(x) = 1 \,\}\bigr| = K.$$

Let $|M\rangle$ denote the uniform superposition over the marked states, and $|U\rangle$ denote that over the unmarked states. The state of the algorithm again evolves entirely within the two-dimensional subspace $\mathrm{span}\{|M\rangle, |U\rangle\}$.

In this case, the angle of rotation per iteration satisfies

$$\sin\theta = \sqrt{\frac{K}{N}},$$

and after t iterations, the success probability is

$$\Pr[\text{measured item is marked}] = \sin^2\bigl((2t+1)\theta\bigr) = \sin^2\!\left((2t+1)\sqrt{\frac{K}{N}}\right).$$

Thus, we require only

$$O\!\left(\sqrt{\frac{N}{K}}\right)$$

oracle calls to find a marked element with high probability.