# LESSON 5.4

## GROVER'S SEARCH ALGORITHM

Valter Uotila

# OUTLINE

# MOTIVATION & BACKGROUND

- Lov Grover [6] developed Grover's search algorithm in 1996
- Note that the title of the original paper is *A fast quantum mechanical algorithm for database search*
- Unlike Deutsch-Jozsa algorithm, Grover's algorithm has practical applications. It forms central part in many other quantum algorithms. More about this in the end.
- This presentation is based on [1, 8, 7]

# CLASSICAL SEARCH PROBLEM

**Problem**

Search an element from an unsorted list.

- On avarage we need to check $\frac{n}{2}$ elements from the list

# CLASSICAL SEARCH PROBLEM

**Problem**

Search an element from an unsorted list.

- On avarage we need to check $\frac{n}{2}$ elements from the list
- To find the element with 100% certainty, we need to check all the *n* elements

# CLASSICAL SEARCH PROBLEM

**Problem**

Search an element from an unsorted list.

- On avarage we need to check $\frac{n}{2}$ elements from the list
- To find the element with 100% certainty, we need to check all the $n$ elements
- Grover's algorithm enables us to find the element with $\sqrt{n}$ steps with high probability

# CLASSICAL SEARCH PROBLEM

**Problem**

Search an element from an unsorted list.

- On avarage we need to check $\frac{n}{2}$ elements from the list
- To find the element with 100% certainty, we need to check all the $n$ elements
- Grover's algorithm enables us to find the element with $\sqrt{n}$ steps with high probability
- This is not exponential speedup but polynomial

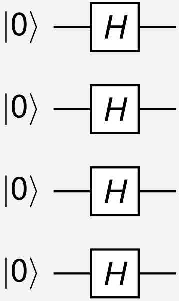
# INITIALIZATION

Initially we assume that we have a list of $2^k$ elements where one of the elements is marked. For example, when $k = 4$ we can have the following list and the marked element is 5:

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

# HADAMARD TRANSFORM

We are familiar with the Hadamard transform. When $k = 4$, we have the circuit

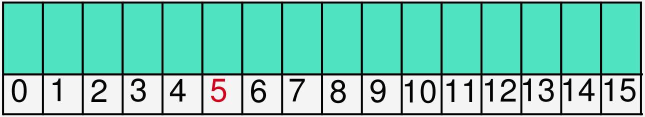


# HADAMARD TRANSFORM

The Hadamard transform creates an equal superposition between all the states. The states correspond to the elements of the list. For example, the marked element 5 corresponds to the state $|1010\rangle$ because the binary representation of 5 is 101.

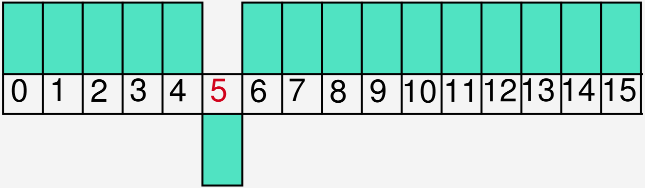
# HADAMARD TRANSFORM

Considering the list after Hadamard transform, the amplitudes corresponding to each element are now equal:



# AGAIN ORACLES

The idea is to create an oracle which flips the phase of the marked element. In the example, that means:



# NAIVE GROVER ORACLE FROM MATRIX

It is easy to implement the oracle in small cases if we deal with a quantum computer and software that can implement unitary operators based on their matrices. In the case $k = 4$, the oracle matrix is the identity matrix where the diagonal value for the marked element $x_{5,5}$ is changed to $-1$. This matrix selects the corresponding state and flips its phase. If $s$ is the marked element, the oracle is:

$$U_f = I - 2|s\rangle\langle s|.$$

# GROVER ORACLE

We can calculate that $U_f = I - 2|s\rangle\langle s|$ works as we want. First

$$U_f|s\rangle = (I - 2|s\rangle\langle s|)|s\rangle = |s\rangle - 2|s\rangle\langle s|s\rangle = -|s\rangle,$$

because the element's inner product $\langle s|s\rangle$ with itself is always 1. For states $|x\rangle \neq |s\rangle$ in the basis we have

$$U_f|x\rangle = (I - 2|s\rangle\langle s|)|x\rangle = |x\rangle - 2|s\rangle\langle s|x\rangle = |x\rangle,$$

because $\langle s|x\rangle = 0$ for any pair of different basis states. This shows that $U_f$ flips the phase of the marked element.

# GROVER ORACLE CONSTRUCTED FROM GATES

- In a real application, it is not practical to create $2^k$ sized matrices and build gates based on them, although the approach is simple and it gives some intuition
- Thus, we need to express Grover oracle with gates
- This requires one ancilla qubit
- As in the previous lessons, an ancilla qubit is prepared in state $|-\rangle$ because then $X|-\rangle = -|-\rangle$
- Then we use multi-control-CNOT operation

# GROVER ORACLE CONSTRUCTED FROM GATES: EXAMPLE

Now $5 = 1010_2$. Thus we can map it to the binary element 1111 by applying X-gate to the second and last qubit. Because now the marked element is 1111 in the changed basis, we can apply the oracle matrix where $-1$ is in right corner:

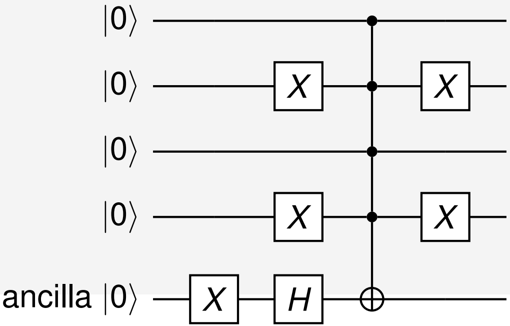
$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & \dots & 1 & 0 \\ 0 & \dots & 0 & -1 \end{bmatrix}$$

This matrix is easy to implement with the following circuit.

# GROVER ORACLE CONSTRUCTED FROM GATES: EXAMPLE

Grover oracle as circuit in the case $k = 4$ and the searched element is 5:

$U_f =$



# GROVER ORACLE CONSTRUCTED FROM GATES

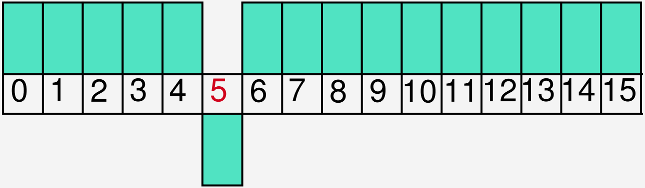
- If we are precise, multi-control-CNOT is not a standard gate either and it could be decocomposed into more fundamental gates (T-gates and CNOTs)

# GROVER ORACLE CONSTRUCTED FROM GATES

- If we are precise, multi-control-CNOT is not a standard gate either and it could be decocomposed into more fundamental gates (T-gates and CNOTs)
- Anyway, these decompositions are not necessarily relevant in order to understand the core idea of Grover's algorithm

# AMPLITUDE AMPLIFICATION

All in all, after applying the oracle, the list is in the state:



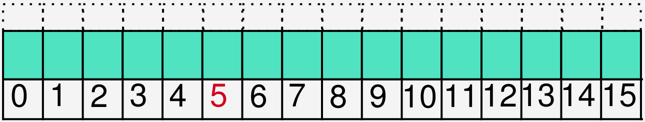

Let $|\varphi\rangle = H^{\otimes n}|0\rangle$ be the uniform superposition.

# AMPLITUDE AMPLIFICATION

Since every element in the list is represented as a vector in $2^4$ dimensional space, we can divide the system into two parts:

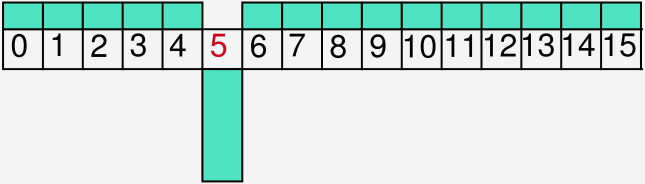
1. part proportional to $|\varphi\rangle$ and
2. part orthogonal to $|\varphi\rangle$.

The proportional part is



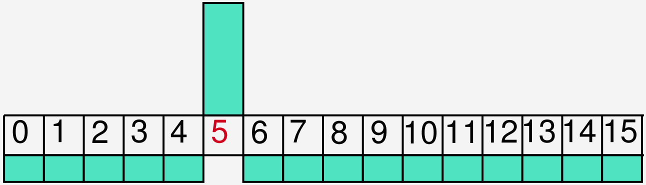
# AMPLITUDE AMPLIFICATION

The orthogonal part is

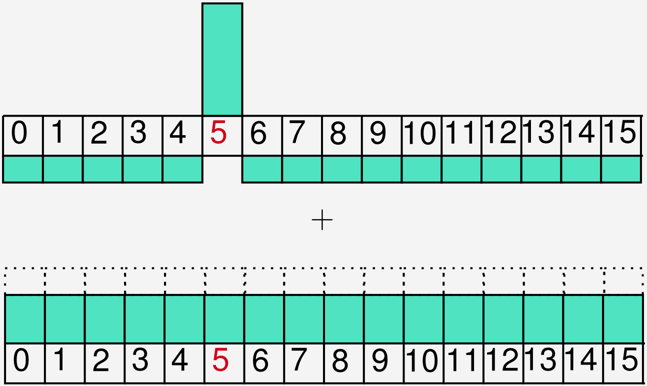


# AMPLITUDE AMPLIFICATION

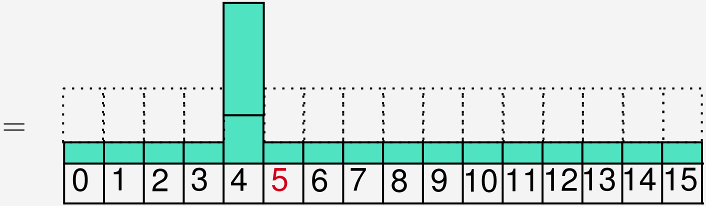
The flipped orthogonal part is



# AMPLITUDE AMPLIFICATION



# AMPLITUDE AMPLIFICATION




When we measure, the probability of measuring the element 5 is the highest. What operators could perform this amplitude amplification?

# DIFFUSION OPERATOR

- Recall that the oracle in Grover's algorithm flips the amplitude of the searched element
- Now we want something which leaves the uniform superposition alone, but flips the sign of ”everything else”, i.e., the states orthogonal to the uniform superposition

The option for this is

$$D = 2|\varphi\rangle\langle\varphi| - I,$$

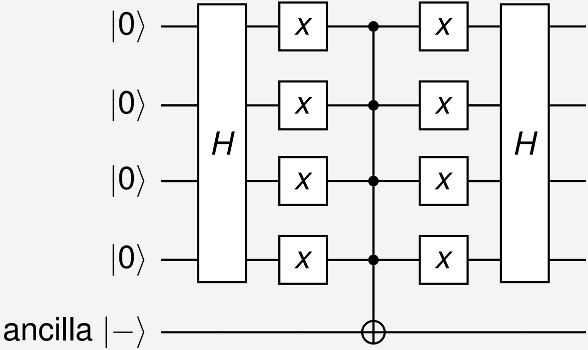
where $|\varphi\rangle$ is the uniform superposition with a single flipped amplitude.

# DIFFUSION OPERATOR

- Similar calculations as we did for oracle $U_f$ show that $D|\varphi\rangle = |\varphi\rangle$ and $D|\psi\rangle = -|\psi\rangle$ for any state $|\psi\rangle$ orthogonal to $|\varphi\rangle$. This shows that $D$ has the wanted effect to the states.
- How do we implement $D$ as a circuit? We know how to implement $U_f$ and we note that $-D = I - 2|\varphi\rangle\langle\varphi|$ looks very similar to $U_f$

# CIRCUIT FOR DIFFUSION OPERATOR

$-D =$



# CIRCUIT FOR DIFFUSION OPERATOR

Recall that we set $|\varphi\rangle = H^{\otimes n}|0\rangle$. Now the circuit in the previous slide first maps

$$|\varphi\rangle = H^{\otimes n}|0\rangle \mapsto H^{\otimes n}H^{\otimes n}|0\rangle = |0\rangle$$

The multi-control-CNOT gate circled with NOTs is triggered when it gets the state $|0\rangle$. Thus it will apply the *NOT* gate to the ancilla qubit: $X|-\rangle = -|-\rangle$. That introduces the flip to the phase.

# CIRCUIT FOR DIFFUSION OPERATOR

On the other hand, if the state in the beginning of the circuit is orthogonal to $|\varphi\rangle$, say $|\psi\rangle$, then the inner product is

$$\langle\varphi|H^{\otimes n}H^{\otimes n}|\psi\rangle = \langle\varphi|\psi\rangle = 0.$$

Thus the state $|\psi\rangle$ is orthogonal also after applying the Hadamard transform. Then some of the controls in the multi-control-CNOT are false, and the NOT gate is not triggered. Finally, the second Hadamard transform returns the query register to its state before the transformation.

# GROVER OPERATOR

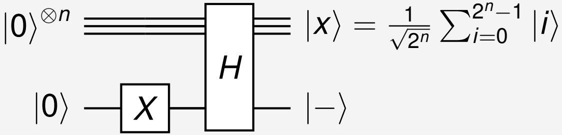
Now the Grover operator is a composition of the oracle $U_f$ and the diffusion operator $D$:

$$G = DU_f.$$

Now we can write the whole algorithm.

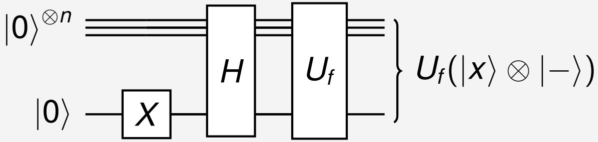
# CONSTRUCTING CIRCUIT: INITIALIZATION

The beginning of the circuit is very similar to Deutsch-Josza algorithm:




In the previous lesson we calculated why the circuit produces the state $\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle|-\rangle$.

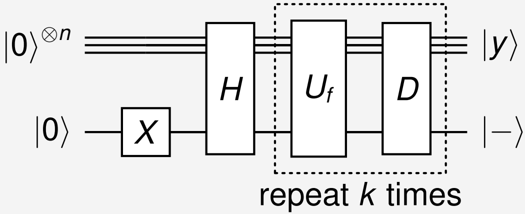
# CONSTRUCTING CIRCUIT: ORACLE




where $U_f(|x\rangle \otimes |-\rangle) = (-1)^{f(x)}|x\rangle \otimes |-\rangle$. The function $f(x) = 1$ if $x$ is the element we are searching and otherwise $f(x) = 0$.

Again the reasoning is similar to the case of the Deutsch-Jozsa algorithm.

# CONSTRUCTING CIRCUIT: FULL ALGORITHM



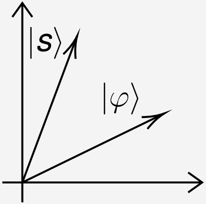

We will discuss later the optimal value for $k$. After applying Grover operator for suitably many times, we measure the first $n$ qubits. This should return the correct answer with high probability.

# GEOMETRIC INTERPRETATION

We can reason the optimal value for $k$ with a geometric argument. Because quantum states are linear combinations of vectors in high dimensional Hilbert space, we can visualize how the Grover operator $G = DU_f$ maps the states. We start studying the uniform superposition $|\varphi\rangle = H^{\otimes n}|0\rangle$ and the solution state $|s\rangle$. The idea is that the uniform superposition $|\varphi\rangle$ is the initial state where the algorithm starts and $|s\rangle$ is the state whose probability we want to maxime by applying Grover operator suitably many times.

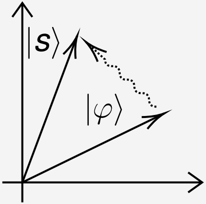
# GEOMETRIC INTERPRETATION

Visually (following example in [1])



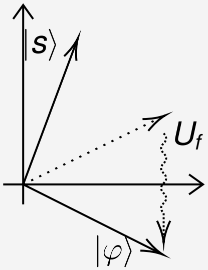
# GEOMETRIC INTERPRETATION

We aim to move the uniform superposition close to the solution



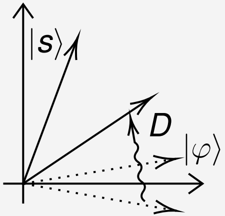
# GEOMETRIC INTERPRETATION

The Grover oracle flips the uniform superposition to the other side of one of the axis



# GEOMETRIC INTERPRETATION

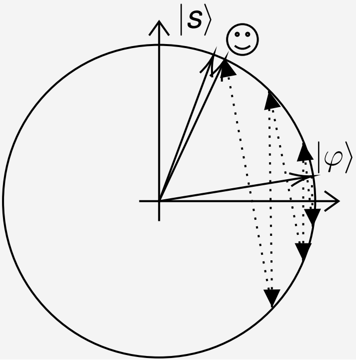
The diffusion operator flips the state to the other side of the initial uniform superposition




Now we see that the output vector after a single application of the Grover operator has moved the vector closer to the solution.

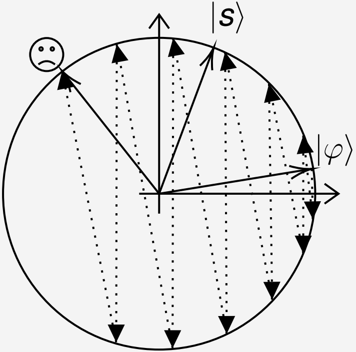
# GEOMETRIC INTERPRETATION

When we repeat the process suitably we can get close to the solution



# GEOMETRIC INTERPRETATION

If we repeat too many times, we start getting further from the solution



# WHAT IS OPTIMAL NUMBER OF GROVER OPERATORS?

The optimal number of Grover operators is

$$\frac{\pi}{2}\sqrt{N},$$

where $N = 2^n$ and $n$ is the length of the bit strings. The reason for this number can be deduced from this geometrical setting [1] but it requires a bit more mathematical machinery.

# HOW MUCH FASTER IS ALGORITHM

- Our classical algorithms work in $\mathcal{O}(N)$ time
- Grover search requires approximately $\sqrt{N}$ Grover operations and thus the time is $\mathcal{O}(\sqrt{N})$
- This is not exponential but quadratic speedup

# DEMONSTRATIONS

- Grover search on Quirk
- Grover search using Qiskit
- Grover search using Pennylane

# GROVER IN PRACTICE

Some selected papers from Quantum Algorithm Zoo:
https://quantumalgorithmzoo.org/

- Grover Adaptive Search for Constrained Polynomial Binary Optimization [4]
- Speedup Shor's algorithm: Factoring Safe Semiprimes with a Single Quantum Query [5]
- 3-SAT [2]
- Network flows [3]
- String matching [9]

# REFERENCES I


[1] Xanadu quantum codebook - learn quantum computing interactively online with pennylane, 2022.

[2] A. Ambainis.
Quantum search algorithms.
2005.

[3] A. Ambainis and R. Spalek.
Quantum algorithms for matching and network flows, 2005.

[4] A. Gilliam, S. Woerner, and C. Gonciulea.
Grover adaptive search for constrained polynomial binary optimization.
*Quantum*, 5:428, apr 2021.

[5] F. Grosshans, T. Lawson, F. Morain, and B. Smith.
Factoring safe semiprimes with a single quantum query, 2015.

# REFERENCES II


[6] L. K. Grover.
A fast quantum mechanical algorithm for database search.
*arXiv e-prints*, pages quant–ph/9605043, May 1996.

[7] D. Koch, L. Wessing, and P. M. Alsing.
Introduction to coding quantum algorithms: A tutorial series using qiskit.
*arXiv:1903.04359 [quant-ph]*, Mar 2019.
arXiv: 1903.04359.

[8] C. Lectures.
*A practical introduction to quantum computing - Elias Fernandez-Combarro Alvarez - (4/7)*.
Feb 2020.

[9] H. Ramesh and V. Vinay.
String matching in $\tilde{o}(\sqrt{n} + \sqrt{m})$ quantum time, 2000.