

CS290A/Phys250, Spring 2007:

# Quantum Information and Quantum Computation

Wim van Dam

Harold Frank Hall, Room 2151

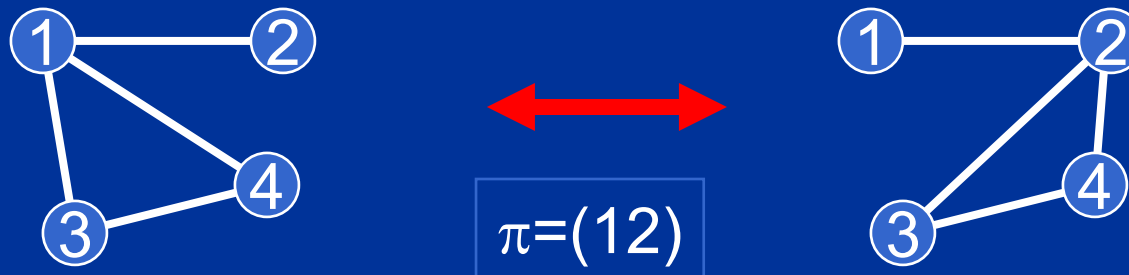
vandam@cs

[http://www.cs.ucsb.edu/~vandam/teaching/s07\\_CS290/](http://www.cs.ucsb.edu/~vandam/teaching/s07_CS290/)

# Q: Graph Isomorphism

Graphs  $G_1$  and  $G_2$ , are they isomorphic?

Example:



“Does there exist a  $\pi \in S_n$  such that  $\pi(G_1) = G_2$ ?”

When two graphs are isomorphic, then there is a short “certificate” that proves this fact.

But what to do when  $G_1$  and  $G_2$  are not isomorphic?

# A: Arthur - Merlin Proofs

Using interaction, an omnipotent prover Merlin can convince a limited verifier Arthur that two graphs are not isomorphic.

Sketch:

Arthur produces a sequence (of modest length) of graphs, where each graph is a randomly permuted version of  $G_1$  or  $G_2$ . He shows this sequence to Merlin and demands that Merlin tells him which graphs are permutations of  $G_1$  and which ones of  $G_2$ .

If and only if  $G_1$  and  $G_2$  are not isomorphic, Merlin will be capable of doing so. If  $G_1 \cong G_2$ , Merlin will make a mistake half the time.

If Merlin is always right, Arthur will get convinced very quickly that the graphs are indeed not isomorphic.

# The Deal with NP(-complete)

**NP** (Nondeterministic Polynomial time) is the class of decision problems for which the “Yes” instances have short, and easily verifiable proofs (easily = polynomial time in input size).

Examples:  $\{ n \in \mathbf{N} \text{ is not prime} \}$  and  $\{ f(x)=0 \text{ for a } x \in \{0,1\}^n \}$ .

Optimization problems are often related to **NP** problems.

Example: The Traveling Salesman Problem (TSP) can be solved if we can solve the decision problems of the kind “Does ... have a route that visits all cities, with the length of the trip  $\leq L$ ?”

**NP-complete** problems are those **NP** problems (example: TSP) whose solution would enable us to solve any other **NP** problem.

# Why not $BQP=NP$ ?

Why do we not think that quantum computers can solve all of **NP**?

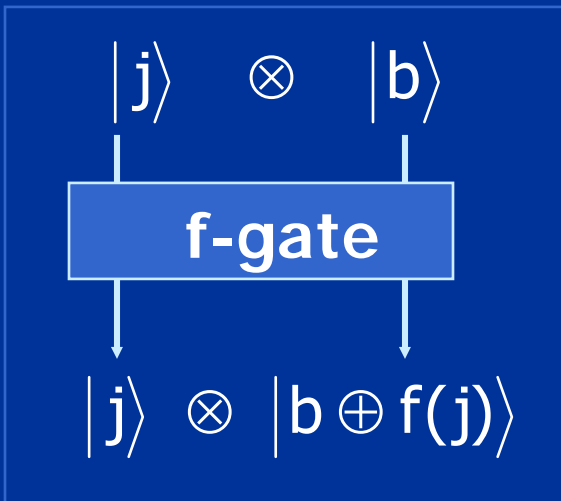
If we could quantum search arbitrary functions  $F:\{0,1\}^n\rightarrow\{0,1\}$  for solutions  $x\in\{0,1\}^n$  with  $F(x)=1$  in time  $\text{poly}(n)$ , then we would indeed have  $NP\subseteq BQP$ .

We know however that that is not possible.

The best way we can quantum search a database of size  $N$ , is by using Grover's algorithm, which has the optimal query and time complexity of  $\Theta(\sqrt{N})$ .

# Quantum Querying Functions

We assume that we have the network component:



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

There are things about  $F$  that

- are hidden,
- we can assume,
- we want to know,
- we are not interested in.

**We want to minimize the queries to the f-box when solving problems.**

# General Set Up

Consider a function  $f:\{0,\dots,N-1\} \rightarrow \{0,1\}$ .

We want to know some properties of  $f(0),\dots,f(N-1)$  where  $f$  is easy to compute, but  $N$  is too big.

**General Quantum Approach:** By calculating superpositions of  $f(j)$  values:

$$\sum_{j=0}^{N-1} a_j |j, b\rangle \mapsto \sum_{j=0}^{N-1} a_j |j, b \oplus f(j)\rangle$$

we might gain some 'quantum advantage'.

# Quantum Searching

Consider a black box function  $f:\{0,\dots,N-1\} \rightarrow \{0,1\}$ , where for one target  $t \in \{0,\dots,N-1\}$  we have  $f(t)=1$ .

Task: Find  $t$  with a minimum of queries to  $f$ .

Solution: Lov Grover's quantum search algorithm requires only  $O(\sqrt{N})$  queries and is optimal.

This algorithm consists of a repeated sequence of Fourier transforms over  $\mathbb{Z}/N$ , and phase flip operations

$$U_t : |j\rangle \mapsto (-1)^{f(j)} |j\rangle$$

and

$$U_0 : |j\rangle \mapsto \begin{cases} -|0\rangle & \text{if } j = 0 \\ +|j\rangle & \text{otherwise} \end{cases}$$

# Grover Iteration

The 'Grover Iteration' is defined by

$$G_f = -\text{Four}_N \cdot U_0 \cdot \text{Four}_N^{-1} \cdot U_t$$

Note that  $U_t$  be implemented with one call to the function  $f$  in combination with the phase-flip trick: If  $f: |j, b\rangle \mapsto |j, b \oplus f(j)\rangle$ , then  $f: |j\rangle \otimes |-\rangle \mapsto (-1)^{f(j)} |j\rangle \otimes |-\rangle$ .

Instead of the Fourier transformation over  $\mathbf{Z}/N$ , we can also use other 'mixing operations'.

For example, if  $N=2^n$  then  $H \otimes \dots \otimes H$  works as well.

The Grover iteration 'amplifies' the amplitude of the correct state  $|t\rangle$  with  $f(t)=1$ , at the expense of the others.

# Grover's Algorithm

Given a black box function  $f: \{0, \dots, N-1\} \rightarrow \{0, 1\}$ .

Create the uniform superposition (using  $\text{Four}_N|0\rangle$ ):

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle$$

Apply the Grover iteration  $G_f = -\text{Four} \cdot U_0 \cdot \text{Four}^{-1} \cdot U_t$   
a number of times to  $|\psi\rangle$ .

Measure the register for answer  $t'$   
(and check that  $t'$  indeed does give  $f(t')=1$ ).

# Roughly Analyzing Grover

At any given moment, we can describe the state as

$$|\psi\rangle = a|t\rangle + \beta \sum_{j=0}^{N-1} |j\rangle \quad \text{with } a^2 + N\beta^2 \approx 1$$

Initially we have  $a=0$ ,  $\beta=1/\sqrt{N}$  and we want  $a$  to grow to 1.

After one Grover iteration  $-\text{Four} \cdot U_0 \cdot \text{Four}^{-1} \cdot U_t$  on  $|\psi\rangle$  we have for the new  $a$  amplitude:  $a' = a + 2\beta$ .

For this, you only have to use the facts that:  $U_0 = I - 2|0\rangle\langle 0|$ ,  $U_t = I - 2|t\rangle\langle t|$ ,  $\text{Four} \cdot \text{Four}^{-1} = I$ , and  $\text{Four}|0\rangle\langle 0|\text{Four}^{-1} = \sum_{j,k} |j\rangle\langle k|/N$ .

Because  $\beta \approx \sqrt{(1-a^2)}/\sqrt{N}$ , after  $O(\sqrt{N})$  repeats we have  $a \approx 1$ .

# Analyzing Grover's Algorithm

A proper analysis of the previous algorithm shows that after the  $k$ -th iteration, the amplitude of the target state "t" is:  $\langle t | \psi_k \rangle = \sin(\theta(2k+1)/2)$  with  $\sin(\theta) = 2\sqrt{(N-1)}/N$ .

For large enough  $N$ , this gives  $\theta \approx 2/\sqrt{N}$ , such that  $\langle t | \psi_k \rangle \approx \sin((2k+1)/\sqrt{N})$ , showing that  $k \approx \frac{1}{4}\pi\sqrt{N}$  works.

[KLM, Chapter 8] gives a more detailed analysis that also shows that with  $M$  solutions (instead of 1), you only need  $\approx \frac{1}{4}\pi\sqrt{(N/M)}$  queries to the black box function  $f$ . (If  $N/M = 4$ , then only one call is required.)