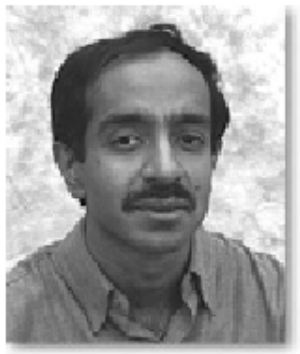
Grover Algorithm

Anuj Dawar

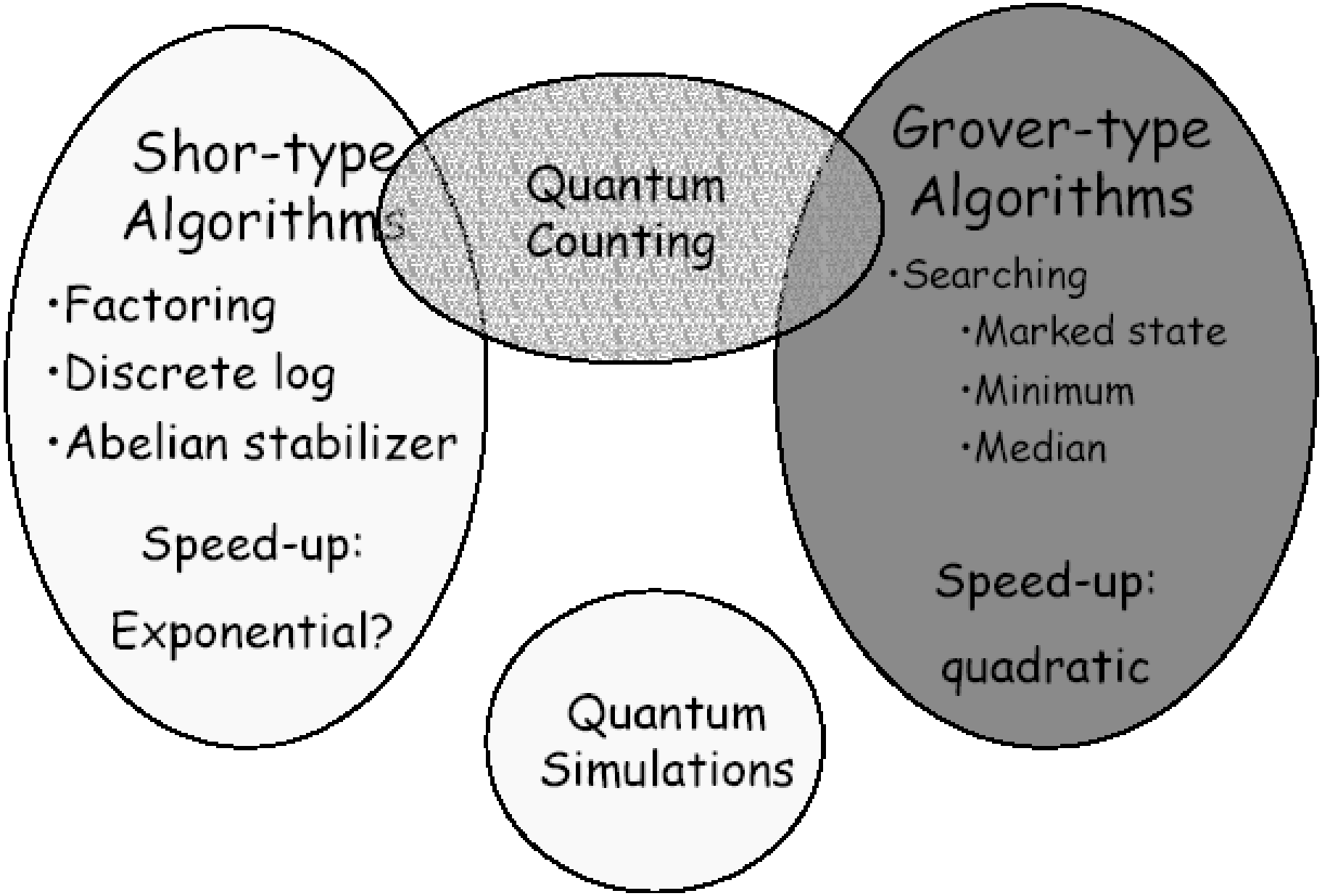
Grover's Algorithm



Lov K. Grover
Bell Labs

Grover
Sesame Street

Quantum Algorithms



Unsorted Database

- Example: Telephone Book

Find the name of the person
with phone number: 3397 0454

- Very difficult task!
- If there are N entries in the phone book, it will take an average of N/2 queries to find the name

Unsorted Database

- Example: Telephone Book

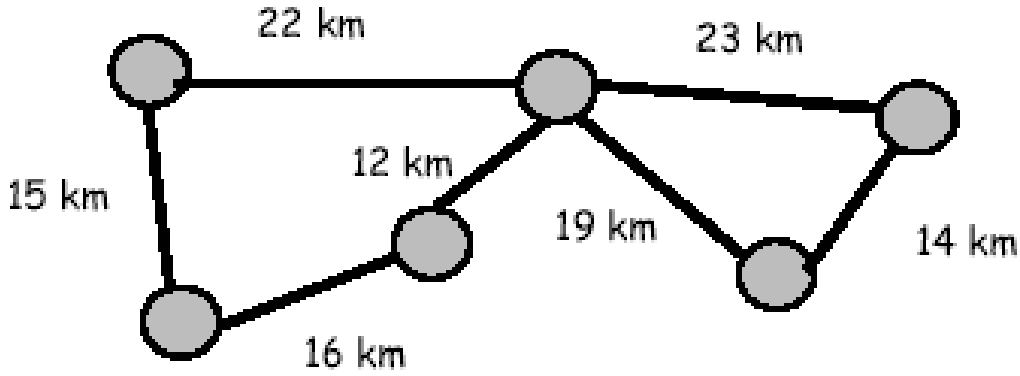
Find the name of the person
with phone number: 3397 0454

- This is not such a good example
  - There exists a more efficient solution
  - The search time is linear with respect to the size of the problem
  - That is because the number of possible inputs scales linearly with the size of the problem

Another example

Traveling salesman Problem

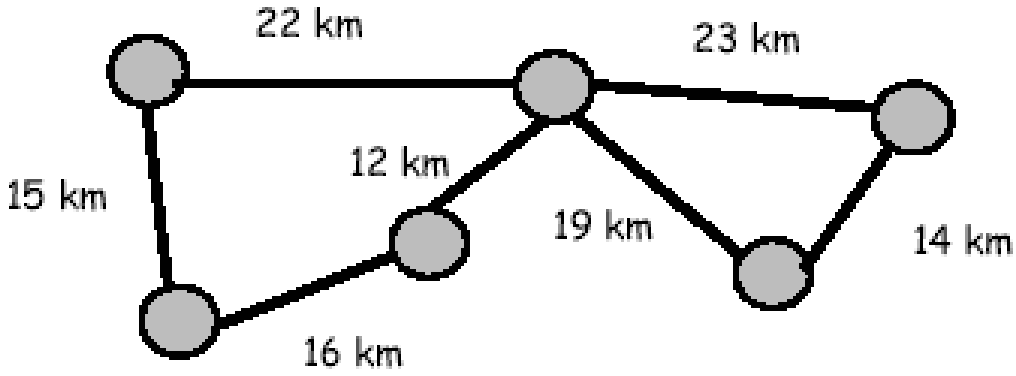
Given a network and a number, *k*, is there a tour through all the cities of length less than *k*?



- What is the size of this problem?
  - The network might be directed (It could take longer to get from A to B than from B to A)
  - Each city could be connected to every other city
  - Therefore if there are c cities, there could be c(c-1) edges
  - Each edge requires log k bits to store

Traveling salesman Problem

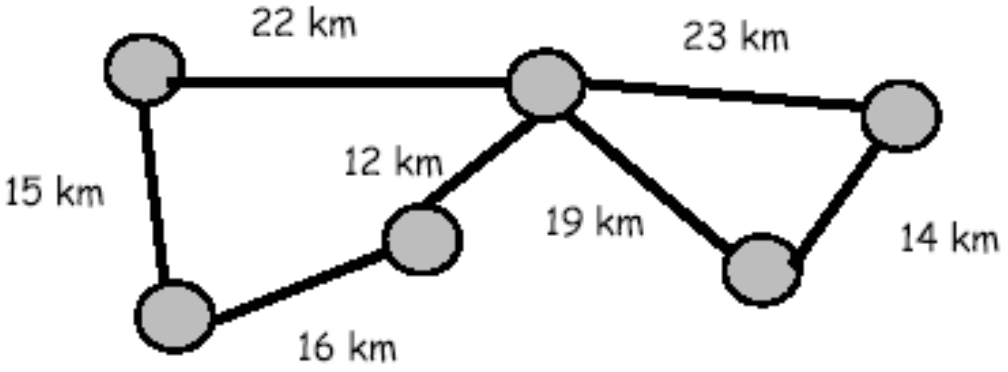
Given a network and a number, *k*, is there a tour through all the cities of length less than *k*?



- The size of the problem is $O(c^2 \log k)$,
- How many possible tours are there?
  - Equal to the number of permutations of cities
  - There are c! possible tours
  - $c! \sim O(\exp(c \log c))$
- Therefore the number of possible inputs scales exponentially with the size of the problem

Traveling salesman Problem

Given a network and a number, *k*, is there a tour through all the cities of length less than *k*?



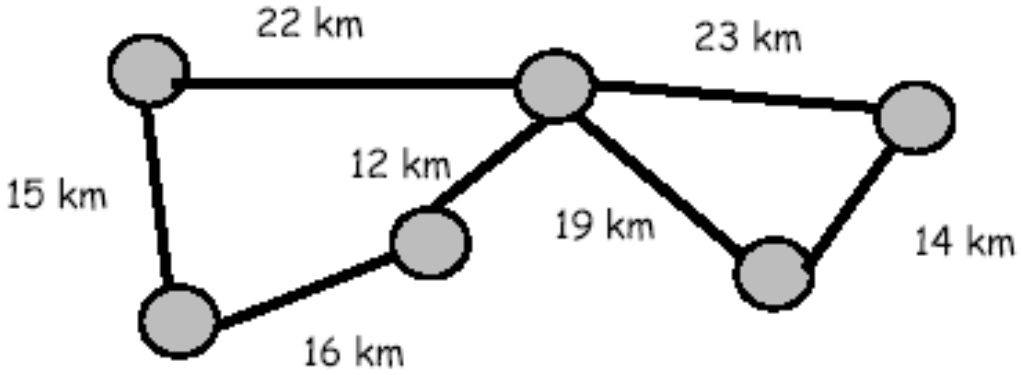
- Suppose there is exactly one solution
- We could try to solve the problem by choosing a tour and testing if it is less than k
- It would take on average c!/2 attempts to find the solution

This is a better example of problems for Grover

- Grover's algorithm allows us to find a solution by using only $O(\sqrt{c!})$ attempts

Traveling salesman Problem

Given a network and a number, *k*, is there a tour through all the cities of length less than *k*?



- $O(\sqrt{c}!)$ is still exponential
- We haven't made the problem "tractable"
- Suppose there are 10 billion permutations
- Grover's algorithm would only require one hundred thousand queries
- Classically we would require an average of 5 billion queries

Advantage of Grover, although not "tractable"

The Quantum Oracle

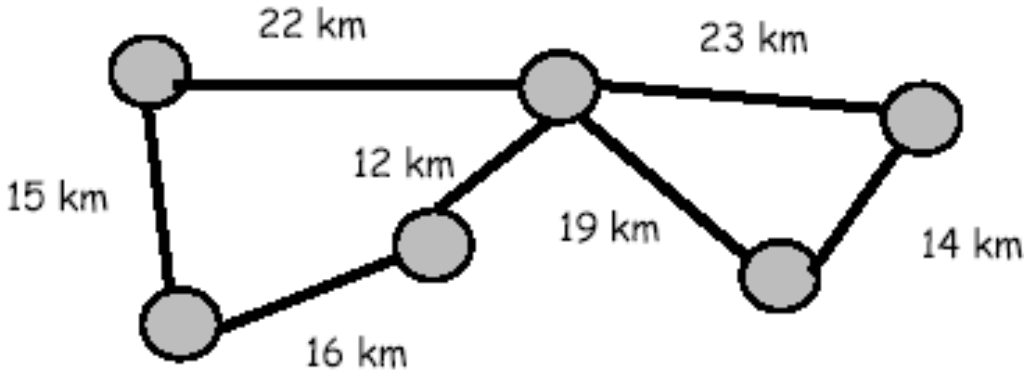
oracle [or'a-kl] *n* a medium or agency, especially in ancient Greece, of divine revelation; a person of great wisdom; a wise utterance

- Not an all-knowing device
- Simple a device which can efficiently check whether a given solution is correct
- A witness
- Telephone book:
  - Is Jones the name of the person with phone number 3397 0454?

Example of oracle

Another example of Oracle, more realistic

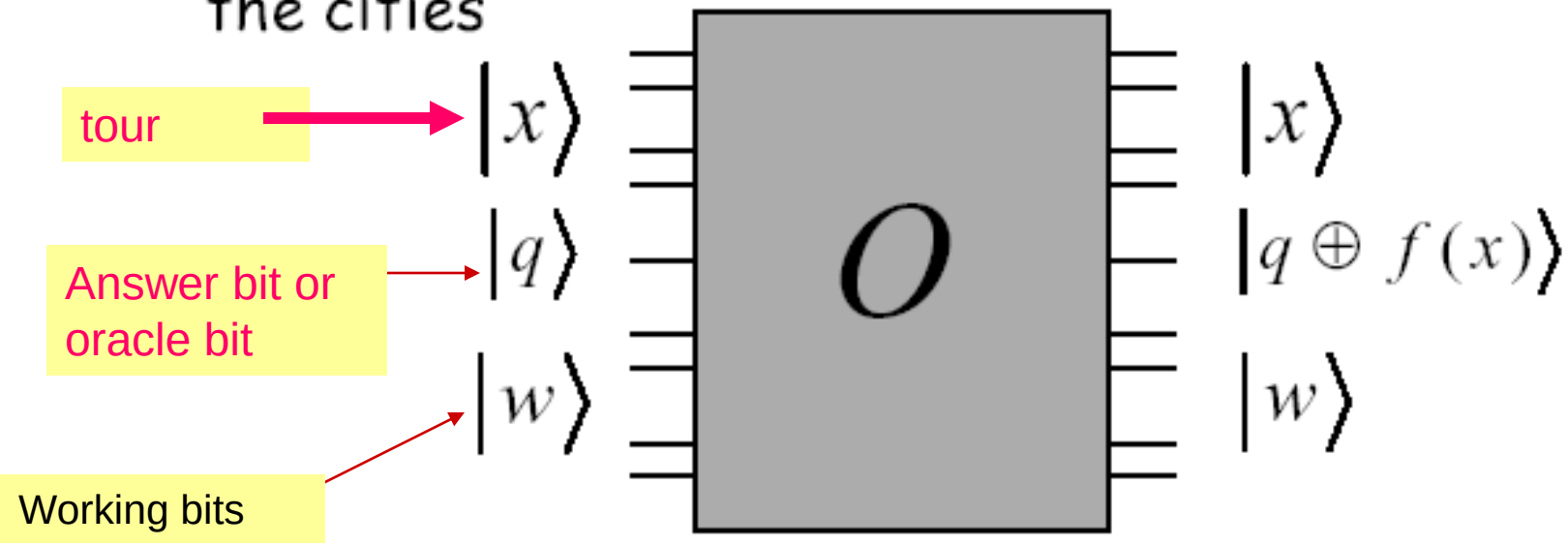
Given a network and a number, k, is there a tour through all the cities of length less than k?



- We could write a computer program to check whether a tour is a valid solution
- We could make it reversible
- Therefore, we could implement it on a quantum computer

The Quantum Oracle

- Imagine that the bit string, x, represents a tour of the cities



$$|x\rangle|q\rangle|w\rangle \xrightarrow{O} |x\rangle|q \oplus f(x)\rangle|w\rangle$$

$$f(x) = \begin{cases} 1, & \text{if } x \text{ is a solution} \\ 0, & \text{if } x \text{ is not a solution} \end{cases}$$

The Quantum Oracle

- The quantum oracle bit-flips the oracle qubit if the input is a valid solution
- The inner workings of the oracle are by no means `magical'
- In the traveling salesman example, the oracle qubit would be flipped if x encoded a tour of the cities with a distance less than k.
- By abstracting the problem using an oracle, we can forget about the specific problem we are trying to solve

Possible Exam Problems

- Formulate an oracle (reversible circuit) for the following problems:
  - 1. Graph coloring with of a planar map.
  - 2. Graph coloring with the minimum number of colors of an arbitrary graph.
  - 3. Satisfiability.
  - 4. Set Covering.
  - 5. Euler Path in a graph.
  - 6. Hamiltonian Path in a graph.
  - 7. Cryptographic Puzzle like SEND+MORE=MONEY.