A quantum computer is a device for computation that makes direct use of quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data. Quantum computers are different from traditional computers based on transistors. The basic principle behind quantum computation is that quantum properties can be used to represent data and perform operations on these data.[1] A theoretical model is the quantum Turing machine, also known as the universal quantum computer.

Although quantum computing is still in its infancy, experiments have been carried out in which quantum computational operations were executed on a very small number of qubits (quantum bits). Both practical and theoretical research continues, and many national government and military funding agencies support quantum computing research to develop quantum computers for both civilian and national security purposes, such as cryptanalysis.[2]

If large-scale quantum computers can be built, they will be able to solve certain problems much faster than any classical computer using the best currently known algorithms (for example integer factorization using Shor's algorithm or the simulation of quantum many-body systems). Furthermore, there exist quantum algorithms, such as Simon's algorithm, which run exponentially faster than any possible probabilistic classical algorithm.[3] Given enough resources, a classical computer can simulate an arbitrary quantum computer. Hence, ignoring computational and space constraints, a quantum computer is not capable of solving any problem which a classical computer cannot.[4]

Integer factorization is believed to be computationally infeasible with an ordinary computer for large integers if they are the product of few prime numbers (e.g., products of two 300-digit primes).[7] By comparison, a quantum computer could efficiently solve this problem using Shor's algorithm to find its factors. This ability would allow a quantum computer to decrypt many of the cryptographic systems in use today, in the sense that there would be a polynomial time (in the number of digits of the integer) algorithm for solving the problem. In particular, most of the popular public key ciphers are based on the difficulty of factorizing integers (or the related discrete logarithm problem which can also be solved by Shor's algorithm), including forms of RSA. These are used to protect secure Web pages, encrypted email, and many other types of data. Breaking these would have significant ramifications for electronic privacy and security.

However, other existing cryptographic algorithms do not appear to be broken by these algorithms.[8][9] Some public-key algorithms are based on problems other than the integer factorization and discrete logarithm problems to which Shor's algorithm applies, like the McEliece cryptosystem based on a problem in coding theory.[8][10] Lattice based cryptosystems are also not known to be broken by quantum computers, and finding a polynomial time algorithm for solving the dihedral hidden subgroup problem, which would break many lattice based cryptosystems, is a well-studied open problem.[11] It has been proven that applying Grover's algorithm to break a symmetric (secret key) algorithm by brute force requires roughly $2^{n/2}$ invocations of the underlying cryptographic algorithm, compared with roughly $2^n$ in the classical case,[12] meaning that symmetric key lengths are effectively halved: AES-256 would have the same security against an attack using Grover's algorithm that AES-128 has against classical brute-force search (see Key size). Quantum cryptography could potentially fulfill some of the functions of public key cryptography.

Besides factorization and discrete logarithms, quantum algorithms offering a more than polynomial speedup over the best known classical algorithm have been found for several problems,[13] including the simulation of quantum physical processes from chemistry and solid state physics, the approximation of Jones polynomial, and solving Pell's equation. No mathematical proof has been found that shows that an equally fast classical algorithm cannot be discovered, although this is considered unlikely. For some problems, quantum computers offer a polynomial spe
The most well-known example of this is quantum database search, which can be solved by Grover's algorithm using quadratically fewer queries to the database than are required by classical algorithms. In this case the advantage is provable. Several other examples of provable quantum speedups for query problems have subsequently been discovered, such as for finding collisions in two-to-one functions and evaluating NAND trees.

Consider a problem that has these four properties:

1. The only way to solve it is to guess answers repeatedly and check them,
2. The number of possible answers to check is the same as the number of inputs,
3. Every possible answer takes the same amount of time to check, and
4. There are no clues about which answers might be better: generating possibilities randomly is just as good as checking them in some special order.

An example of this is a password cracker that attempts to guess the password for an encrypted file (assuming that the password has a maximum possible length).

For problems with all four properties, the time for a quantum computer to solve this will be proportional to the square root of the number of inputs. That can be a very large speedup, reducing some problems from years to seconds. It can be used to attack symmetric ciphers such as Triple DES and AES by attempting to guess the secret key.

Grover's algorithm can also be used to obtain a quadratic speed-up over a brute-force search for a class of problems known as NP-complete.

Since chemistry and nanotechnology rely on understanding quantum systems, and such systems are impossible to simulate in an efficient manner classically, many believe quantum simulation will be one of the most important applications of quantum computing.[14]

There are a number of practical difficulties in building a quantum computer, and thus far quantum computers have only solved trivial problems. David DiVincenzo, of IBM, listed the following requirements for a practical quantum computer:[15]

- scalable physically to increase the number of qubits;
- qubits can be initialized to arbitrary values;
- quantum gates faster than decoherence time;
- universal gate set;
- qubits can be read easily.