Project topic:

IMPLEMENTATION OF SHOR'S ALGORITHM FOR INTEGER FACTORIZATION

SHOR'S ALGORITHM

Shor's algorithm is a polynomial-time quantum computer algorithm for integer factorization.[1] Informally, it solves the following problem: Given an integer {\display style N}N, find its prime factors. It was invented in 1994 by the American mathematician Peter Shor.

On a quantum computer, to factor an integer {\display style N}N, Shor's algorithm runs in polynomial time (the time taken is polynomial in {\display style \log N}{\display style \log N}, the size of the integer given as input).[2] Specifically, it takes quantum gates of order {\display style O\!\left((\log N)^{2}(\log \log N)(\log \log \log N)\right)}{\display style O\!\left((\log N)^{2}(\log \log N)(\log \log \log N)\right)} using fast multiplication,[3] thus demonstrating that the integer-factorization problem can be efficiently solved on a quantum computer and is consequently in the complexity class BQP. This is almost exponentially faster than the most efficient known classical factoring algorithm, the general number field sieve, which works in sub-exponential time: {\display style O\!\left(e^{1.9(\log N)^{1/3}(\log \log N)^{2/3}}\right)}{\display style O\!\left(e^{1.9(\log N)^{1/3}(\log \log N)^{2/3}}\right)}.[4] The efficiency of Shor's algorithm is due to the efficiency of the quantum Fourier transform, and modular exponentiation by repeated squaring.[5]

If a quantum computer with a sufficient number of qubits could operate without succumbing to quantum noise and other quantum-DE coherence phenomena, then Shor's algorithm could be used to break public-key cryptography schemes, such as the widely used RSA scheme. RSA is based on the assumption that factoring large integers is computationally intractable. As far as is known, this assumption is valid for classical (non-quantum) computers; no classical algorithm is known that can factor integers in polynomial time. However, Shor's algorithm shows that factoring integers is efficient on an ideal quantum computer, so it may be feasible to defeat RSA by constructing a large quantum computer. It was also a powerful motivator for the design and construction of quantum computers, and for the study of new quantum-computer algorithms. It has also facilitated research on new cryptosystems that are secure from quantum computers, collectively called post-quantum cryptography.

FRONT END:

PYTHON: Programs are created using qiskit Module that was developed by IBM company for quantum computing. And the code is written in Jupiter notebook of anaconda.

BACK END: IBM_16_MELBOURNE

This is a backend system provided by IBM company to do quantum computing projects.