Instructor (me): James R. Lee
TAs: Haotian Jiang and Chris Kang
Course info: https://homes.cs.washington.edu/~jrl/teaching/cse599Q/
CSE 599Q: Intro to Quantum Computation

Course description:
An introduction to the field of quantum computing from the perspective of computer science.
Quantum computing involves the revolutionary paradigm of computers that exploit the parallelism of the quantum mechanical laws of the universe. Topics covered include:
- The roles of quantum mechanics
- Quantum computing (quantum money, quantum states, and quantum algorithms)
- Quantum information theory (entangled states, measurements, and quantum channels)
- Quantum state tomography (computing and distinguishing quantum states)
- Quantum error correction
- Quantum “holography”

Prerequisites: A background in undergraduate level linear algebra, probability theory, and/or theory.

Lectures:
- Jan 19: Computing with parallelism
- Jan 26: Quantum computing and parallelism
- Feb 2: Quantum algorithms
- Feb 9: Quantum error correction
- Feb 16: Quantum computing and parallelism
- Feb 23: Quantum algorithms
- Mar 2: Quantum error correction
- Mar 9: Quantum computing and parallelism
- Mar 16: Quantum algorithms
- Mar 23: Quantum error correction
- Mar 30: Quantum computing and parallelism
- Apr 6: Quantum algorithms
- Apr 13: Quantum error correction
- Apr 20: Quantum computing and parallelism
- Apr 27: Quantum algorithms

Reference material:
- Quantum Computer Science: An Introduction (Mermin)
- Quantum Computation and Quantum Information (Nielsen and Chuang)

Related courses:
- Quantum computing (SU, CMU)
- Quantum computation and quantum information (CS, UTD, CMU)
- Classical, quantum, and complexity theory (Berkeley)
- Quantum mechanics on the quantum level
- Quantum computing for the intermediate (lecture, seminar)
"Quantum computing is... nothing less than a distinctively new way of harnessing nature... it will be the first technology that allows useful tasks to be performed in collaboration between parallel universes."

"When a quantum factorization engine is factorizing a 250-digit number, the number of interfering universes will be of the order of $10^{500}$. This staggeringly large number is the reason why Shor's algorithm makes factorization tractable. I said [earlier in the book] that the algorithm requires only a few thousand [or maybe a million] operations. I meant, of course, a few thousand parallel operations in each universe that contributes to the answer. All those computations are performed in parallel, in different universes, and share their results through interference."

Quotes from David Deutsch (cofounder of quantum computing)
Math $\cap$ CS $\cap$ Physics

Foundations of Quantum Info.

Math

Physics

CS

Algorithms

Quantum Computation
quantum mechanics arose from observations

Blackbody radiation problem

“ultraviolet catastrophe”

Photoelectric effect

Spectral lines

Rydberg formula
Foundations of quantum mechanics: 1900–1925

Quantum computation: 1980+
(Benioff, Feynman, Manin, Deutsch, ...)

(hacking the universe)

Copenhagen interp. of quantum mech.
all aboard the hype train
Questions:

- Are quantum computers more powerful than classical computers?
- For what problems?
- Near-term prospects for demonstrating this? ("quantum supremacy")
Computational efficiency

Problem: Multiplying two 500-digit numbers

\[ \begin{array}{c}
39764257 \ldots \\
64291107 \ldots \\
\end{array} \begin{array}{c}
3253432 \\
4132463 \\
\end{array} \]

\[ \begin{array}{c}
\ldots \\
\ldots \\
\end{array} \begin{array}{c}
9760296 \\
9920000 \\
\end{array} \]

FET-based algorithm: \( O(n \log n) \) time
(e.g. multi. two \( 1,000,000 \)-digit #s
or a PSY in \( \sim 1 \) ms)

Problem: Primality testing

Given an \( n \)-digit number, say whether it's prime.

→ Miller-Rabin test (Randomized)

\( \Pr[\text{alg. outputs the correct answer}] \geq 1 - 10^{-20} \)
PS4 can check primality of a 500 digit # \( m < 1 \mu \text{s.} \)

\[
\{ \text{Argarwal-Kayal-Saxena 2002:} \quad \text{Efficient deterministic alg. for primality testing} \}\]

**Problem: Factoring n-digit numbers**

Given an \( n \)-digit \( N \), find a nontrivial factor.

\[
\sqrt{N} = \sqrt{10^n} = 3^n
\]

Pollard’ 96 \( \theta(10^{-\frac{1}{4}}) \) time for an \( n \)-digit #

\[
\text{RSA - 284} \quad \text{factored a 284-digit #}
\]

\[
[\text{RSA - 2048} \leftarrow 2^{10^6}] \quad \text{RSA cryptosystem}
\]
Shor '94: Quantum computers can factor n-digit numbers in \( O(n^2) \) steps

Quantum cell phone: Solve \( n = 500 \) in < 4 m

15 = 5 \times 3 \ \checkmark \ \ (\text{quantum computer})