This is CS4048!

GCR:wzj3vua

DiVincenzo's criteria

DiVincenzo's criteria

In the year 2000, David DiVincenzo proposed a wishlist for the experimental characteristics of a quantum computer. DiVincenzo's criteria have since become the main guideline for physicists and engineers building quantum computers.

DiVincenzo's criteria

1. Well-characterized and scalable qubits. Many of the quantum systems that we find in nature are not qubits, since we can't just isolate two specific quantum states and tarder them. We must find a way to make them behave as such. Moreover, we need to put many of these systems together.

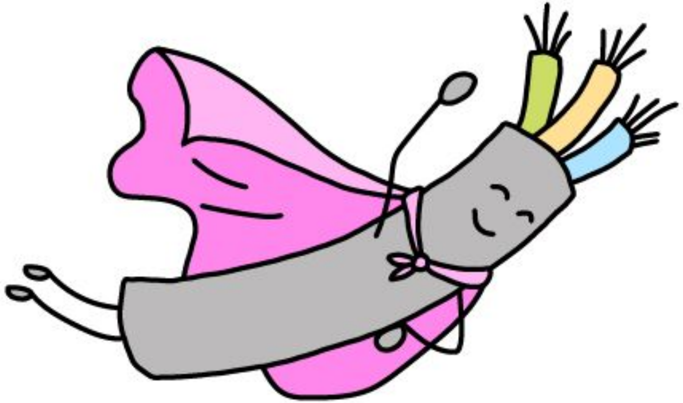
2. Qubit initialization. We must be able to prepare the same state repeatedly within an acceptable margin of error.

3. Long coherence times. Qubits will lose their quantum properties after interacting with their environment for a while. We need them to last long enough so that we can perform quantum operations.

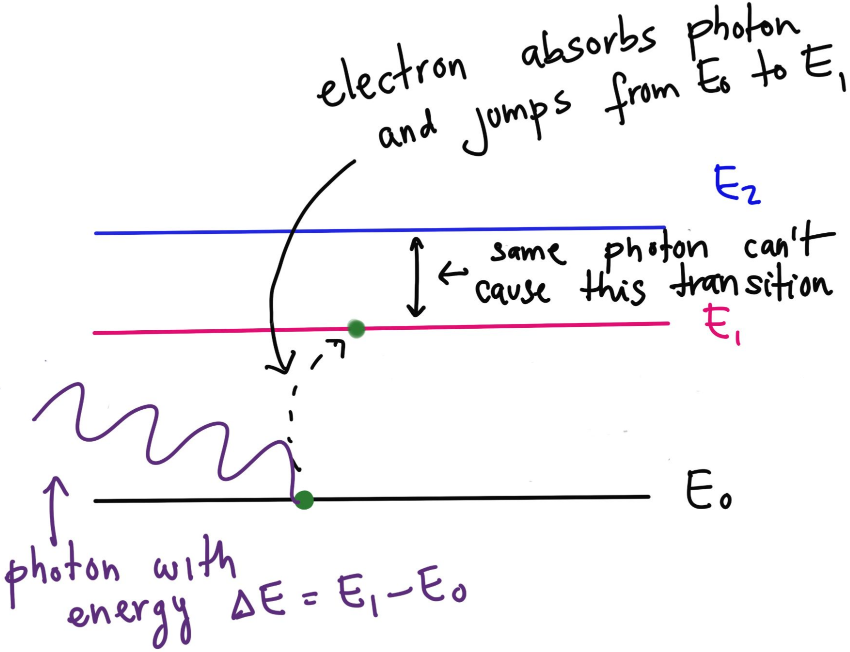
4. Universal set of gates. We need to perform arbitrary operations on the qubits. To do this, we require both single-qubit gates and two-qubit gates.

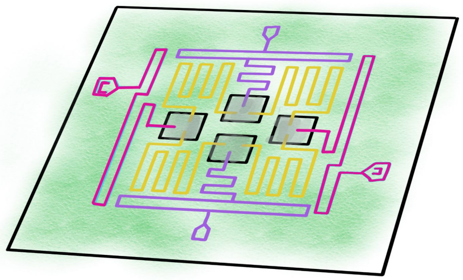
5. Measurement of individual qubits. To read the result of a quantum algorithm, we must accurately measure the final state of a pre-chosen set of qubits.

SUPERCONDUCTING QUBITS

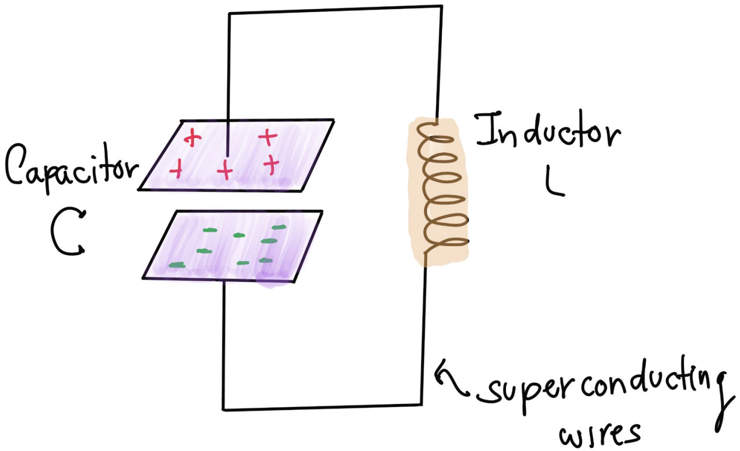


SUPERCONDUCTING QUBITS

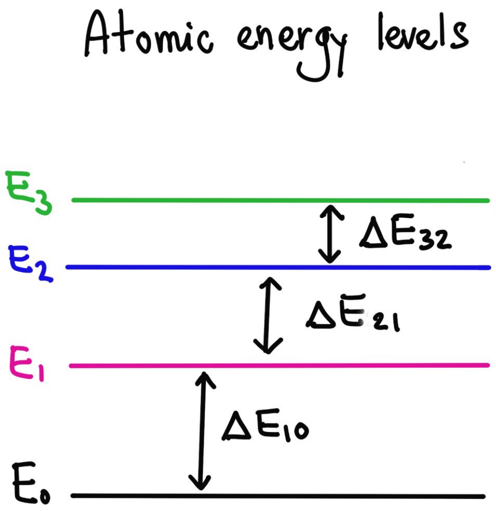




Superconducting Qubits

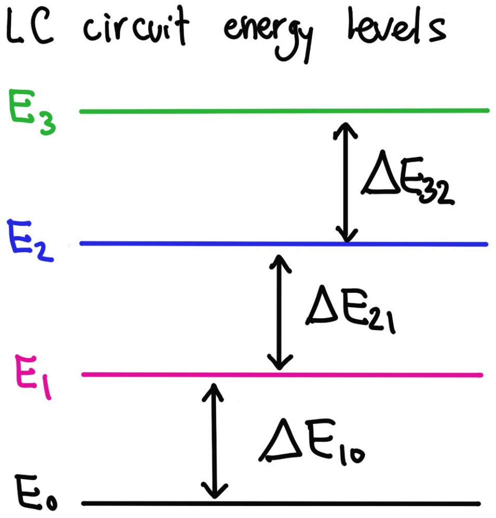


Superconducting Qubits



$\Delta E_{10} \neq \Delta E_{21} \neq \Delta E_{32}$

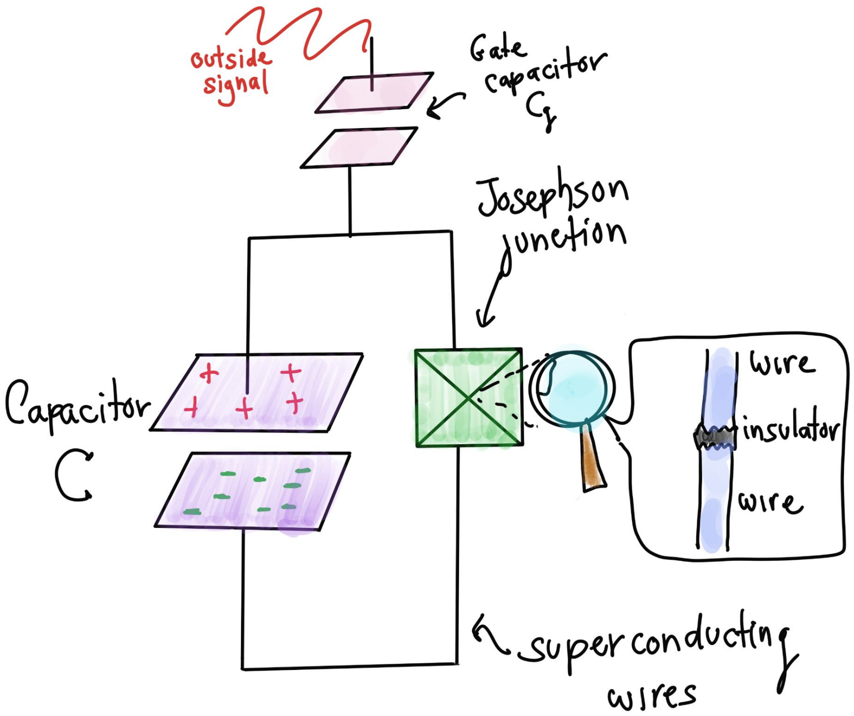
(non-uniform)



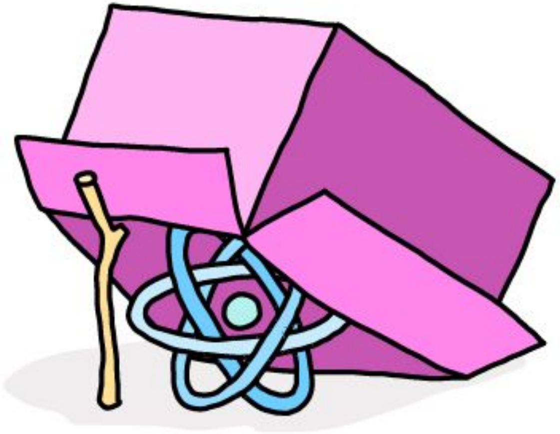
$\Delta E_{10} = \Delta E_{21} = \Delta E_{32}$

(uniform)

Superconducting Qubits



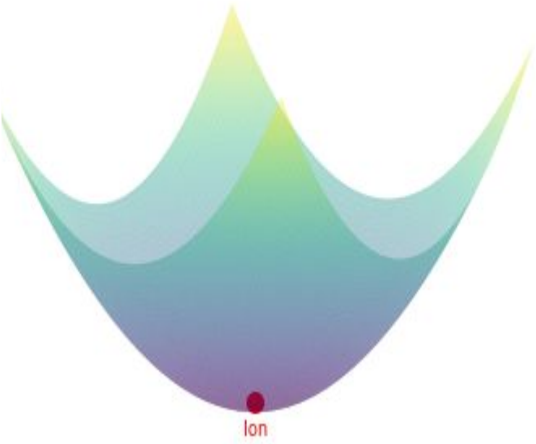
Trapped ion Qubit



Trapped Ion

Why do we use ions, i.e., charged atoms, as qubits? The main reason is that they can be contained (that is, trapped) in one precise location using electric fields.

It is not easy to create electric fields that contain the ion in a tiny region of space. The ideal configuration of an electric field —also known as a potential— would look like this



Trapped Ion

Calcium (Ca) and Strontium (Sr) enable scalable ion-trapped quantum computing.

- Vacuum Chamber: Steel chamber with electrodes at 5.37 K.
- Ionization: Lasers convert Ca and Sr atoms into ions.
- Trapping: Electric fields hold ions 50 μm above the chip, cooled by lasers.

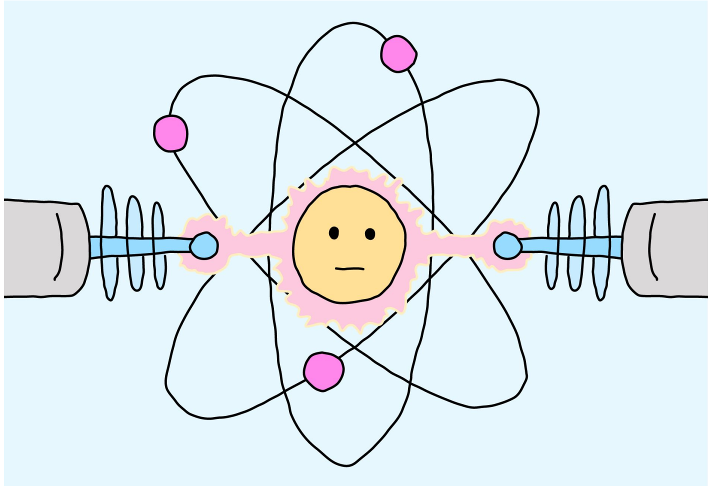
Sr+ Ions: Qubits storing quantum information.

Ca+ Ions: Absorb energy to cool Sr+ ions.

Laser pulses entangle Sr+ and Ca+ ions to form quantum gates.

Entangled Sr+ ions transfer quantum information.

Neutral Atom Qubit



Neutral Atom

Neutral Atom Computing uses uncharged atoms (neutral atoms) as qubits, typically arranged in arrays using optical tweezers.

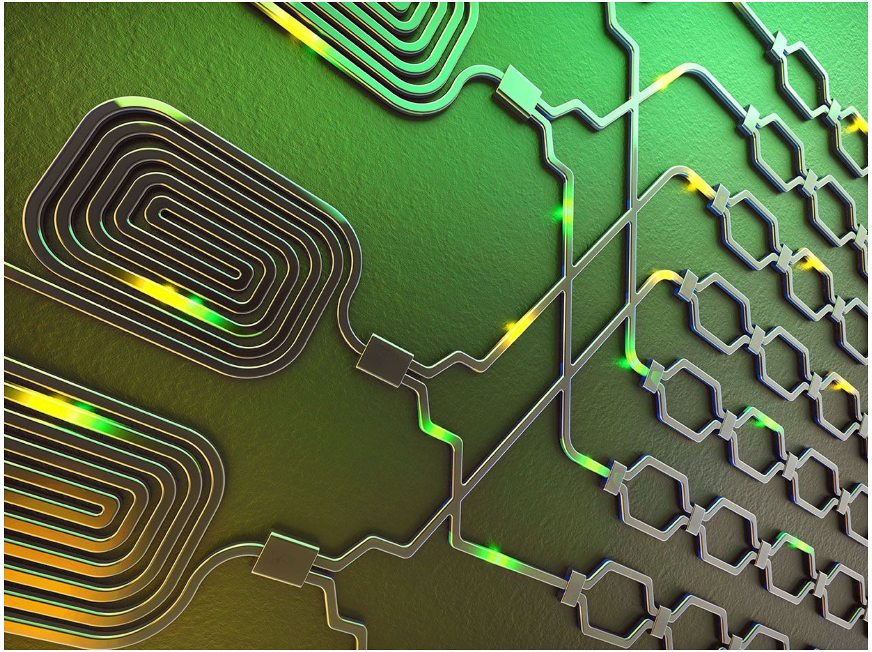
Atoms are manipulated and entangled via lasers or microwaves, with their internal states encoding quantum information.

Neutral atom systems offer scalability due to their compactness and the ability to create large arrays of qubits. They also benefit from long coherence times and flexible architectures for quantum operations.

Neutral Atom

1. Possible to work with a large number of qubits.
2. Fault-tolerant model and error correction are much more accessible.
3. Powerful computational problems can be solved using this platform.
4. Absolute temperature is required to carry out the computation.

Photonic Qubit



Photonics Qubit

Photonic qubits are quantum bits encoded in the properties of photons, such as polarization, phase, or photon number.

These qubits can exist in superpositions, allowing them to perform quantum computations.

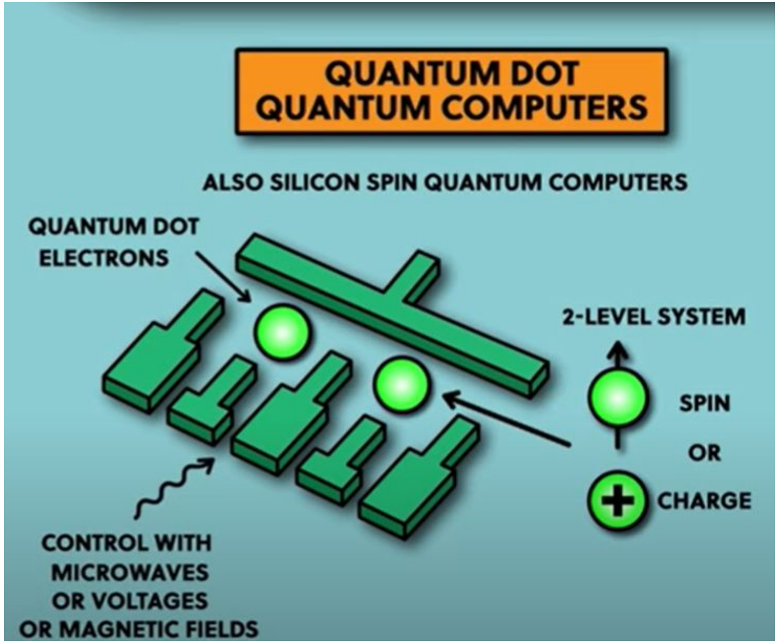
Photons are ideal for quantum communication due to their ability to travel long distances without significant loss.

They are used in quantum systems for their high speed and low interference with the environment.

Photonics

1. Qubits in this technology are much more stable.
2. A large number of photons can be entangled.
3. Computation can be done at room temperature.
4. This technology has achieved quantum supremacy.
5. Less fault-tolerant and difficult to correct errors.

Quantum DOts



Quantum Dots

Quantum dots are nanoscale semiconductor structures that confine electrons in three dimensions, creating quantized energy levels.

These confined electrons can be manipulated to represent qubits, where the electron's spin or charge state can encode quantum information.

Their small size and tunable properties make them promising candidates for scalable quantum computers and quantum communication systems.

Key Terms

Fidelity : refers to a measure of how accurately a quantum state or quantum operation matches its ideal or intended version. It is a critical metric used to evaluate the quality and reliability of quantum processes.

Modalities: refer to different physical implementations of qubits, such as trapped ions, superconducting circuits, photonic systems, or quantum dots. Each modality uses distinct technologies and principles to encode, manipulate, and measure quantum information.

Key Terms

Cryogenic: refers to extremely low temperatures, typically below -150°C, used to study or manipulate materials and systems. In quantum computing, cryogenic environments are essential for technologies like superconducting qubits to minimize thermal noise and maintain quantum coherence.

Coherence times: refer to the duration for which a quantum system, like a qubit, can maintain its quantum state (superposition or entanglement) before it is disrupted by noise or decoherence. Longer coherence times are crucial for performing reliable quantum computations and minimizing errors.

Qubit Type	Pros/Cons	Select Players
Superconducting	***Pros***: High gate speeds and fidelities. Can leverage standard lithographic processes. Among first qubit modalities so has a head start.	rigetti, Google, IBM Q, QuTech, OQC, IQM, qci Quantum Circuits, Inc, 本源量子 Origin Quantum
	Cons: Requires cryogenic cooling; short coherence times; microwave interconnect frequencies still not well understood.	
Trapped Ions	***Pros***: Extremely high gate fidelities and long coherence times. Extreme cryogenic cooling not required. Ions are perfect and consistent.	IONQ, AQT, QUANTINUUM, oxford ionics, Universal Quantum
	Cons: Slow gate times/ operations and low connectivity between qubits. Lasers hard to align and scale. Ultra-high vacuum required. Ion charges may restrict scalability.	

Photonics	***Pros***: Extremely fast gate speeds and promising fidelities. No cryogenics or vacuums required. Small overall footprint. Can leverage existing CMOS fabs.	PsiQuantum, XANADU, QUIX QUANTUM, ORCA Computing
	Cons: Noise from photon loss; each program requires its own chip. Photons don't naturally interact so 2Q gate challenges.	
Neutral Atoms	***Pros***: Long coherence times. Atoms are perfect and consistent. Strong connectivity, including more than 2Q. External cryogenics not required.	ColdQuanta, QuEra Computing Inc., atom computing, PASQAL
	Cons: Requires ultra-high vacuums. Laser scaling challenging.	

Silicon Spin/Quantum Dots	***Pros***: Leverages existing semiconductor technology. Strong gate fidelities and speeds.	intel Silicon Quantum Computing diraq QUANTUM MOTION QUANTUM BRILLIANCE
	Cons: Requires cryogenics. Only a few entangled gates to-date with low coherence times. Interference/cross-talk challenges.	

QC hardware

https://medium.com/quantum-untangled/quantum-hardware-in-a-nutshell-50cc70c1ffd4
https://pennylane.ai/qml/demos/tutorial_photonics
https://pennylane.ai/qml/demos/tutorial_sc_qubits

https://pennylane.ai/qml/demos/tutorial_neutral_atoms

Quantum Complexity Theory

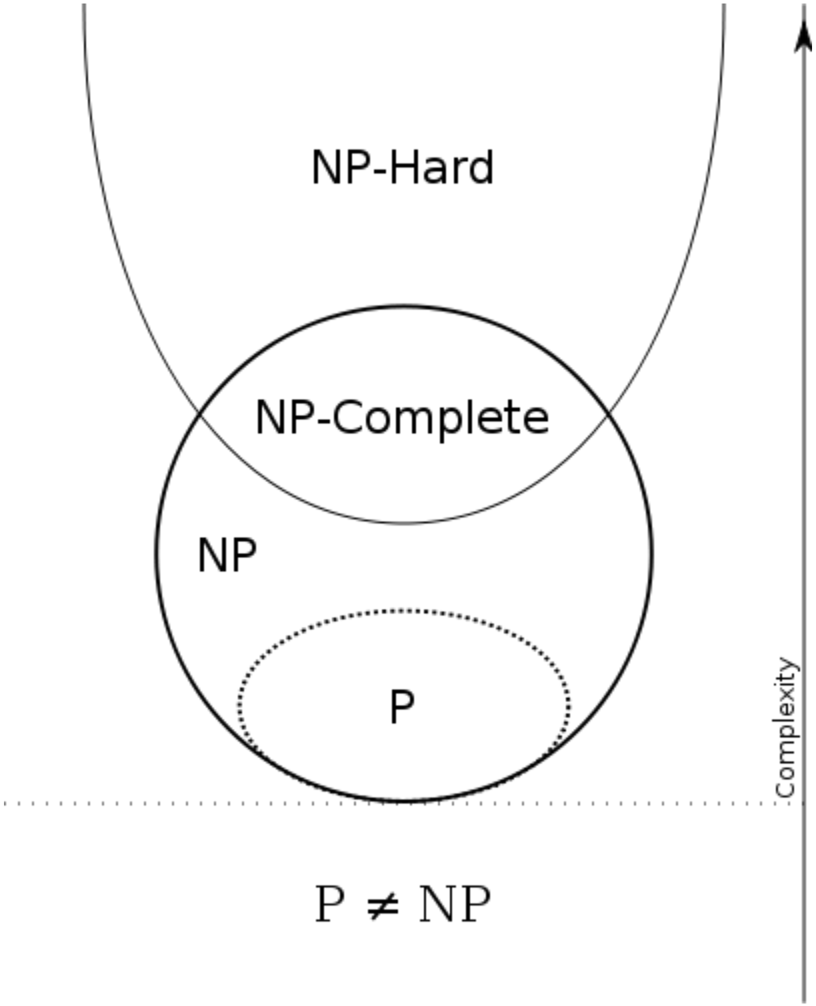
" QUANTUM COMPUTING CAN SOLVE PROBLEMS THAT CLASSICAL COMPUTERS CAN'T "

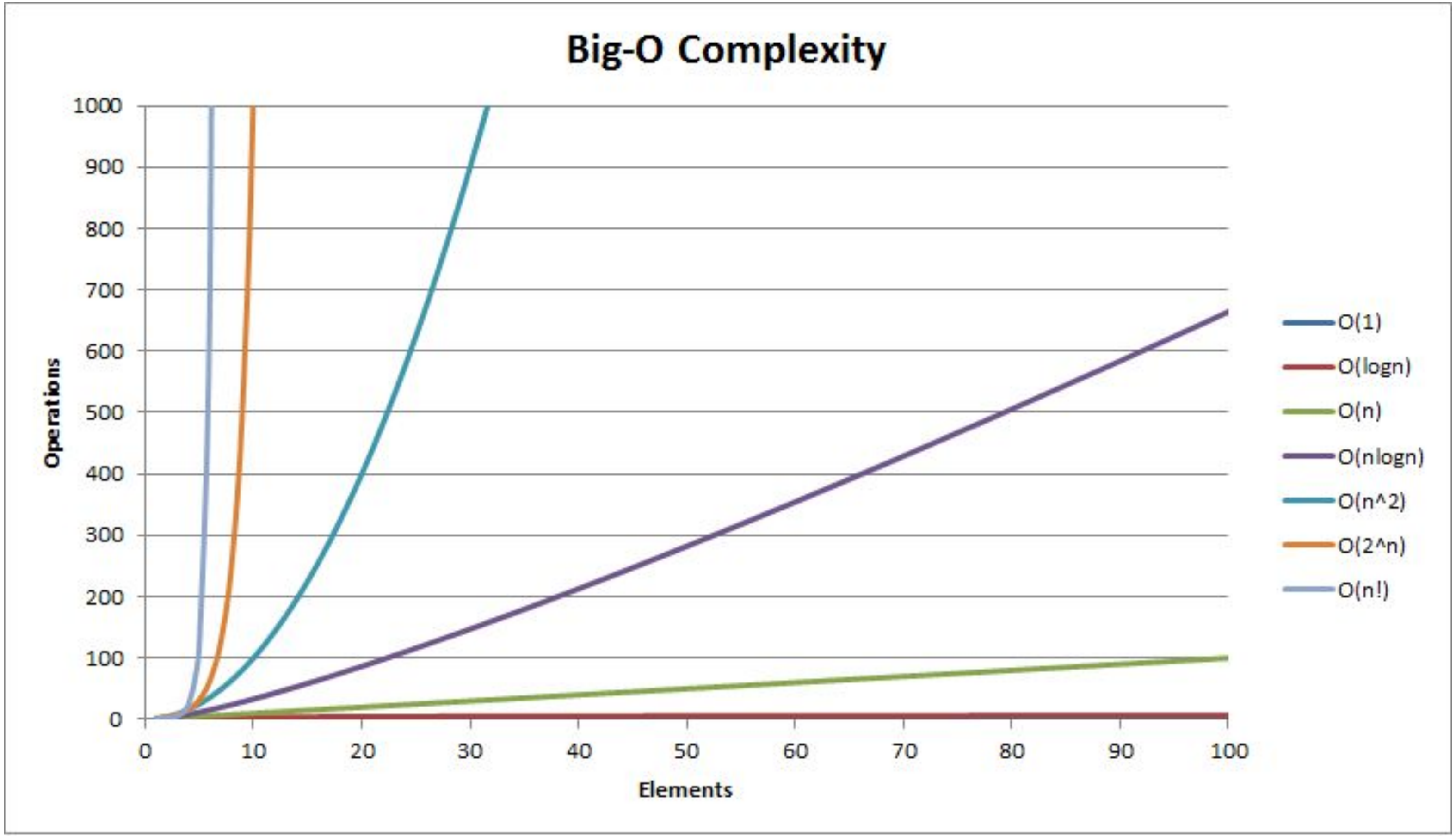
Have you heard this?

Classical Complexity Classes

This is the study of:

How long it takes for a classical computer to solve different types of problems?





Big-O Complexity
Operations
Elements
1000
900
800
700
600
500
400
300
200
100
0
0
10
20
30
40
50
60
70
80
90
100
O(1)
O(logn)
O(n)
O(nlogn)
O(n^2)
O(2^n)
O(n!)

Decision Problem

All the problems that the four categories above (P, NP, NP-Complete, NP-Hard) describe are known as Decision Problems.

A Decision Problem is any problem that has a yes or no answer. For Example:

- Given an integer, is it even or odd?
- Given a graph, does a Hamiltonian Cycle exist in it? (Given a bunch of points with lines between them, does a line exist that goes through each point exactly ONCE exist?)

P

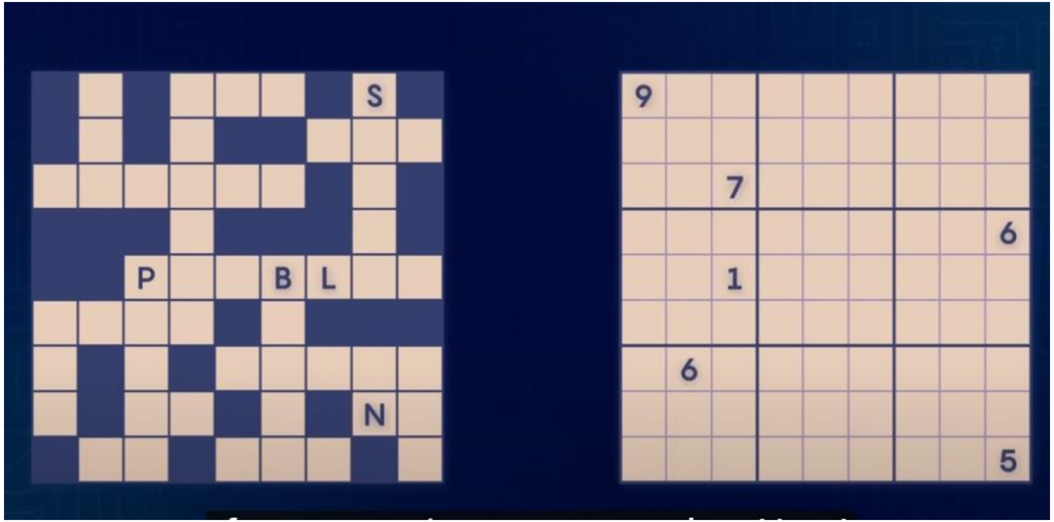
P stands for “polynomial time”
It is the collection of decision problems(problems with a “yes” or “no” answer) that can be solved by a deterministic machine in polynomial time.

- **Matrix Multiplication** – O(n^3)
- **Breadth-First Search (BFS)** – O(V+E)
- **Graph Coloring** – O(kV+E)
- **Sorting a List of Numbers** – O(nlogn)

NP

NP stands for "nondeterministic polynomial time" and it represents the set of all problems that have a solution that can be verified in a reasonable amount of time by a classical computer.

Problems that can be solved by a non-deterministic machine in polynomial time.



NP

NP class problems don’t have a polynomial runtime to solve, but have a polynomial run-time to verify solutions (difficult to solve, easy to check a given answer)

- **Traveling Salesman Problem (TSP)** – O(n!)
- **Knapsack Problem** – O(2^n)
- **Clique Problem** – O(2^n)
- **Vertex Cover Problem –** O(2^n)

NP Complete

NP-Complete is the set of problems that other problems in NP can be reduced to in polynomial time.

This means if we can find an algorithm that can efficiently solve an NP-Complete problem, then other problems in NP can also be solved efficiently IF they can be reduced to the NP-Complete problem form.

A problem that falls into NP-Complete is the 3-SAT problem.

NP Hard

NP-Hard is the set of problems that are at least as hard as the NP-Complete problems. NP-Hard problems don't have to be in NP and they don't have to be decision problems either.

If you go back to the Euler Diagram, you'll find that it intersects with NP but also goes beyond it.

A more precise definition is that a problem X can be considered NP-Hard if there is an NP-Complete problem Y such that Y can be reduced to X.

NP Hard

If any NP-hard problem can be solved in polynomial time, all NP problems can be.

However, NP-hard problems are not necessarily in NP, as they may not have a solution that can be verified in polynomial time.

NP Hard

These problems can be computationally more challenging than those in NP because they may require more than polynomial time or might not even be decidable. Solving an NP-hard problem is generally considered very difficult, and no polynomial-time solution is known.

P Space

PSPACE stands for “polynomial space” and it represents the class of problems that can be solved using a polynomial amount of space on a classical computer.

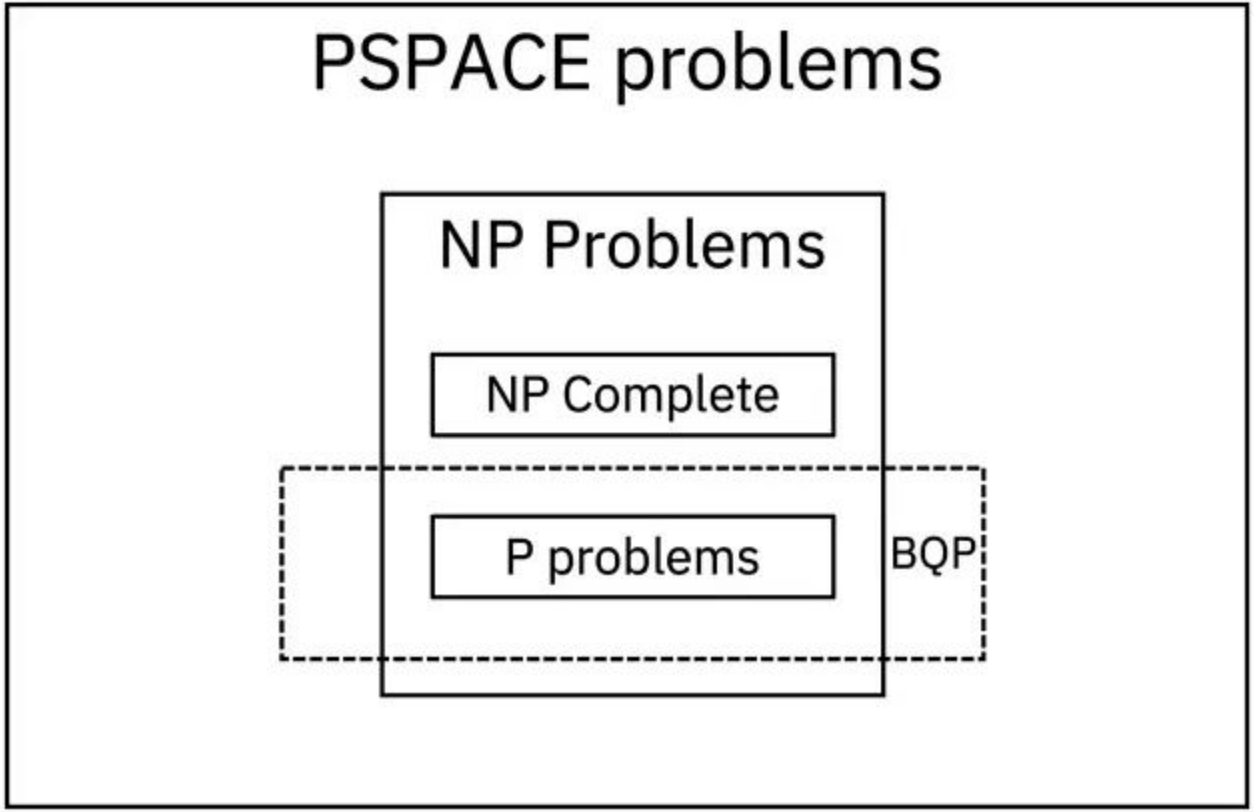
It is a complexity class that contains both P and NP, and it is considered a larger class than both of them.

In other words, any problem that can be solved in polynomial time or polynomial space is also a PSPACE problem.

P Space

PSPACE is a complexity class that contains decision problems solvable using a polynomial amount of memory, regardless of the time it takes.

These problems may require an exponential amount of time, but they are constrained by a polynomial limit on space usage.


PSPACE problems
NP Problems
NP Complete
P problems
BQP

BQP

BQP or Bounded-Error Quantum Polynomial Time, is a class of decision problems that is solvable by a Quantum Computer in polynomial time with an error probability of at most ⅓ for all instances.

This means that most quantum computing algorithms have to be run multiple times to obtain an accurate answer.

BQP

Factoring Integers (Shor's Algorithm)

Problem: Given an integer N, find its prime factors.

Classical Complexity: Believed to be in NP (no polynomial-time classical algorithm is known).

Quantum Advantage: Shor's algorithm solves it in polynomial time, making it a classic example of a problem in BQP.

BQP

Discrete Logarithm Problem

Problem: Given $g^x \bmod p = h$, find x, where g and h are known, and p is a prime number.

Classical Complexity: Exponential time for large numbers.

Quantum Advantage: Shor’s algorithm also provides a polynomial-time solution.

BQP

Grover’s Search Algorithm
simulation of quantum systems

Hamiltonian Energy Estimation

Quantum Circuit Satisfiability

Graph Isomorphism Problem

Quantum Walk-Based Search Problems

Linear Systems Problem (HHL Algorithm)

Principal Component Analysis (Quantum PCA)

Quantum Approximate Optimization Problems (QAOA)

Phase Estimation Problems

You can find more BQP problems in https://quantumalgorithmzoo.org/

References

https://qc-at-davis.github.io/QCC/Complexity-Classes/Complexity-Classes.html

https://medium.com/@saarbk/the-frontiers-of-quantum-computing-complexity-classes-and-the-relation-between-bqp-and-np-cfdcd74ace68

Key TakeAways

Last Notes

You have probably encountered several fancy terms and technologies that you may have never heard about.

Don't be worried!.

Quantum computing is a highly specialized field, and many things will not make sense the first time you read about them. These things require a lot of time to understand intuitively, but it is gratifying when you do so.

Last Notes

People didn’t realize how powerful deep learning was until it became practical. It was hard to judge its true power just from theory.

The same goes for quantum computing.

Last Notes

There will be “winters” in Quantum Computing!

Future Topics to Explore

Quantum Cryptography

Quantum Error Correction

Quantum Hardware

Fault Tolerance and Robust Quantum Computing

"The universe operates on quantum principles—so why shouldn't your curiosity and imagination?"

ENTANGLEMENT KEEPS PARTICLES CONNECTED NO MATTER HOW FAR APART THEY ARE.
LET'S STAY CONNECTED IN A QUANTUM WAY!

Signing off Sumaiyah Zahid

If you think you understand quantum mechanics, you don't understand quantum mechanics.

— Richard P. Feynman —