Quantum Chemistry on Quantum Computers

#7 Computational Complexities

Kenji Sugisaki[1,2,3]

[1]*Department of Chemistry, Graduate School of Science, Osaka City University, Japan*
[2]*JST PRESTO, Japan*
[3]*CQuERE, TCG CREST, India*

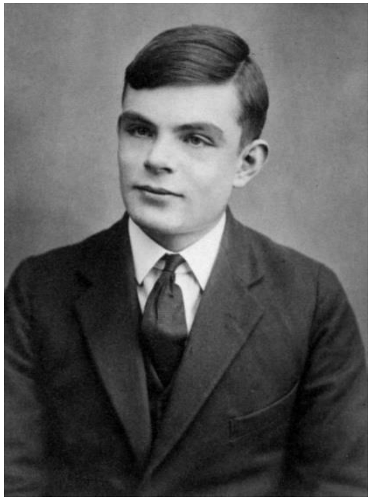
Motivations

1) Sum of two integers of *N*-digits
2) Product of two integers of *N*-digits
3) Find the greatest common divisor of two integers of *N*-digits
4) Factorization of an *N*-digit integer
5) Sort *N* integers in ascending order

Some problems are easy but others are not.

- ***How to quantify the difficulty of problems?***
  - Investigate the scaling of the number of elementary steps against the problem size *N.*
  - E.g., How much the computational cost will increase if the problem size *N* becomes twice?
- ***How to classify problems in terms of complexities?***

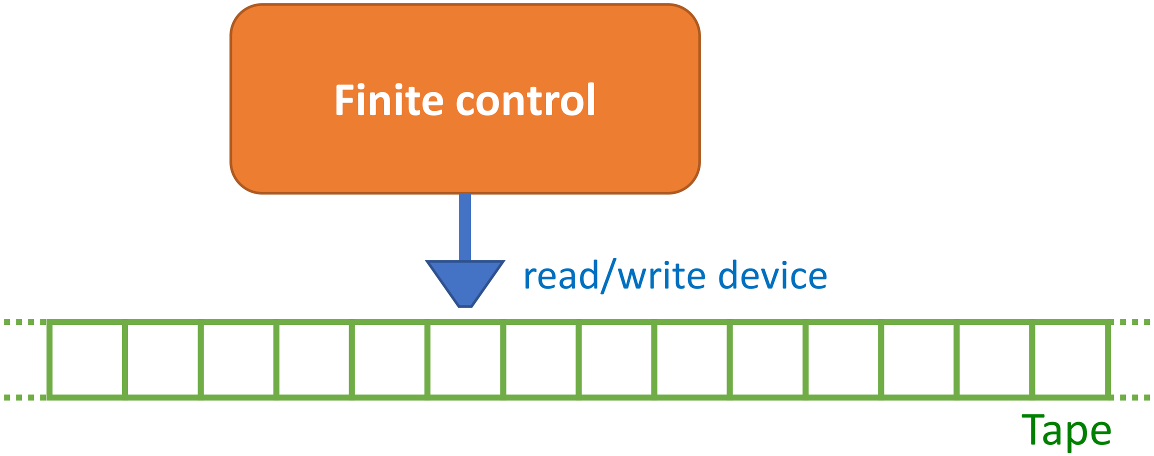
Turing Machine



Alan Turing

Photo taken from Wikipedia:
https://en.wikipedia.org/wiki/Alan_Turing

- Mathematical model of computation
- Consists of a control unit with a finite-state (CPU) and a tape unit with a head that moves from one tape cell to a neighboring one in each unit of time (Memory)



1) Read the symbol *X*
2) Write the symbol *Y* according to the state of CPU *S* and the symbol *X*
3) Unmove or move the tape to left or right
4) Change the state to *S'*
5) Return to step 1 or halt

Big *O* Notations

Characterize functions according to their growth rates

$f = O(g)$ if there exists a constant c such that $f(n) \le c \cdot g(n)$ for every sufficiently large n.

$$f(n) = 2n + 3 \qquad f = O(n)$$

$$f(n) = n^2 + 4n \longrightarrow f = O(n^2)$$

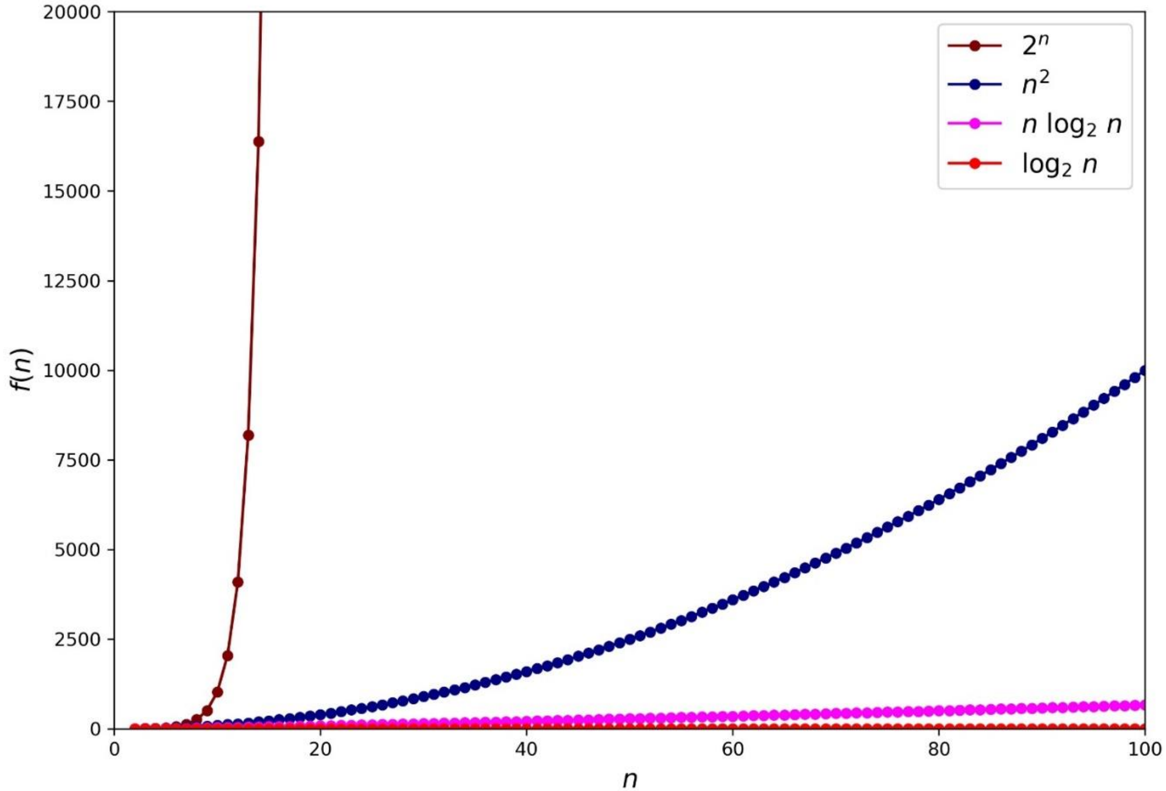
$$f(n) = 2^n + n \qquad f = O(2^n)$$

Big *O* tilde notation: Ignores logarithmic factors

$$\tilde{O}\big(h(n)\big) = O(h(n)\log n)$$

Example: Fast Fourier transformation of length *n* can be solved in $O(n \log n) = \tilde{O}(n)$

Relationship between size, order and computational steps



- Practically solvable problems should be in the polynomial scaling.
- Exponential scaling is usually very hard to solve, except for small problem size n.
- From the practical point of view, problems in exponential scaling is *not always* harder than that in polynomial scaling.

E.g., 1.00001^n vs. n^{10}

22025 vs. 10^{60} for $n = 10^6$

Complexity classes

A set of functions those can be computed within given resource bounds.
We will pay attention to *Boolean functions*, those have only one bit of output.

Time complexity … computational time (number of steps)
Space complexity … memory space

☆ The class **P** (polynomial-time)

Decision problems those can be solved by a deterministic Turing machine using a **polynomial amount of computation time**, or polynomial time.

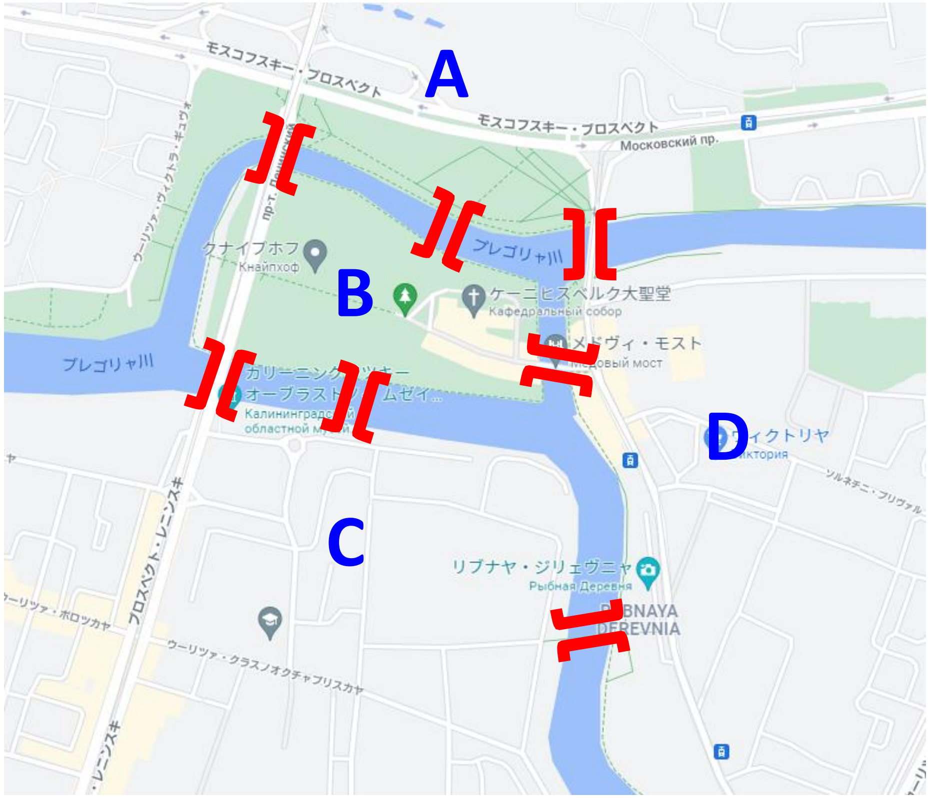
$O(n^c)$, where c is some constant $c > 0$.

Summation, multiplication, etc.
Primality test (AKS algorithm)
Finding the Eulerian path

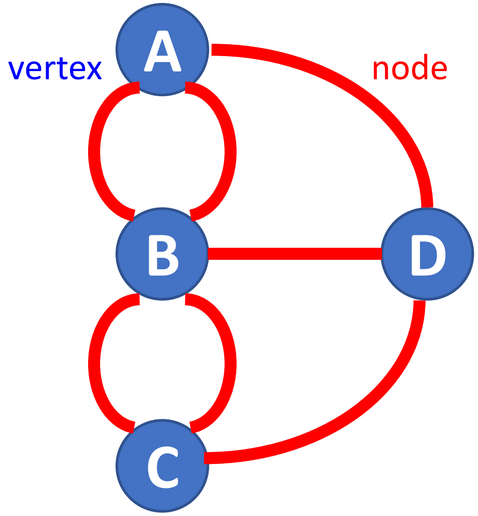
* Note that the computational cost of the AKS algorithm is $\tilde{O}(\log^{21/2} n)$ in the original proposal, and it is quite difficult to apply to large integers.

Königsberg seven bridge problem and Eulerian path

Kaliningrad, Russia



Is it possible to devise a walk through the city that would cross each of those bridges once and only once?



Leonhald Euler

Immanuel Kant,
a philosopher

David Hilbert,
a mathematician

To cross each of node once and only once,

1) The graph is connected.
2) The number of vertices with odd number of nodes should be 0 or 2.

Complexity classes

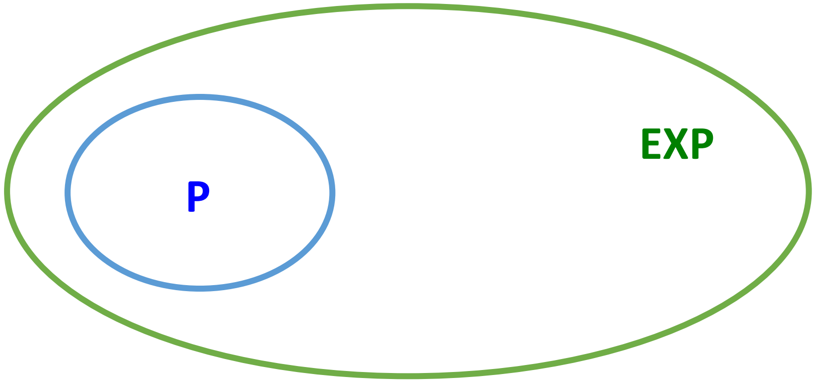
☆ The class **EXP** (exponential-time)

Decision problems those can be solved by a deterministic Turing machine using a **exponential amount of computation time**, or exponential time.

$2^{O(p(n))}$, where $p(n)$ is a polynomial function of n.

Finding the Hamiltonian cycle

It is obvious that $\mathbf{P} \subseteq \mathbf{EXP}$.



The class NP (non-deterministic polynomial-time)

1) Allows non-deterministic choices (e.g., using dices, choice based on human intuition)

2) Use the computational cost of the luckiest case

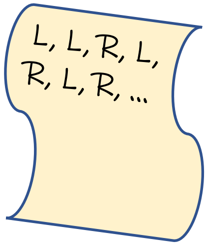
3) Ignore if the answer is "No"

→ Solvable in polynomial time under three conditions above.

For problems in class **NP**, we can verify whether the instance is correct or not using deterministic Turing machine in polynomial time.

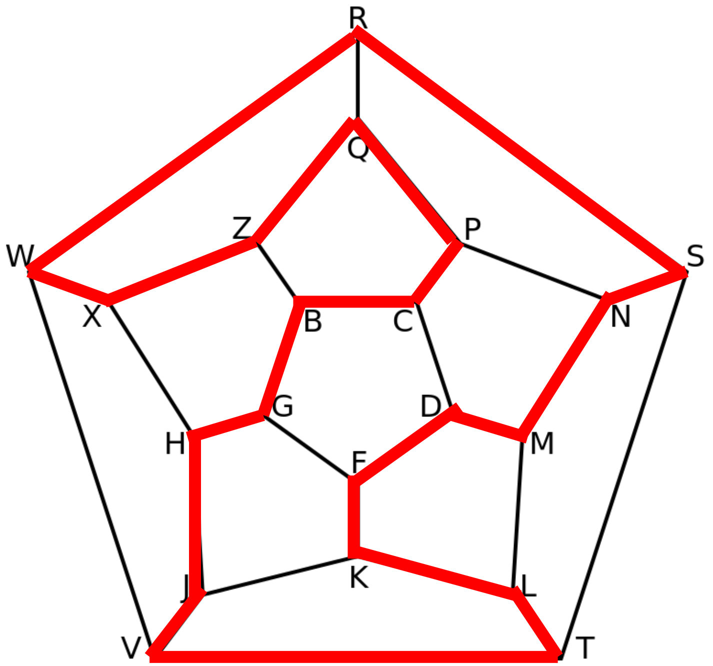
Existence of the Hamiltonian cycle
Satisfiability problem (SAT)

$$\mathbf{NP} \subseteq \mathbf{EXP}$$



Longleat hedge maze, UK
Photo taken from Wikipedia:
https://en.wikipedia.org/wiki/Longleat

Icosian game and Hamiltonian cycle



Icosian game

Finding a cycle along the edges of a dodecahedron such that every vertex is visited a single time, and the ending point is the same as the starting point.

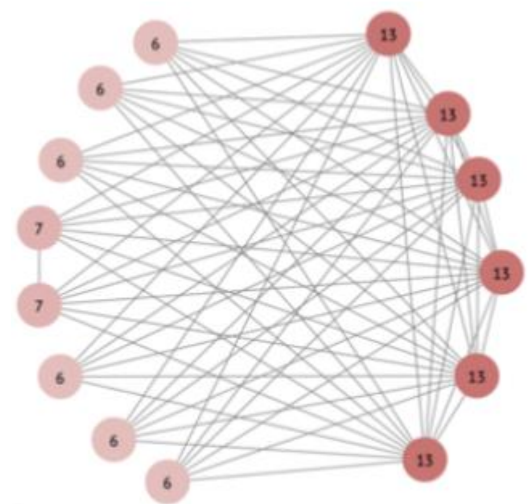


Figure taken from J. Sleegers, D. van den Berg, Looking for the hardest Hamiltonian cycle problem instances. In *Proceedings of the 12th International Joint Conference on Computational Intelligence – ECTA*, pp. 40-48, 2020. DOI: 10.5220/0010066900400048

Finding a Hamiltonian cycle is quite difficult for the graphs with large number of vertices/edges.

Once the Hamiltonian cycle is given, verification is very easy.

Satisfiability problem (SAT)

A propositional logic formula, also called Boolean expression, is built from variables, operators AND ($\wedge$), OR ($\vee$), NOT ($\neg$), and parentheses.

$$\underbrace{(x_1 \vee x_2)} \wedge \underbrace{(x_1 \wedge \overline{x_3})} \wedge \underbrace{(x_2 \vee x_3)}$$

Literals

Clauses

or equivalently, (x_1 or x_2) and (x_1 and not x_3) and (x_2 or x_3)

TRUE: $x_1 = 1$, $x_2 = 1$, and $x_3 = 0$

n-SAT problem: each clause has at most *n* literals.

Completeness and Hardness

☆ **NP**-complete

The problem Q is in **NP**-complete if

- Q belongs to class **NP** and
- Any problems in class **NP** can be translated to the particular instance of problem Q in polynomial time

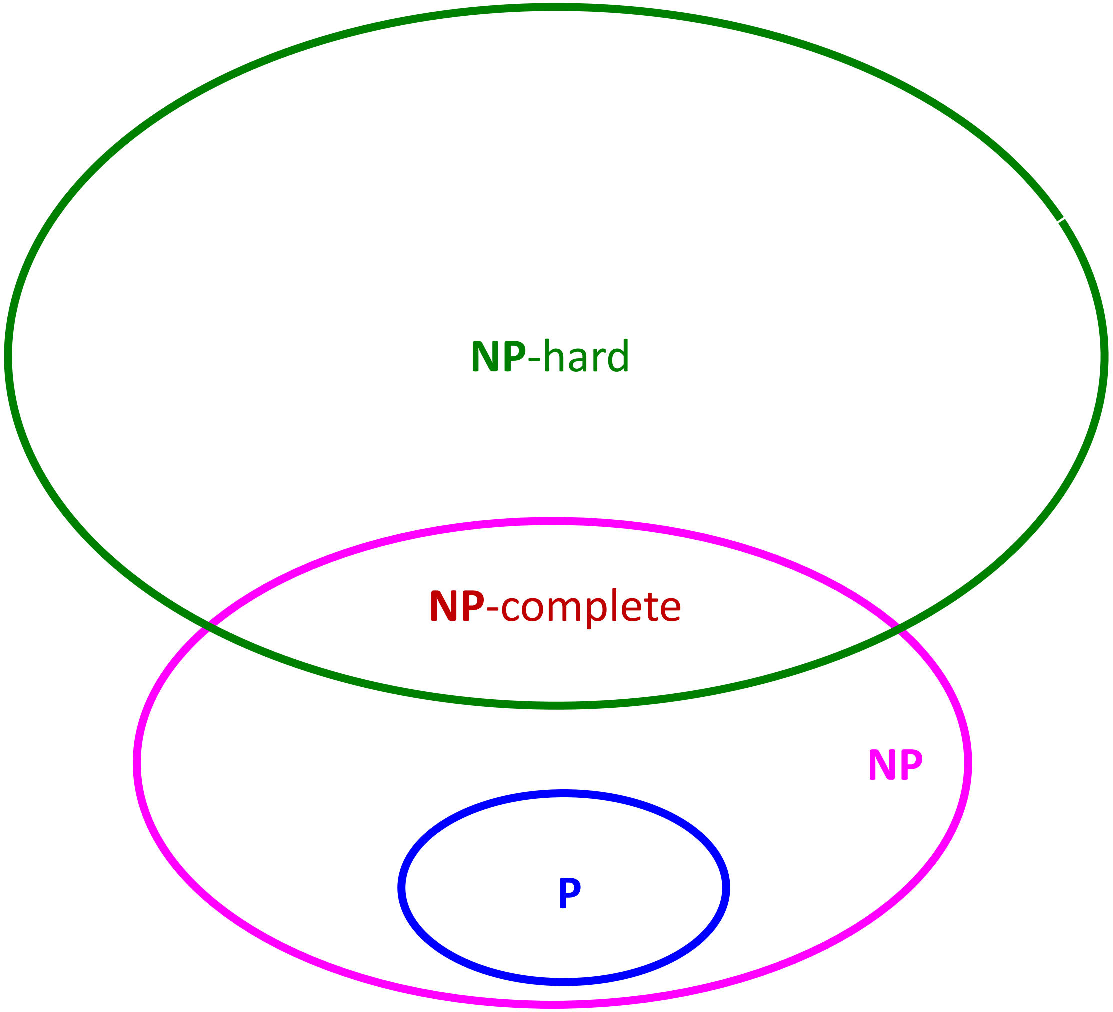
NP-complete problem is the most difficult problem in the class **NP**

Existence of the Hamiltonian cycles
3-SAT

☆ **NP**-hard

The problem Q is in **NP**-hard if Q is at least as hard as the hardest problems in **NP**

Computational complexity class with a Venn diagram



Here, we assume that **P ≠ NP**

Class BPP

Probabilistic Turing machine (PTM)

A non-deterministic Turing machine with two transition functions δ_0 and δ_1.
To execute a PTM, we choose in each step with probability 1/2 to apply the transition δ_0 and with probability 1/2 to apply δ_1.

☆ The class **BPP** (bounded-error probabilistic polynomial-time)

Decision problems those can be solved by a **probabilistic Turing machine** using a **polynomial amount of computation time**, or polynomial time, with an error probability bounded away from 1/3 for all instances.

Complexity class for quantum computers

☆ The class **BQP** (bounded-error quantum polynomial-time)

Decision problems those can be solved by a **quantum computer** using a **polynomial amount of computation time**, or polynomial time, with an error probability bounded away from 1/3 for all instances.

The class BQP is the quantum analog of BPP

☆ The class **QMA** (Quantum Merlin Arthur)

For problems in class **QMA**, we can verify whether the instance is correct or not using a quantum computer in polynomial time.

The class QMA is the quantum analog of NP

k-Local Hamiltonian

$H = \sum_j H_j$, where each H_j acts only on $\leq k$ qubits

(yes case) *H* has an eigenvalue less than *a*
(no case) All of the eigenvalue of *H* are larger than *b*

where $b - a \geq 1/\mathrm{poly}(N)$

5-local Hamiltonian is **QMA**-complete

A. Y. Kitaev, A. H. Shen, M. N. Vyalyi, Classical and Quantum Computation. Graduate Studies in Mathematics Volume 47, American Mathematical Society, 1999.

3-local Hamiltonian is **QMA**-complete

J. Kempe, O. Regev, *Quantum Info. Comp.* **2003**, *3*, 258–264.

2-local Hamiltonian is **QMA**-complete

J. Kempe, A, Kitaev, O. Regev, *SIAM J. Comput.* **2006**, *35*, 1070–1097.

1-local Hamiltonian is **P**

Relationships of complexity classes

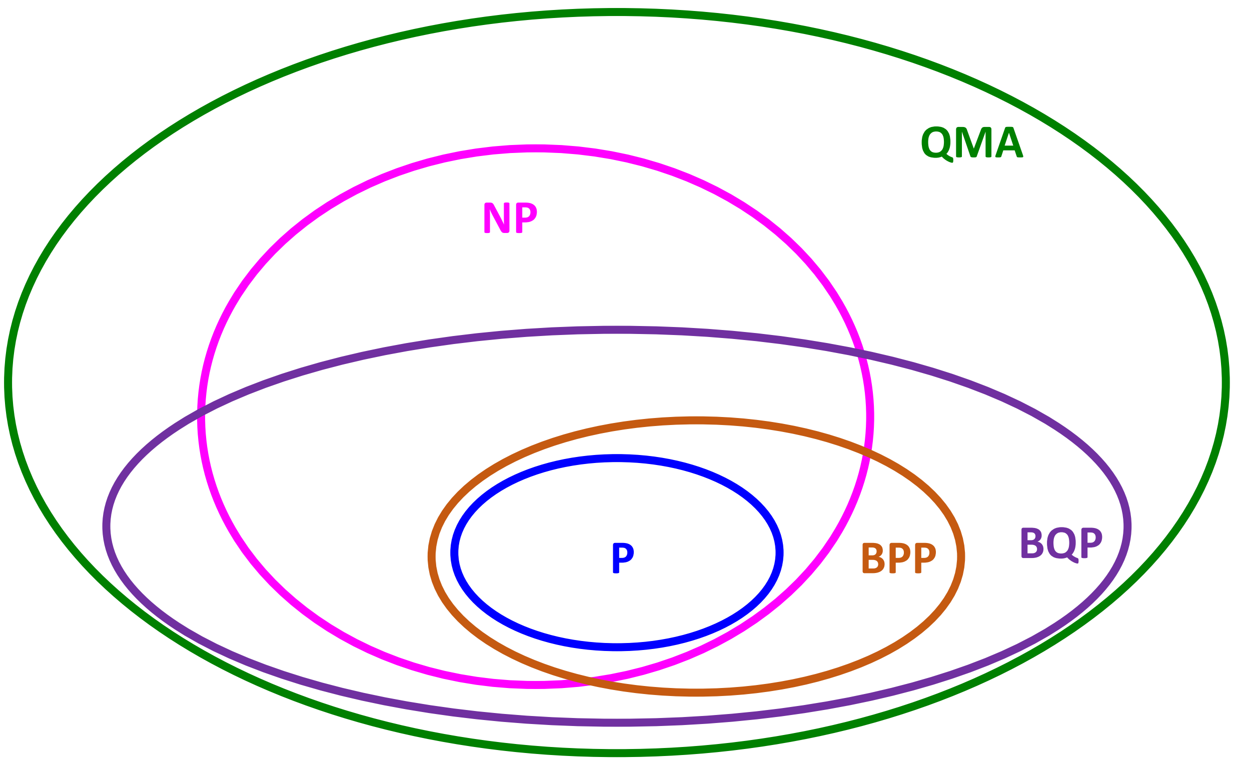
$\mathbf{P} \subseteq \mathbf{BPP} \subseteq \mathbf{BQP}$

Problems in class **P** can be solved in polynomial time using a quantum computer

$\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{QMA}$

$\mathbf{BQP} \subseteq \mathbf{QMA}$

$\mathbf{NP} \not\subseteq \mathbf{BQP}$



Complexity classes of electronic structure problems

THE JOURNAL OF CHEMICAL PHYSICS **141**, 234103 (2014)

On the NP-completeness of the Hartree-Fock method for translationally invariant systems

James Daniel Whitfield[1,a)] and Zoltán Zimborás[2,3,b)]
[1] *Vienna Center for Quantum Science and Technology, Department of Physics, University of Vienna, Boltzmanngasse 5, Vienna, Austria*
[2] *Department of Computer Science, University College London, Gower Street, WC1E 6BT London, United Kingdom*
[3] *Department of Theoretical Physics, University of the Basque Country UPV/EHU, P.O. Box 644, E-48080 Bilbao, Spain*

arXiv:2103.08215

Electronic Structure in a Fixed Basis is QMA-complete

Bryan O'Gorman* Sandy Irani† James Whitfield** Bill Fefferman‡

March 16, 2021

Complexity classes of density functional theory

LETTERS
PUBLISHED ONLINE: 23 AUGUST 2009 | DOI: 10.1038/NPHYS1370
nature physics

Computational complexity of interacting electrons and fundamental limitations of density functional theory

Norbert Schuch[1]* and Frank Verstraete[2]*

Finding an universal functional is **QMA**-complete

PRL **98,** 110503 (2007) PHYSICAL REVIEW LETTERS week ending 16 MARCH 2007

Quantum Computational Complexity of the *N*-Representability Problem: QMA Complete

Yi-Kai Liu,[1] Matthias Christandl,[2] and F. Verstraete[3,4]
[1]*Computer Science and Engineering, University of California, San Diego, California, USA*
[2]*Centre for Quantum Computation, Centre for Mathematical Sciences, DAMTP, University of Cambridge, Wilberforce Road, Cambridge CB3 0WA, United Kingdom*
[3]*Facultät für Physik, Universität Wien, Boltzmanngasse 5, A-1090 Wien, Austria*
[4]*Institute for Quantum Information, Caltech, Pasadena, California, USA*
(Received 17 September 2006; published 16 March 2007)

Complexity classes of density functional theory

New Journal of Physics
The open access journal at the forefront of physics
Deutsche Physikalische Gesellschaft DPG | IOP Institute of Physics

Computational complexity of time-dependent density functional theory

J D Whitfield[1], M-H Yung[2,3], D G Tempel[3], S Boixo[4] and A Aspuru-Guzik[3]
[1] Vienna Center for Quantum Science and Technology, University of Vienna, Department of Physics, Boltzmanngasse 5, Vienna A-1190, Austria
[2] Center for Quantum Information, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, People's Republic of China
[3] Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA 02138, USA
[4] Google, Venice Beach, CA 90292, USA
E-mail: JDWhitfield@gmail.com

Received 19 May 2014, revised 8 July 2014
Accepted for publication 14 July 2014
Published 15 August 2014
New Journal of Physics **16** (2014) 083035
doi:10.1088/1367-2630/16/8/083035

Computing the time-dependent Kohn–Sham potential is **BQP**

Complexity classes of variational quantum algorithms

PHYSICAL REVIEW LETTERS **127,** 120502 (2021)

Editors' Suggestion

Training Variational Quantum Algorithms Is NP-Hard

Lennart Bittel[*] and Martin Kliesch[†]

Quantum Technology Group, Heinrich Heine University Düsseldorf, 40225 Düsseldorf, Germany

(Received 25 February 2021; accepted 10 August 2021; published 17 September 2021)

Review papers

☆ General reviews

J. Watrous, Quantum Computational Complexity. In *Encyclopedia of Complexity and Systems Science*, Springer, New York, 2009.

☆ Computational complexities of quantum chemical calculations

J. D. Whitfield, P. J. Love, A. Aspuru-Guzik, *Phys. Chem. Chem. Phys.* **2013**, *15*, 397–411.

☆ BQP-complete problems

P. Wocjan, S. Zhang, arXiv:quant-ph/0606179.

☆ QMA-complete problems

A. D. Bookatz, *Quantum Inf. Comp.* **2014**, *14*, 361–383; arXiv:1212.6312.

☆ Book

S. Arora, B. Barek, *Computational Complexity. A Modern Approach*, Cambridge University Press, 2009.