A *real world* example

(credit to `https://arxiv.org/pdf/quant-ph/9809016.pdf`)
We want to find the factors of $N = 21$

1. pick an integer $a < N$ ($a = 11$);
2. pick m such that $N^2 < 2^m < 2 * N^2$. In our case $m = 9$;
3. compute using *quantum parallelism* $f(k) = a^k \mod N$
   for all integers from 0 to $2^m - 1$ (*i.e.*, 511)
   (a) start with an m-qubit state $|00...0\rangle$
   (b) apply the Hadamard transformation to get a superposition

$$\frac{1}{\sqrt{2^m}}(|00...0\rangle + |00...1\rangle + ... + |11...1\rangle = \frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} |k\rangle$$

corresponding to all integers $0 \leq k < 2^m$

(c) add a 5-bit register $|0\rangle$ for the *output* of the function.

the function is encoded in the quantum state

$$\frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} |k, f(k)\rangle$$

in our example a total of 14 qubits are required
(9 for k and 5 for the output of the function)

4. we need a state whose amplitude has the same period as f
   (a) measure the last $\lceil \log_2 N \rceil$ qubits
   (b) a random value u is obtained
   what is interesting is just the effect the measurement has on our set of superpositions.

The state after measurement is

$$C \sum_k g(k) \, |k, u\rangle$$

C is a scale factor we can ignore whereas

$$g(k) = \begin{cases} 1 & \text{if } f(x) = u \\ 0 & \text{otherwise} \end{cases}$$

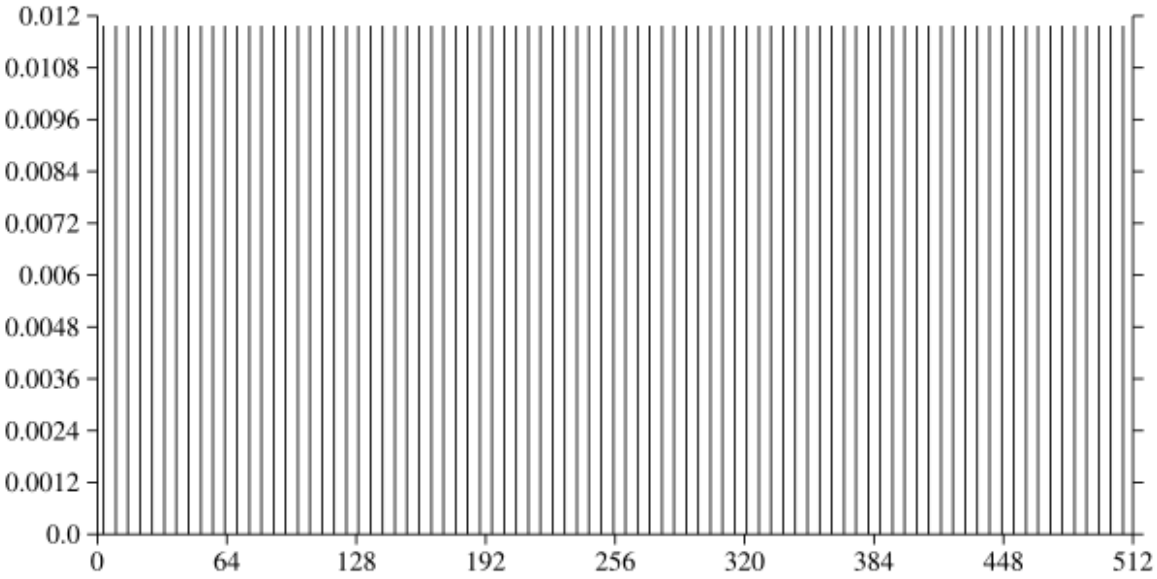
- the k's that actually appear in the sum differ from each other by multiples of the period;
- if we could measure **two** successive k's in the sum it would be trivial to get the period;
- unfortunately the laws of quantum physics permit only one measurement!

5. Suppose that random measurement of the superposition gives 8.

$$x(k) = \begin{cases} \sqrt{r/2^m} & \text{if } k \mod r = 3 \\ 0 & \text{otherwise} \end{cases}$$

(3 is the min $k > 0$ such that $11^k \mod 21 = 8$)

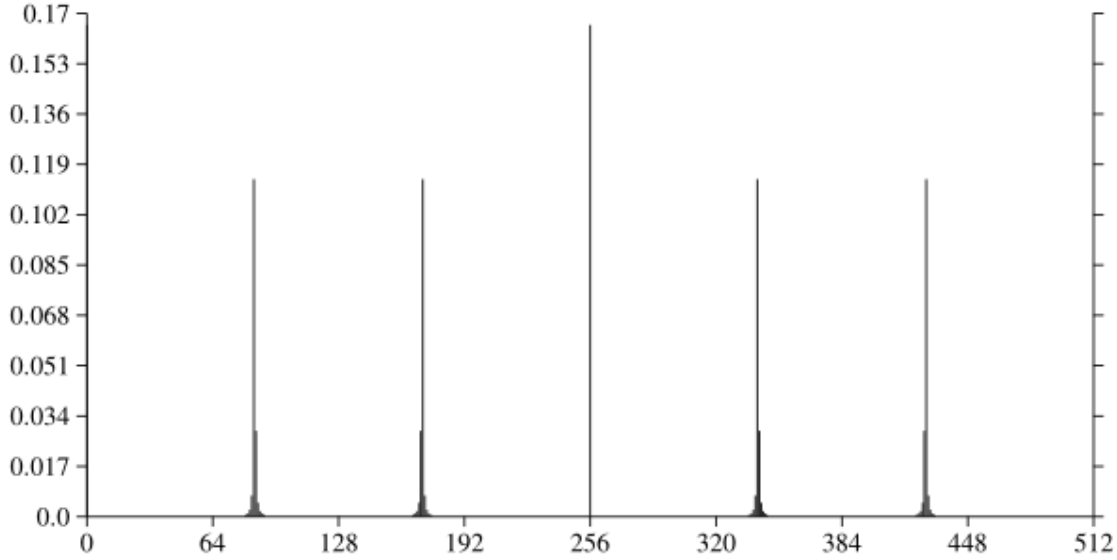
the state clearly shows the periodicity of f



6. if 2^m is an exact multiplier of r, the result of the QFT is **exactly**

$$c(k) \begin{cases} \neq 0 & \text{if } k \mod \frac{2^m}{r} = 0 \\ = 0 & \text{otherwise} \end{cases}$$

(the Fourier transform of a state with period r is a state with period $\frac{2^m}{r}$). Otherwise, the transform *approximates* the exact case and most of the amplitude is attached to integers close to multiples of $\frac{2^m}{r}$.



7. measure the state of the QFT in the standard basis.
   (suppose the result is $v = 427$)
   When the period r is a power of 2, then $v = j\frac{2^m}{r}$ for some j and it is trivial to find r.
   When the period is **not** a power of 2, it is necessary to resort to the *continued fraction expansion* of $\frac{v}{2^m}$
   This gives as a result $r = 6$ (we are "lucky" is an even number)
   Remember that either $11^{\frac{6}{2}} - 1 = 1330$ or $11^{\frac{6}{2}} + 1 = 1332$ must have a common factor with $N = 21$.
   Since gcd(21,1330)=7 and gcd(21,1332)=3, we have done.
   Easy, is it not?

What could go wrong?

1. the value v was not close enough to a multiple of $\frac{2^m}{r}$;
2. the period r and the multiplier j could have had a common factor so that p was actually a factor of the period not the period itself;
3. the period r is odd;
4. few other...

Shor shows that few repetitions of the algorithm yields a factor of N with probability > 0.5.

Two remarks about Shor's algorithm

1. The best known classical factoring algorithms (*e.g.*, Quadratic Field Sieve and Number Field Sieve) do much better than period finding methods attack by exploiting additional structures in the problem;
2. what we can **currently prove** is that Shor's algorithm achieves an exponential speedup over any classical factoring algorithm that works via reduction to period-finding.

Grover's algorithm

A large class of problems can be specified as search problems

- *find some x_s in a set of possible solutions such that statement $P(x_s)$ is true* (x_s can be, but it is not necessarily, unique)
- a basic NP-hard search problem
  - in the general case the fastest known classical algorithm is the exhaustive search;
  - with N possible inputs and n the number of bits required to represent the input, the search requires $O(2^n)$ steps
- Grover's algorithm solves the problem in $O(2^{\frac{n}{2}})$ steps.

- Let U_P be the quantum gate that implements the classical function $P(x)$ for testing the truth of the statement

$$U_P : |x, 0\rangle = |x, P(x)\rangle$$

- Compute P for all possible inputs x_i, by applying U_p to a register containing the superposition $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x\rangle$ (plus the register for the "output")

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x, P(x)\rangle$$

  suppose there is a single x_0 such that $P(x_0)$

- the amplitude of such a state is $\frac{1}{\sqrt{2^n}}$;
- the probability that a random measurement reports x_0 is only 2^{-n} **exactly** as in the classic case. What could be our move?

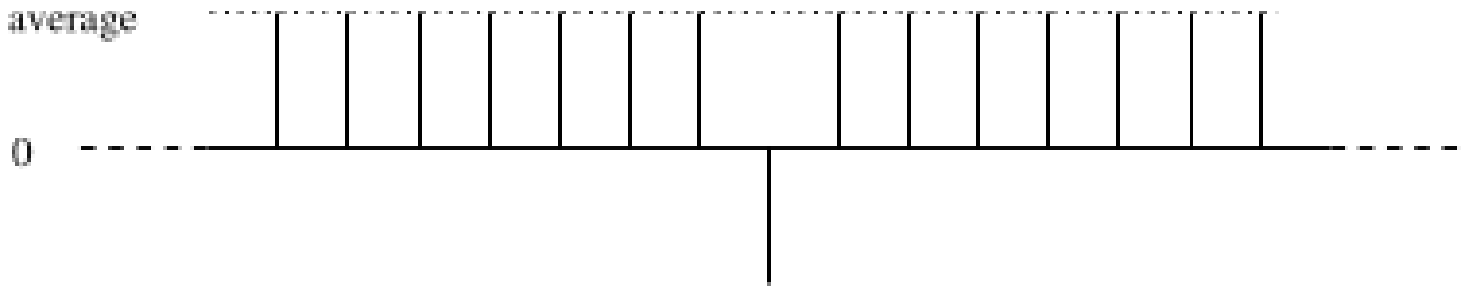
- change the quantum state so as to greatly increase the amplitude of $|x_0, 1\rangle$ and decrease the amplitude of all $|x, 0\rangle$
  - thanks to the amplitude change, if we measure the last qubit (*i.e.*, the output), there is a high probability that the result will be 1;
  - if this is the case, a further measurement of the remaining bits (*i.e.*, of the input) provides the *solution*.
  - how do we change the quantum state to that purpose?
- Grover's algorithm is the answer

1. prepare a register of qubits containing a superposition of all possible values $x_i \in [0...2^n - 1]$;
2. compute $P(x_i)$ on this register;
   - this may sound *weird* but, actually, it is (relatively) simple (no more difficult w.r.t. the classic case, although *slower*)
3. *change amplitude* a_j to $-a_j$ for x_j such that $P(x_j) = 1$;
4. *apply inversion about the average* to increase amplitude of x_j with $P(x_j) = 1$;
5. repeat $\frac{\pi}{4}\sqrt{2^n}$ times the steps 2 through 4;
6. read the result.

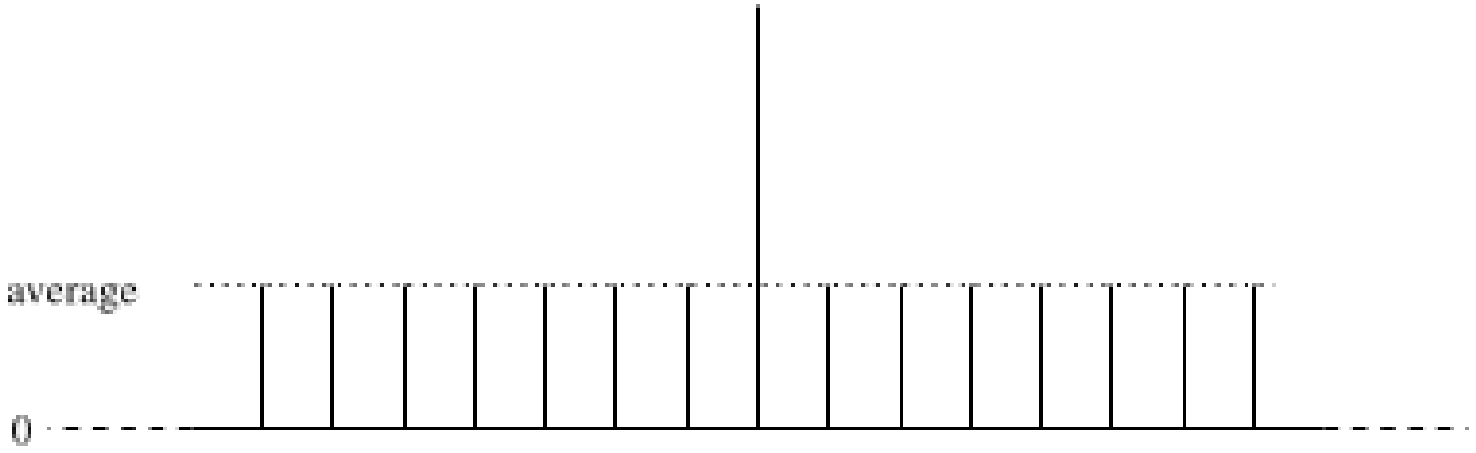
Steps 2 through 4 constitute the Grover's *diffusion operator*.

It is not apparent why this algorithm works... but a couple of pictures may help

- After the change of amplitude we have a situation like this



- and after the inversion about the average we have



The diffusion operator $\mathcal{D}$ has the following effect

$$\mathcal{D} : \sum_j \alpha_j \left|x_j\right\rangle \rightarrow \sum_j (2\mu - \alpha_j) \left|x_j\right\rangle$$

on the superposition x_j (μ is the mean amplitude).

- any amplitude $\mu + \delta$ gets turned into $\mu - \delta$;
- most of the amplitudes are a tiny bit larger than the mean
  - they will become a tiny bit less than the mean by applying $\mathcal{D}$;
- the state we are looking for will be affected more strongly
  - its amplitude is (significantly) less than the mean, and so will become (significantly) greater than the mean!

The diffusion operator causes an interference effect on the states which skims an amplitude of $\frac{1}{\sqrt{n}}$ from all the wrong answers and add it to the right one(s).

It is possible to prove that Grover's algorithm is "optimal"
(up to a constant factor).
Moreover, if there is only a single x_0 such that $P(x_0)$ is *true*

- after $\frac{\pi}{8}\sqrt{2^n}$ iterations the failure rate is 0.5;
- after $\frac{\pi}{4}\sqrt{2^n}$ iterations the failure rate drops to 2^{-n};
- after $\frac{\pi}{2}\sqrt{2^n}$ iterations the failure rate is close to 1!

This should be no surprising:

- quantum procedures are unitary transformations (in other words, rotations in a complex space);
- repeated applications of a quantum transform may rotate the state closer and closer to the desired state;
- eventually the rotation goes beyond the desired state and then the *distance* (re)starts to increase if there are further rotations.

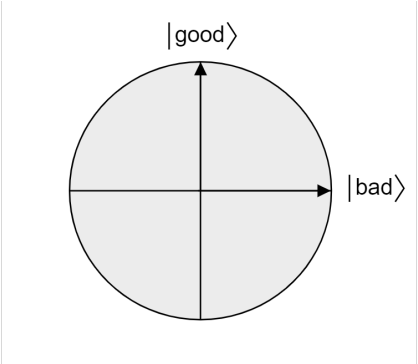
A geometrical perspective on Grover's algorithm

(`https://learn.microsoft.com/en-us/azure/quantum/concepts-grovers`)
Suppose $|bad\rangle$ is the superposition of all states such that $P(x)$ is false;

$$|bad\rangle = \frac{1}{\sqrt{2^n - 1}} \sum_{x:P(x)=0} |x\rangle$$

$|good\rangle$ is the state such that $P(x)$ is true; $|bad\rangle$ and $|good\rangle$:

- are mutually exclusive sets and orthogonal;
- form the orthogonal basis of a plane in the vector space



Suppose $|\psi\rangle$ is an arbitrary state that lives in the plane spanned by $|good\rangle$ and $|bad\rangle$. Then

$$|\psi\rangle = \alpha\,|good\rangle + \beta\,|bad\rangle$$

(α and β are real numbers). Let's introduce the reflection operator $\mathcal{R}_{|\psi\rangle} = 2\,|\psi\rangle\,\langle\psi| - \mathcal{I}$ (reflection w.r.t. the direction of $|\psi\rangle$).

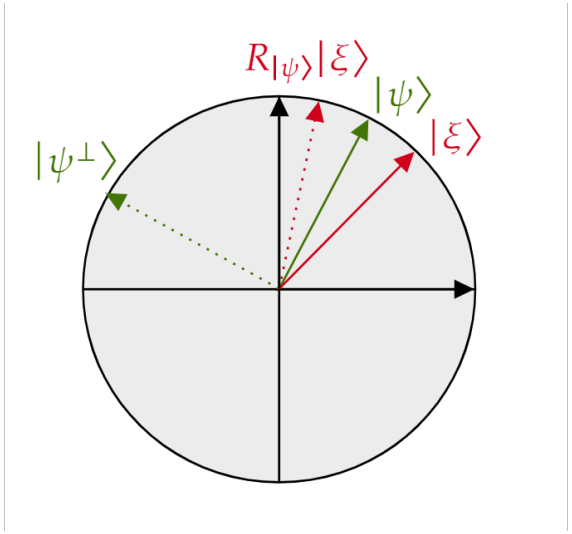
If you take the orthogonal basis of the plane formed by $|\psi\rangle$ and its orthogonal complement $|\psi^{\perp}\rangle$, any other state $|\xi\rangle$ can be decomposed in

$$|\xi\rangle = \mu\,|\psi\rangle + \nu\,|\psi^{\perp}\rangle$$

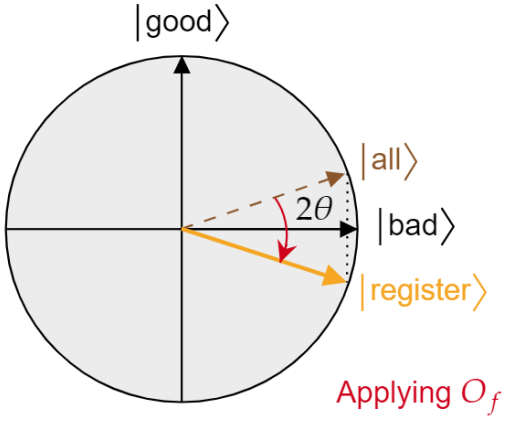
if we apply $\mathcal{R}_{|\psi\rangle}$ to $|\xi\rangle$

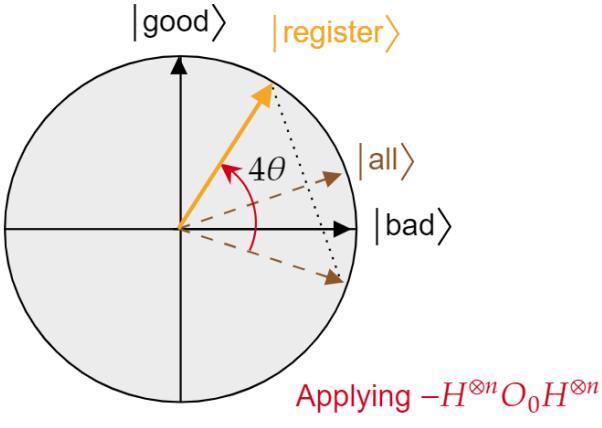
$$\mathcal{R}_{|\psi\rangle}\,|\xi\rangle = \mu\,|\psi\rangle - \nu\,|\psi^{\perp}\rangle$$

That is $\mathcal{R}_{|\psi\rangle}$ inverts the component orthogonal to $|\psi\rangle$ but leaves the $|\psi\rangle$ component unchanged



Each Grover's iteration is a composition of $\mathcal{R}_{|bad\rangle}$ and $\mathcal{R}_{|all\rangle}$





The combined effect of each Grover's iteration is a counterclockwise rotation of an angle 2θ.

- θ is simply the angle between $|all\rangle$ and $|bad\rangle$.
  Assuming there is a single $|good\rangle$ state

$$\theta = \arccos\left(\langle all| \, |bad\rangle\right) = \arccos\left(\sqrt{\frac{2^n - 1}{2^n}}\right)$$

- the angle between the state of the register and the $|good\rangle$ state decreases with each iteration
- the probability that the measurement shows the *good* answer is $|\langle good| \, |register\rangle|^2$
- it turns out to be a function of k (number of Grover's iterations)

$$P_{success} = \sin^2\left[(2k+1)\arccos\left(\sqrt{\frac{2^n - 1}{2^n}}\right)\right]$$

- $k_{optimal}$ is the smallest positive integer that (approximately) maximizes the success probability function

$$k_{optimal} = \frac{\pi}{4 \arccos \sqrt{1 - 2^{-n}}} - \frac{1}{2} = \frac{\pi}{4}\sqrt{2^n} - \frac{1}{2} - O(\sqrt{2^{-n}})$$

The sequence of Grover's iterations can be viewed as a sort of "backward" quantum random walk:

- from a distributed state
- back to a state focused around the single correct component

A classical random walk explores an area proportional to the square root of the number of steps;

A quantum random walk explores an area proportional to the number of steps.

This difference provides a clue about the quadratic speedup.

QC in practice (?)

So far we have assumed (implicitly...) that the qubits are *error free*.

Actually qubits *fail* (classic bits fail as well!).

Besides that, the *decoherence* phenomenon also hinders QC, whereas there is no equivalent phenomenon in classical computing.

The decoherence phenomenon

When the quantum system is not perfectly isolated, the apparatus itself influences the QC procedure

- the transformations required during the computations are not longer *unitary*;
- the arising of the decoherence crushes the constructive and destructive interference;

The decoherence time must be sufficiently longer than the expected execution time of the computation.

Errors in computing

The existence of computational errors is not limited to the quantum *realm*.

- Since the 50's of the XX century, Von Neumann showed that a computer with noisy components can work reliably by employing *sufficient* redundancy;
- it is not straightforward to use the same approach to neutralize errors in a quantum computer;
- quantum error correction requires many (*e.g.*, one thousand) additional imperfect *physical* qubits to provide one *reliable* logical qubit (like those we considered previously).

Quantum Algorithms

1. Amplitude Amplification (Grover's search). Quadratic speedup;
2. Phase Estimation (eigenvalues of a unitary transformation). Quadratic speedup;
3. Quantum Fourier Transform. Exponential speedup;
4. Hamiltonian simulation (the original Feynman's idea). Exponential speedup.

Is quadratic speedup enough?
(credit to Hoefler Torsten
`https://www.youtube.com/watch?v=0QuUFyCKUgY`)

What if a Quantum Computer

- had 10000 logical qubits (remember, this means from 10 to 1000 more physical bits);
- had 10 μs logical gate time;
- could simultaneously perform any gate operation on all qubits in a single cycle;
- had all-to-all connectivity.

What is the I/O bandwidth?

$$10000 \times 1(\text{bit/gate})/(10 \times 1^{-6}\text{s}) = 1\text{Gbit/s}$$

How fast can we evaluate something on a QC?

Let us consider just multiplications (as if addition were free) and ignore all overheads. We end up with

Operation throughput	**Quantum Computer**	**GPU**
16-bit floating point	10.5 Kop/s	195 Top/s
32-bit integer	0.83 Kop/s	9.75 Tops/s
boolean logical	235 Kop/s	4992 Tops/s

Grover's algorithm requires $\sqrt{N}$ invocations instead of N.
Since the GPU is (very roughly) $\sim 2 \times 10^{10}$ faster.
What is the N break-even value?

$$\sim 4 \times 10^{20}$$

Summary and Conclusions

- Quantum Computing is **not** a new paradigm
  (it has been around for, at least, 40 years);
- QC does **not** evaluate all possible solutions at the same time;
- QC requires a suitable formulation of the problems to exploit the properties of the *superposition*;
- Only in a few cases QC may offer an *exponential* speedup;
- Current technology does not support practical usage of QC for solving real-world problems;
- However, never forget that the "idea" of digital cash has been floating around for about 25 years before Bitcoin appeared...

References

[1] R. Feynman. Simulating physics with computers. International Journal of Theoretical Physics, 21(6/7):467–488, 1982.

[2] R. Feynman. Quantum mechanical computers. Optics News, 11:11–20, 1985.

[3] A. Ekert, R. Josza. Quantum Computation and Shor's factoring algorithm. Rev. Mod. Phys. Vol. 68. No. 3. 1996.

[4] D. Di Vincenzo. Quantum Computation. Science. Vol. 270. 1995.

[5] B. M. Terhal. Quantum error correction for quantum memories, Rev. Mod. Phys. Vol. 87. No. 2. 2015.

[6] J. Preskill. Quantum computing and the entanglement frontier. Rapporteur Talk at the 25th Solvay Conference on Physics, Brussels https://doi.org/10.1142/8674 (World Scientific, 2012).

[7] J. Preskill. Quantum Computing in the NISQ era and beyond. https://arxiv.org/abs/1801.00862

[8] A. Matuschak, M. Nielsen, Quantum Computing for the Very Curious, https://quantum.country/qcvc, San Francisco (2019).

[9] `https://quantumalgorithmzoo.org/`

[10] `https://scottaaronson.blog/`