Quantum Computing based on Tensor Products
DFT and Factorization of Integers

Guillermo Morales Luna

Computer Science Section
CINVESTAV-IPN

E-mail: gmorales@cs.cinvestav.mx

5-th International Workshop on Applied Category Theory
Graph-Operad Logic

Agenda

Agenda

1. Quantum Computation of the Discrete Fourier Transform

2. Shor Algorithm
   - Quantum Algorithm to Calculate the Order of a Number

$[\![0, n-1]\!] = \{0, 1, \ldots, n-1\}$.

Given $f : [\![0, n-1]\!] \to \mathbb{C}$ its discrete Fourier transform is $\hat{f} : [\![0, n-1]\!] \to \mathbb{C}$

$$\forall j \in [\![0, n-1]\!] : \quad \hat{f}(j) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \exp\left(\frac{2\pi ijk}{n}\right) f(k). \qquad \left[i = \sqrt{-1}\right]$$

For

$$\mathbf{f} = \sum_{j=0}^{n-1} f(j)\mathbf{e}_j \in \mathbb{C}^n,$$

its discrete Fourier transform is

$$\mathrm{DFT}(\mathbf{f}) = \hat{\mathbf{f}} = \sum_{j=0}^{n-1} \hat{f}(j)\mathbf{e}_j \in \mathbb{C}^n.$$

DFT is linear transform and, w.r.t. the canonical basis, it is represented by the unitary matrix $\mathrm{DFT} = \frac{1}{\sqrt{n}}\left(\exp\left(\frac{2\pi ijk}{n}\right)\right)_{jk}$

DFT^H coincides with DFT except that the exponents in each entry have sign "–".

In particular,

$$\forall j \in [\![0, n-1]\!] : \ \mathrm{DFT}(\mathbf{e}_j) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \exp\left(\frac{2\pi ijk}{n}\right) \mathbf{e}_k. \quad (1)$$

and obviously,

$$\mathrm{DFT}(\mathbf{f}) = \sum_{j=0}^{n-1} f(j)\, \mathrm{DFT}(\mathbf{e}_j). \quad (2)$$

Now, let us assume that $n = 2^\nu$ is a power of 2.
DFT can be calculated by fast Fourier transform FFT. This is a typical procedure of time complexity $O(\nu 2^\nu) = O(n \log n)$.
Through the inherent parallelism of quantum computing the procedure can be reduced to time complexity $O(\nu)$.

Let us observe that, on one side, $\mathbb{H}_\nu = \mathbb{C}^n$, and by identifying each $j \in [\![0, 2^\nu - 1]\!]$ with $\varepsilon_j = \varepsilon_{j,\nu-1} \cdots \varepsilon_{j,1}\varepsilon_{j,0}$:

$$\begin{aligned} \mathrm{DFT}(\mathbf{e}_{\varepsilon_j}) &= \bigotimes_{k=0}^{\nu-1} \frac{1}{\sqrt{2}}\left(\mathbf{e}_0 + \exp\left(\frac{\pi i j}{2^k}\right)\mathbf{e}_1\right) \\ &= \tfrac{1}{\sqrt{2}}\left(\mathbf{e}_0+\exp\left(\tfrac{\pi i j}{2^0}\right)\mathbf{e}_1\right)\otimes\tfrac{1}{\sqrt{2}}\left(\mathbf{e}_0+\exp\left(\tfrac{\pi i j}{2^1}\right)\mathbf{e}_1\right)\otimes\cdots\otimes\tfrac{1}{\sqrt{2}}\left(\mathbf{e}_0+\exp\left(\tfrac{\pi i j}{2^{\nu-1}}\right)\mathbf{e}_1\right) \end{aligned} \quad (3)$$

The products appearing in this tensor product suggest the operators $Q_k : \mathbb{H}_1 \to \mathbb{H}_1$ and their "controlled" versions:

$$Q_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\frac{\pi i}{2^k}\right) \end{bmatrix} \;,\; Q^c_{kj} = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\pi i \frac{j}{2^k}\right) \end{bmatrix}.$$

Thus, for instance, if $j = 1$ then $Q^c_{k1} = Q_k$ while if $j = 0$ then $Q^c_{k0} = I$.
For $\mathbf{x}_0 = \frac{1}{\sqrt{2}}(\mathbf{e}_0 + \mathbf{e}_1) = H(\mathbf{e}_0)$, $Q^c_{kj}(\mathbf{x}_0) = \frac{1}{\sqrt{2}}\left(\mathbf{e}_0 + \exp\left(\pi i \frac{j}{2^k}\right)\mathbf{e}_1\right)$.
Each $j \in [\![0, 2^\nu - 1]\!]$ is represented by ε_j. Then, $\forall\, \ell \in [\![0, \nu - 1]\!]$, $\frac{\varepsilon_{j,\ell} 2^\ell}{2^k} = \frac{\varepsilon_{j,\ell}}{2^{k-\ell}}$.

$$\exp\left(\pi i \frac{j}{2^k}\right) = \exp\left(\pi i \frac{\sum_{\ell=0}^{\nu-1} \varepsilon_{j,\ell} 2^\ell}{2^k}\right) = \prod_{\ell=0}^{\nu-1} \exp\left(\pi i \frac{\varepsilon_{j,\ell}}{2^{k-\ell}}\right)$$

and consequently,

$$Q^c_{kj} = Q^c_{k-\nu+1,\varepsilon_{j,\nu-1}} \circ \cdots \circ Q^c_{k-1,\varepsilon_{j,1}} \circ Q^c_{k,\varepsilon_{j,0}}.$$

Since k ranges from 0 to $\nu - 1$ there will be required $2(2\nu - 1)$ gates $Q^c_{\kappa\varepsilon}$, $\kappa \in [\![-(\nu - 1), \nu - 1]\!]$, $\varepsilon \in \{0, 1\}$.
Whenever $j < 2^{\nu_1}$, with $\nu_1 \leq \nu$, all digits with indexes $\nu_1 - 1$ or $\nu - 1$ have value 0, hence the corresponding controlled gates are the identity map.

For each $(j,k) \in [\![0, 2^\nu - 1]\!] \times [\![0, \nu - 1]\!]$,

$$P_{jk} = Q^c_{k-\nu_1+1,\varepsilon_{j,\nu_1-1}} \circ \cdots \circ Q^c_{k-1,\varepsilon_{j,1}} \circ Q^c_{k,\varepsilon_{j,0}}, \qquad (4)$$

where $\nu_1 = \lceil \log_2 j \rceil + 1$. Then: $P_{jk}(\mathbf{x}_0) = \frac{1}{\sqrt{2}}\left(\mathbf{e}_0 + \exp\left(\pi i \frac{j}{2^k}\right)\mathbf{e}_1\right)$.
For a fixed $j \in [\![0, 2^\nu - 1]\!]$, for each $k = 0, \ldots, \nu - 1$, $P_{jk}(\mathbf{x}_0)$ at the right of eq. (3) will appear in an order left to right w.r.t. eq. (3). Then:

$$\begin{array}{rcl}
Q^c_{0,\varepsilon_{j,0}}(\mathbf{x}_0) & = & P_{j0}(\mathbf{x}_0) \\
Q^c_{1,\varepsilon_{j,0}} \circ Q^c_{0,\varepsilon_{j,1}}(\mathbf{x}_0) & = & P_{j1}(\mathbf{x}_0) \\
Q^c_{2,\varepsilon_{j,0}} \circ Q^c_{1,\varepsilon_{j,1}} \circ Q^c_{0,\varepsilon_{j,2}}(\mathbf{x}_0) & = & P_{j2}(\mathbf{x}_0) \\
\vdots & & \vdots \\
Q^c_{\nu-1,\varepsilon_{j,0}} \circ \cdots \circ Q^c_{2,\varepsilon_{j,\nu-3}} \circ Q^c_{1,\varepsilon_{j,\nu-2}} \circ Q^c_{0,\varepsilon_{j,\nu-1}}(\mathbf{x}_0) & = & P_{j,\nu-1}(\mathbf{x}_0)
\end{array}$$

For each $k \in [\![0, \nu - 1]\!]$, the $Q^c_{\ell,\varepsilon_{j,k-\ell}}$, with $\ell = 0, \dots, k$, are applied consecutively and they are selecting the digits in the base-2 representation of j going from the most significant till the least significant.

Henceforth, it is necessary to apply the reverse operator to switch the bits order in each $j \in [\![0, 2^\nu - 1]\!]$.

Each bit ε is represented by the basic vector $\mathbf{e}_\varepsilon$. Consequently, each controlled operator $Q^c_{k,\varepsilon}$, with domain in $\mathbb{H}_1$ can be identified with the operator $\mathbf{x} \mapsto Q^{c2}(\mathbf{x}, \mathbf{e}_\varepsilon)$ where

$$Q^{c2} = (I \otimes Q_k) \circ C \circ (I \otimes Q_k^H) \circ C \circ (Q_k \otimes I). \quad (5)$$

Algorithm for the Fourier transform

Input. $n = 2^\nu$, $\mathbf{f} \in \mathbb{C}^n = \mathbb{H}_\nu$.

Output. $\hat{\mathbf{f}} = \mathrm{DFT}(\mathbf{f}) \in \mathbb{H}_\nu$.

Procedure $\mathtt{DFT}(n, \mathbf{f})$

1. Let $\mathbf{x}_0 := H(\mathbf{e}_0)$.
2. For each $j \in [\![0, 2^\nu - 1]\!]$, or equivalently, for each $(\varepsilon_{j,\nu-1} \cdots \varepsilon_{j,1}\varepsilon_{j,0}) \in \{0,1\}^\nu$, do (in parallel):
   1. For each $k \in [\![0, \nu - 1]\!]$ do (in parallel):
      1. Let $\delta := R_k\left(\varepsilon_j\big|_k\right)$ be the reverse of the chain consisting of the $(k+1)$ less significant bits.
      2. Let $\mathbf{y}_{jk} := \mathbf{x}_0$.
      3. For $\ell = 0$ to k do { $\mathbf{y}_{jk} := Q^{c2}(\mathbf{y}_{jk}, \mathbf{e}_{\delta_{j,\ell}})$ (see eq. (5)) }
   2. Let $\mathbf{y}_j := \mathbf{y}_{j0} \otimes \cdots \otimes \mathbf{y}_{j,\nu-1}$ (see eq. (3)).
3. Output as result $\hat{\mathbf{f}} = \sum_{j=0}^{2^\nu-1} f_j \mathbf{y}_j$.

Agenda

1. Quantum Computation of the Discrete Fourier Transform
2. Shor Algorithm
   - Quantum Algorithm to Calculate the Order of a Number

Modular multiplicative groups

- For $n, m \in \mathbb{Z}$, its greatest common divisor is $d = \gcd(n, m)$ where d divides n and m and any other common divisor divides also d.
- Euclid's Algorithm calculates, for two given n and m, $d = \gcd(n, m)$ and express as $d = an + bm$, with $a, b \in \mathbb{Z}$.
- n and m are relative prime if $\gcd(n, m) = 1$.
- $\Phi(n) = \{m \in [\![1, n]\!] | \gcd(n, m) = 1\}$.
- $\phi(n) = \mathrm{card}(\Phi(n))$: Euler's function at n.
- $(\Phi(n)$, multiplication modulo $n)$ is a group of order $\phi(n)$.
- If $m \in \Phi(n)$ then $m^{\phi(n)} = 1 \bmod n$.
- For each integer $m \in \Phi(n)$ there exists a minimal element r, divisor of $\phi(n)$, such that $m^r = 1 \bmod n$. Such an r is the order of m in $\Phi(n)$.

Let n be an integer to be factored

1. Select an integer m such that $1 < m < n$.
2. If $\gcd(n, m) = d > 1$, then d is a non-trivial factor of n.
3. Otherwise, $m \in \Phi(n)$.
    1. If m has an even order r, then $k = m^{\frac{r}{2}}$ will be such that $k^2 = 1 \bmod n$, and $(k-1)(k+1) = 0 \bmod n$.
    2. By calculating $\gcd(n, k-1)$ and $\gcd(n, k+1)$, one gets non-trivial factors of n.

First problem

Find an element of even order in $\Phi(n)$

If m is chosen randomly, the probability that m has even order is $1 - \frac{1}{2^\ell}$ where ℓ is the number of prime factors in n.
Hence, the probability that after k attempts the sought witnessing number has not been found is $2^{-k\ell}$ and this tends to zero quickly as k increases.

Biggest problem

Calculate the order of a current element m in $\Phi(n)$

Let $\nu = \lceil \log_2 n \rceil$, ν is the size of n.
$O(n) = O(2^\nu)$, thus an exhaustive procedure has exponential complexity with respect to the input size. Shor's algorithm is based over a polynomial-time procedure in ν to calculate the order of an element.

Calculating the Order of a Number

Let $n \in \mathbb{N}$ and $\nu = \lceil \log_2 n \rceil$ be its size.
Let κ s.t. $n^2 \leq 2^\kappa < 2n^2$, i.e. $\kappa = \lceil 2 \log_2 n \rceil$.
There will be necessary to use $\kappa + \nu$ qubits and all calculations will lie in

$$\mathbb{H}_{\kappa+\nu} = \mathbb{H}_\kappa \otimes \mathbb{H}_\nu, \text{ of dimension } 2^{\kappa+\nu} = 2^\kappa \cdot 2^\nu.$$

$\forall m \in \Phi(n)$, let $V_m : \mathbb{H}_{\kappa+\nu} \to \mathbb{H}_{\kappa+\nu}$,

$$V_m : \mathbf{e}_{\varepsilon_j} \otimes \mathbf{e}_{\varepsilon_i} \mapsto \mathbf{e}_{\varepsilon_j} \otimes \mathbf{e}_{\varepsilon_{f(i,j,m)}} \tag{6}$$

where $f(i,j,m) = (j + m^i) \bmod n$. f is r-periodic w.r.t. its first argument i.

Elements whose Order is a Power of 2

Suppose $m \in \Phi(n)$ whose order r is a power of 2.
Let $P_1 = H^{\otimes\kappa} \otimes I^{\otimes\nu}$, $H, I : \mathbb{H}_1 \to \mathbb{H}_1$ Hadamard's operator and identity.

$$P_1(\mathbf{e_0} \otimes \mathbf{e_0}) = \frac{1}{\sqrt{2^\kappa}} \sum_{\varepsilon \in \{0,1\}^\kappa} \mathbf{e}_\varepsilon \otimes \mathbf{e_0}.$$

Let's write $\mathbf{s}_1 = P_1(\mathbf{e_0} \otimes \mathbf{e_0})$. By applying V_m,

$$V_m(\mathbf{s}_1) = \frac{1}{\sqrt{2^\kappa}} \sum_{i=0}^{2^\kappa - 1} \mathbf{e}_{\varepsilon_i} \otimes \mathbf{e}_{\varepsilon_{f(i,0,m)}}.$$

Let $\mathbf{s}_2 = V_m(\mathbf{s}_1)$. Let $J_j = \{i | 0 \leq i \leq 2^\kappa - 1 : i = j \bmod r\}$.
$[\![0, 2^\kappa - 1]\!] = \bigcup_{j=0}^{r-1} J_j$, and each set J_j has cardinality $s = \frac{2^\kappa}{r} \in \mathbb{Z}$. Thus

$$\mathbf{s}_2 = \frac{1}{\sqrt{2^\kappa}} \sum_{j=0}^{r-1} \left(\sum_{i \in J_j} \mathbf{e}_{\varepsilon_i} \right) \otimes \mathbf{e}_{\varepsilon_{mj}}. \quad (7)$$

By a Measurement, it is chosen a vector $\mathbf{e}_{\varepsilon_i} \otimes \mathbf{e}_{\varepsilon_{mj_0}}$, $i \in J_{j_0}$, for a fixed $j_0 \leq r$, with probability $\frac{r}{2^\kappa}$. The corresponding state is

$$\mathbf{s}_3 = \sum_{i=0}^{2^\kappa - 1} g(i) \mathbf{e}_{\varepsilon_i} \otimes \mathbf{e}_{\varepsilon_{mj_0}}. \tag{8}$$

where $g : i \mapsto \begin{cases} \sqrt{\frac{r}{2^\kappa}} & \text{if } i \in J_{j_0} \\ 0 & \text{if } i \notin J_{j_0} \end{cases}$ is also r-periodic. $\hat{g}$ is periodic, with period proportional to $\frac{1}{r}$. On other side:

$$\check{\mathbf{s}}_3 = \mathrm{DFT}^H(\mathbf{s}_3) = \sqrt{\frac{r}{2^\kappa}} \sum_{k=0}^{s-1} \left(\frac{1}{\sqrt{2^\kappa}} \sum_{\ell=0}^{2^\kappa - 1} \exp\left(-\frac{2\pi i \ell}{2^\kappa} (kr + j_0) \right) \mathbf{e}_\ell \right) \otimes \mathbf{e}_{\varepsilon_{mj_0}},$$

and, by interchanging the summation order we get:

$$\mathbf{s}_4 = \check{\mathbf{s}}_3 = \frac{1}{\sqrt{r}} \left(\sum_{\ell=0}^{2^\kappa - 1} \left(\frac{1}{s} \sum_{k=0}^{s-1} \exp\left(-\frac{2\pi i \ell k}{s} \right) \right) \exp\left(-\frac{2\pi i \ell j_0}{2^\kappa} \right) \mathbf{e}_\ell \right) \otimes \mathbf{e}_{\varepsilon_{mj_0}}. \tag{9}$$

Since $\exp\left(-\frac{2\pi i\ell}{s}\right)$ is a s-th root of unit, $\frac{1}{s}\sum_{k=0}^{s-1}\exp\left(-\frac{2\pi i\ell k}{s}\right)$ is either 1 or 0 depending on whether ℓ has the form $\ell = ts$, with $t = 0,\ldots,r-1$.

$$\mathbf{s}_4 = \frac{1}{\sqrt{r}}\left(\sum_{t=0}^{r-1}\exp\left(-\frac{2\pi i t j_0}{r}\right)\mathbf{e}_{\frac{2^\kappa t}{r}}\right)\otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}}. \quad (10)$$

By a measurement it is obtained $\frac{2^\kappa t_0}{r}$, with $t_0 \in [\![0, r-1]\!]$, each with probability r^{-1}.

If $t_0 = 0$, then it is not possible to obtain any information about r and the procedure should be repeated.

Otherwise, it is obtained the rational value $\frac{r_0}{r_1} = \frac{t_0}{r}$. The values r_0 and r_1 are known, but till this point neither t_0 nor r are known. Nevertheless, a fortiori r_1 should divide r. Thus, the quantum algorithm should be applied once more with $m_1 = m^{r_1}$ as input. In a recursive way, the factorization $r = r_1 r_2 \cdots r_p$ is got, containing at most $\log_2 r$ factors.

Algorithm to find a divisor of the order of an element

Input. $n \in \mathbb{N}$, $m \in \Phi(n)$ of order a power of 2.

Output. r such that $r|o(m)$.

Procedure `DivisorOrderPower2`(n, m)

1. Let $\nu := \lceil \log_2 n \rceil$, $\kappa := 2\nu$.
2. Let $V_m : \mathbb{H}_{\kappa+\nu} \to \mathbb{H}_{\kappa+\nu}$ be defined as in eq. (6).
3. Let $\mathbf{s}_1 := (H^{\otimes\kappa} \otimes I^{\otimes\nu})(\mathbf{e_0} \otimes \mathbf{e_0})$.
4. Let $\mathbf{s}_2 := V_m(\mathbf{s}_1)$.
5. Let $\mathbf{s}_3 := \sum_{i=0}^{2^\kappa-1} g(i)\mathbf{e}_{\varepsilon_i} \otimes \mathbf{e}_{\varepsilon_{m^{j_0}}}$ be the equivalent state to "take a measurement" in $\mathbf{s}_2$. g is determined by eq. (8).
6. Let $\mathbf{s}_4 := \mathtt{IDFT}(2^\kappa, \mathbf{s}_3)$.
7. Let $\mathbf{e}_{\varepsilon_k} \otimes \mathbf{e}_{\varepsilon_{m^{j_0}}}$ be a measurement of $\mathbf{s}_4$.
8. If $k == 0$ then repeat from step 3. Else, let $\frac{r_0}{r_1} = \frac{k}{2^\kappa}$ and output as result r_1.

Algorithm to calculate the order of an element

Input. $n \in \mathbb{N}$, $m \in \Phi(n)$ of order a power of 2.

Output. r such that $r = o(m)$.

Procedure $\texttt{OrderPower2}(n, m)$

1. Initially $r := 1$ and $m_1 := m$.
2. Repeat
   1. let $r_1 := \texttt{DivisorOrderPower2}(n, m_1)$;
   2. update $r := r \cdot r_1$;
   3. update $m_1 := m_1^{r_1} \bmod n$.

   until $r_1 == 1$.
3. Output r.

Elements with Arbitrary Order

Let us drop the assumption that order r is a power of 2.

As before, let V_m be defined as in eq. (6): $\mathbf{s}_1 = (H^{\otimes\kappa} \otimes I^{\otimes\nu})(\mathbf{e_0} \otimes \mathbf{e_0})$ and

$$\mathbf{s}_2 = V_m(\mathbf{s}_1) = \frac{1}{\sqrt{2^\kappa}} \sum_{j=0}^{r-1} \left(\sum_{i \in J_j} \mathbf{e}_{\varepsilon_i} \right) \otimes \mathbf{e}_{\varepsilon_{mj}}. \qquad (11)$$

where the sets J_j are equivalence classes, but in the current case their cardinalities may differ. If $u = 2^\kappa \bmod r$ and $s = (2^\kappa - u)/r$ then u classes will have $s + 1$ elements and the remaining classes will have s elements. Lets$_j = s + 1$ for $j = 1, \dots, u$ and $s_j = s$ for $j = u + 1, \dots, r - 1, 0$. Then the state after taking a measurement, as in eq. (8), is, for some $j_0 \in [\![0, r-1]\!]$:

$$\mathbf{s}_3 = \sum_{i=0}^{2^\kappa - 1} g(i)\mathbf{e}_{\varepsilon_i} \otimes \mathbf{e}_{\varepsilon_{mj_0}}. \qquad (12)$$

where $g : i \mapsto \begin{cases} \frac{1}{\sqrt{s_{j_0}}} & \text{if } i \in J_{j_0} \\ 0 & \text{if } i \notin J_{j_0} \end{cases}$

$$\mathbf{s}_4 = \check{\mathbf{s}}_3 = \frac{1}{\sqrt{2^\kappa}} \left(\sum_{\ell=0}^{2^\kappa-1} \left(\frac{1}{\sqrt{s_{j_0}}} \sum_{k=0}^{s_{j_0}-1} e^{\left(-\frac{2\pi i \ell k r}{2^\kappa}\right)} \right) e^{\left(-\frac{2\pi i \ell j_0}{2^\kappa}\right)} \mathbf{e}_\ell \right) \otimes \mathbf{e}_{\varepsilon_{m j_0}} . \qquad (13)$$

The coefficients involving the inner summation never will be zero (since r does not divide 2^κ, there is no "complete sample" of s_{j_0}-th roots of unit).

In a measurement for the first qubit, the probability to choose $\mathbf{e}_\ell \otimes \mathbf{e}_{\varepsilon_{m j_0}}$ is

$$P(\ell) = \frac{1}{\sqrt{2^\kappa s_{j_0}}} \left| \sum_{k=0}^{s_{j_0}-1} \exp\left(-\frac{2\pi i \ell k r}{2^\kappa} \right) \right|^2$$

and the maxima of those values correspond to $\ell = \mathtt{ClosestInteger}\left(\frac{k 2^\kappa}{r}\right)$.

Suppose that after a measurement, it is chosen $\mathbf{e}_{\ell_k} \otimes \mathbf{e}_{\varepsilon_{m j_0}}$, with $\ell_k = \mathtt{ClosestInteger}\left(\frac{k 2^\kappa}{r}\right)$. Then, when divided by 2^κ we get $\frac{\ell_k}{2^\kappa} \sim \frac{k}{r}$, and from here we should know r.

Continued fractions

If $\frac{p}{q} \in \mathbb{Q}^+$, its continued fraction is

$$\frac{p}{q} = a_0 + \frac{1}{a_1 + \frac{1}{\cdots + \frac{1}{a_v}}} = [a_0, a_1, \ldots, a_v] \tag{14}$$

where $a_0, a_1, \ldots, a_v \in \mathbb{N} - \{0\}$.
For each $w \leq v$, $[a_0, a_1, \ldots, a_w]$ is the w-th convergent of $\frac{p}{q}$, and is a rational approximation of $\frac{p}{q}$.

Continued Fractions Algorithm

Input. $\frac{p}{q} \in \mathbb{Q}$.

Output. $[a_0, a_1, \ldots, a_v]$: continued fraction representing $\frac{p}{q} \in \mathbb{Q}$.

Procedure `ContinuedFraction`$(\frac{p}{q})$

1. Initially $lst := []$ (the empty list) and $xcurr := \frac{p}{q}$.
2. While the denominator of $xcurr$ is greater than 1 do
   1. Let $i :=$ `IntegerPart`$(xcurr)$;
   2. let express $\frac{p_1}{q_1} = xcurr$;
   3. update $xcurr := \frac{q_1}{p_1 - iq_1}$;
   4. update $lst := lst * [i]$.
3. Update $lst := lst * [xcurr]$.
4. Output lst.

Algorithm to find divisors of the order of an element

Input. $n \in \mathbb{N}$, $m \in \Phi(n)$.

Output. r such that $r|o(m)$.

Procedure `DivisorOrder`(n, m)

1. Let $\nu := \lceil \log_2 n \rceil$, $\kappa = \lceil 2 \log_2 n \rceil$.
2. Let $V_m : \mathbb{H}_{\kappa+\nu} \to \mathbb{H}_{\kappa+\nu}$ as in eq. (6).
3. Let $\mathbf{s}_1 := (H^{\otimes \kappa} \otimes I^{\otimes \nu})(\mathbf{e_0} \otimes \mathbf{e_0})$.
4. Let $\mathbf{s}_2 := V_m(\mathbf{s}_1)$.
5. Let $\mathbf{s}_3 := \sum_{i=0}^{2^\kappa - 1} g(i) \mathbf{e}_{\varepsilon_i} \otimes \mathbf{e}_{\varepsilon_{m^{j_0}}}$ be the state equivalent to "take a measurement" in $\mathbf{s}_2$. g is as in eq. (12).
6. Let $\mathbf{s}_4 := \mathtt{IDFT}(2^\kappa, \mathbf{s}_3)$.
7. Let $\mathbf{e}_{\varepsilon_{\ell_k}} \otimes \mathbf{e}_{\varepsilon_{m^{j_0}}}$ a measurement of $\mathbf{s}_4$.

8. If $\ell_k == 0$ then repeat from step 3. Else
   1. Let $[a_0, a_1, \dots, a_v] := \texttt{ContinuedFraction}\left(\frac{\ell_k}{2^\kappa}\right)$;
   2. Let $[c_0, c_1, \dots, c_v]$ be the convergents list; and
   3. output the list of denominators less than n of those convergents.

From the obtained divisors of orders, it is possible to find the orders themselves in a similar manner as was sketched in the procedure `OrderPower2`, but in this case it is necessary to track all divisors provided by the above procedure `DivisorOrder`.