量子計算 - 計算モデルの概要と EXACT, THRESHOLD アルゴリズムの紹介

@cympfh

ここではあくまでも計算モデルとして量子計算を眺め, なにが実現出来るかを調べる. 物理学的な視点にはあまり興味がない.

序章

まず量子計算を支える qbit (量子ビット) がどのような性質を持つかを説明する. 次にどのようなゲート (回路) が実現可能で qbit を操作できるかを紹介する. ただしいずれも物理的原理までは立ち入らず紹介するだけに留める.

qbit (量子ビット, Qbit, qubit)

古典コンピュータにおける計算には bit を用いる. これは 0 または 1 のいずれかの状態を取るものである. 対して qbit はこの 2 つの状態を確率的に持つ. 具体的には 2 つの状態の線型結合として記述される.

qbit が 0 である状態を $|0\rangle$, qbit が 1 である状態を $|1\rangle$ と書く (bra-ket 記法という) ことにし, 一般の状態はこの 2 つの重ねあわせ (線型結合):

$$\alpha|0\rangle + \beta|1\rangle$$

で表される. ここで α, β は複素数を取り $(\alpha, \beta \in \mathbb{C})$, また物理学の要請から

$$|\alpha|^2 + |\beta|^2 = 1$$

という制約を要請される.

> **復習**: 複素数 $z \in \mathbb{C}$ は, 実数 $x, y \in \mathbb{R}$ によって $z = x + yi$ で一意に表現される値で, これについて共役数 $\bar{z} = x - yi (\in \mathbb{C})$ と $|z|^2 = \bar{z} \cdot z = x^2 + y^2 (\in \mathbb{R})$ を定めるのだった.
>
> **補足**: 係数の制約を無視すれば, qbit の取り得る空間というのは 2 つの基底 $|0\rangle, |1\rangle$ からなる二次元の複素数上のベクトル空間である. 制約があるので実際にはこれの部分空間であっても, 部分ベクトル空間ではないが.

観測

qbit は状態の重ね合わせだと言ったが実は実際に観測をすると, $|0\rangle$ または $|1\rangle$ のどちらかに見える.

先程, 係数には制約 $|\alpha|^2 + |\beta|^2 = 1$ があると述べたが, 実はこれらはどちらに観測されるかの確率になっている.

すなわち, ある qbit, $\alpha|0\rangle + \beta|1\rangle$ を実際に観測すると, 確率 $|\alpha|^2$ で $|0\rangle$ を得, 確率 $|\beta|^2$ で $|1\rangle$ を得る. (確率の和はちょうど 1 になっており不都合はない.)

そして観測という行為は qbit に干渉する. 一度状態が確定すると, 以降何度観測をしても初めに得た結果を得るだけである. 即ち, 一度 $|0\rangle$ を観測したならば, その qbit は $|0\rangle = 1 \cdot |0\rangle + 0 \cdot |1\rangle$ に **収束**したと言える.

n qbit

bit を n 個並べたものを n bit と言うように, qbit を n 個並べたものを n qbit と呼ぶことにする.

- n qbit は
  - 自由に一列に並べられる
  - 自由に一部を取り出せる
  - 自由に一部だけを観測できる

特に **並べる** という操作を二項演算子 $\otimes$ で表すことにする. n qbit x と m qbit y を並べることで $n+m$ qbit $x \otimes y$ を得る. ここで並べる場合には順序があるので $x \otimes y \neq y \otimes x$ であることに註意.

簡単に 2 qbit について考える. $|0\rangle$ の右に $|1\rangle$ を並べて得る 2 qbit を

$$|01\rangle := |0\rangle \otimes |1\rangle$$

と書くことにする. すると 2 qbit は

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle$$

の 4 通りの状態を取り得る. 実際にはそれぞれの qbit は重ね合わせであるから, 2 qbit はこの 4 通りの重ね合わせになる:

$$\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

2 qbit のそれぞれが $\beta_0|0\rangle + \beta_1|1\rangle$ と $\gamma_0|0\rangle + \gamma_1|1\rangle$ だったとするとき, 形式的に

$$(\beta_0|0\rangle + \beta_1|1\rangle) \otimes (\gamma_0|0\rangle + \gamma_1|1\rangle) = \beta_0\gamma_0|00\rangle + \beta_0\gamma_1|01\rangle + \beta_1\gamma_0|10\rangle + \beta_1\gamma_1|11\rangle$$

という掛け算をすればよい. 係数はただの掛け算で $|\cdot\rangle$ は横に結合させるだけ. 実際, $|00\rangle$ を観測する確率は, 同時確率なので $|\beta_0|^2|\gamma_0|^2 = |\beta_0\gamma_0|^2$ となっていて, $\alpha_{00} = \beta_0\gamma_0$ とすれば都合がよい. 同様に $\alpha_{ij} = \beta_i\gamma_j$ とすればよく, $|ij\rangle$ を観測する確率は $|\alpha_{ij}|^2$ だと言える. $\sum_{i,j} |\alpha_{ij}|^2 = 1$ は各 qbit の係数の制約から従う.

部分的観測

n qbit の内 1 qbit だけを観測した結果, その qbit の状態は先述したとおり, 観測された状態に確定して固定されるが, 残りの $n-1$ qbit についてはなおも重ね合わせの状態を保ったままで観測が干渉することはない.

例として, 2 qbit

$$\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

を考える. これの 1 qbit 目を観測した結果 $|0\rangle$ を得たとする. 1 qbit 目は 0 で固定されるので, 観測しうる状態は $|00\rangle$ または $|01\rangle$ だけであるので, 観測後の 2 qbit は,

$$\alpha'_{00}|00\rangle + \alpha'_{01}|01\rangle$$

で表される.

α'_{00}, α'_{01} はどうなるかと言うと, これらは結局, 2 qbit 目が $|0\rangle$, $|1\rangle$ で観測される確率 (の閉包根) であって, (それは観測の前後で変化しない)

元の 2 qbit が $\beta_0|0\rangle + \beta_1|1\rangle$ と $\gamma_0|0\rangle + \gamma_1|1\rangle$ だったとすると, 観測後の事後確率なので

- $\alpha'_{00} = \gamma_0$
- $\alpha'_{01} = \gamma_1$

と言える. また先程見たように $\alpha_{00} = \beta_0\gamma_0$ なので, $\alpha_{00} = \beta_0\alpha'_{00}$. 同様に $\alpha_{01} = \beta_0\alpha'_{01}$.

従って β_0 の逆数を単に定数 κ と書くことにすると, 事後の 2 qbit は

$$\kappa\alpha_{00}|00\rangle + \kappa\alpha_{01}|01\rangle$$

と書ける.

さて係数の自乗和が 1 である性質から実は κ は決まる. 即ち,

$$|\kappa|^2(|\alpha_{00}|^2 + |\alpha_{01}|^2) = 1$$

があるので κ の大きさは決まる.

量子ゲート

qbit に対する実現可能な操作で次のようなゲートを作成することが理論上可能.

量子 NOT

次のような操作 X が存在する:

- $X|0\rangle = |1\rangle$
- $X|1\rangle = |0\rangle$

この操作 X は線形写像のように働く. 即ち,

$$X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle$$

となる.

制御 (controlled) NOT
次のような 2 qbit に対する操作 X が存在する:

- $X|i,j\rangle = |i, i \oplus j\rangle$

ここで $\oplus$ は排他的論理和で, $0 \oplus j = j$, $1 \oplus j = 1 - j$.

アダマール (Hadamard) ゲート
次のような H が存在する:

- $H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$
- $H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$

H は二回通すことで恒等写像になる.

$H|0\rangle$ のことを $|+\rangle$, $H|1\rangle$ のことを $|-\rangle$ と書くことにする. この符号はもちろん 2 つの状態が和になってるか差になってるかを意味している.

> **補足**: これも重ね合わせの状態については線形写像のように働く. ところで, 重ね合わせられてない状態というのは, 実際に観測すれば容易に手に入る. それをアダマールゲートに通すと, 2 つの状態が同確率で観測されるような状態の qbit が手に入る. また H を組み合わせることで, 全ての 2^n 状態が等確率で観測できる n qbit を作ることができる.

量子並列性

n qbit の基底の状態 $|ij\ldots k\rangle$ を普通の古典 n bit $i, j, \ldots, k$ と同一視する. n bit を入力にして 1 bit を出力する古典回路 f について, 同程度の効率で計算できる次のような量子ゲート U_f が存在する:

$$U_f(x \otimes |i\rangle) = x \otimes |i \oplus f(x)\rangle$$

ここで x は n qbit. i は 0 または 1 (もちろん) で, $\oplus$ は排他的論理和.

さて, アダマールゲートを用いれば 2 つの状態を全く同等に含んだ量子を作れるのだった. それを U_f に通すことで, **実質的に** $f(0)$ と $f(1)$ を並列に計算するようなことができる. 具体的には次を実行する.

1. $H|0\rangle$ に $|0\rangle$ を並べる
   - $(H|0\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$
2. これを U_f に通す
   - $\frac{1}{\sqrt{2}}|0, f(0)\rangle + \frac{1}{\sqrt{2}}|1, f(1)\rangle$

一度の U_f の計算で $f(0)$ と $f(1)$ が行われているのが分かる. この性質を **量子並列性** という. ただし, これをこのまま観測するだけでは結局そのどちらか $|x, f(x)\rangle$ しか得られない. 並列性のメリットを享受するには工夫が必要である. その古典的な一例であるドイチュのアルゴリズムを次に見る.

ドイチュのアルゴリズム (Deutsch algorithm)

1 bit から 1 bit を出力する古典回路 f について, $f(0) \oplus f(1)$ を一度の U_f (f 相当の計算) で計算することができる.

アルゴリズムは次の通り:

1. $|+\rangle = H|0\rangle$ と $|-\rangle = H|1\rangle$ を得る
   - これを並べたものを $|+-\rangle = |+\rangle \otimes |-\rangle$ とする
2. $|\phi_1, \phi_2\rangle = U_f|+-\rangle$
3. $H|\phi_1, \phi_2\rangle = H|\phi_1\rangle \otimes H|\phi_2\rangle$ を計算して 1 qbit 目を観測する

具体的に計算を追う.

1. $|+-\rangle = |+\rangle \otimes |-\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)$
2. $U_f|+-\rangle = \frac{1}{2}(|0, f(0)\rangle - |0, 1-f(0)\rangle + |1, f(1)\rangle - |1, 1-f(1)\rangle)$
3. $H(U_f|+-\rangle) = \frac{1}{2}\left[(|+\rangle \otimes H|f(0)\rangle) - (|+\rangle \otimes H|1-f(0)\rangle) + (|-\rangle \otimes H|f(1)\rangle) + (|-\rangle \otimes H|1-f(1)\rangle)\right]$

最期の式を更に詳細に計算する.

初めの 2 項 $(|+\rangle \otimes H|f(0)\rangle) - (|+\rangle \otimes H|1-f(0)\rangle)$ を調べる. $f(0), 1-f(0)$ はちょうど一方が 0 なら他方は 1 である.

$f(0) = 0$ のとき,

$$\begin{aligned}(|+\rangle \otimes H|f(0)\rangle) - (|+\rangle \otimes H|1-f(0)\rangle) &= |+\rangle \otimes (|+\rangle - |-\rangle) \\ &= |+\rangle \otimes (\sqrt{2}|1\rangle) \\ &= \sqrt{2}(|+\rangle \otimes |1\rangle)\end{aligned}$$

同様に $f(0) = 1$ のとき,

$$(|+\rangle \otimes H|f(0)\rangle) - (|+\rangle \otimes H|1-f(0)\rangle) = -\sqrt{2}(|+\rangle \otimes |1\rangle)$$

である. この 2 つの場合をまとめて

$$(|+\rangle \otimes H|f(0)\rangle) - (|+\rangle \otimes H|1-f(0)\rangle) = (-1)^{f(0)}\sqrt{2}|+1\rangle$$

と書ける. ここで $|+\rangle \otimes |1\rangle$ を $|+1\rangle$ と書いた.

また残りの 2 項についても同様に

$$(|-\rangle \otimes H|f(1)\rangle) - (|-\rangle \otimes H|1-f(1)\rangle) = (-1)^{f(1)}\sqrt{2}|-1\rangle$$

となる.

というわけで

$$\begin{aligned} H(U_f|+-\rangle) &= \frac{1}{2}\left[(-1)^{f(0)}\sqrt{2}|+1\rangle + (-1)^{f(1)}\sqrt{2}|-1\rangle\right] \\ &= \frac{1}{\sqrt{2}}\left[(-1)^{f(0)}|+1\rangle + (-1)^{f(1)}|-1\rangle\right] \end{aligned}$$

を得る.

2 qbit 目は常に 1 であることがわかる. さて 1 qbit 目にだけ注目すると

$$\frac{1}{\sqrt{2}}\left[(-1)^{f(0)}|+\rangle + (-1)^{f(1)}|-\rangle\right]$$

である. $f(0), f(1)$ によって 4 通りに場合分けをすると,

1. case $f(0)=0, f(1)=0$
   - $\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)) = |0\rangle$
2. case $f(0)=0, f(1)=1$
   - $\frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)) = |1\rangle$
3. case $f(0)=1, f(1)=0$
   - $\frac{1}{\sqrt{2}}(-|+\rangle + |-\rangle)) = -|1\rangle$
4. case $f(0)=1, f(1)=1$
   - $\frac{1}{\sqrt{2}}(-|+\rangle - |-\rangle)) = -|0\rangle$

観測する場合にはその係数の大きさの自乗の確率で状態を得る. 係数はそれぞれ $+1$ または -1 になっているから結局必ず $|0\rangle$ または $|1\rangle$ を得ることになり, それは $f(0)\oplus f(1)$ と一致している. 例えば $f(0)=1, f(1)=0$ の場合は $-|1\rangle$ を得, 観測した結果 $(-1)^2$ の確率で $|1\rangle$ を得る.

qbit の数ベクトル表示

n qbit が取り得る状態は 2^n つの基底

$$|0\ldots000\rangle,\ |0\ldots001\rangle,\ |0\ldots010\rangle,\ \ldots,\ |1\ldots111\rangle$$

の線型結合

$$\sum \alpha_{i_1,i_2,\ldots,i_n}|i_1 i_2 \cdots i_n\rangle$$

で表される. ここでこの qbit の状態を

$$\begin{bmatrix} \alpha_{0\ldots000} \\ \alpha_{0\ldots001} \\ \alpha_{0\ldots010} \\ \cdots \\ \alpha_{1\ldots111} \end{bmatrix} \in \mathbb{C}^{2^n}$$

で表示する.

ユニタリ変換

複素正方行列 U で

$$U^\dagger U = UU^\dagger = I$$

を満たす行列をユニタリ行列という．ここで I は単位行列, $U^\dagger$ は成分の共役を取って転置した行列のこと $(U^\dagger = (U^*)^T)$.

性質

2 つのユニタリ行列 U, V の積 UV もユニタリ行列.

定理

任意のユニタリ変換を再現する量子ゲートが存在する.

ここでユニタリ変換とは, qbit を数ベクトル表示したときに右からユニタリ行列を掛ける操作のことを言う.

$$\begin{bmatrix} \beta_{0\ldots000} \\ \beta_{0\ldots001} \\ \beta_{0\ldots010} \\ \cdots \\ \beta_{1\ldots111} \end{bmatrix} = U \begin{bmatrix} \alpha_{0\ldots000} \\ \alpha_{0\ldots001} \\ \alpha_{0\ldots010} \\ \cdots \\ \alpha_{1\ldots111} \end{bmatrix}$$

は qbit α を量子ゲート U に通した結果 qbit β を得る操作を表す. 逆に U がユニタリ行列であるならば, 量子ゲートで必ず再現できる.

このような行列 U の大きさは $2^n \times 2^n$ と, 2 のべき乗でないといけないが, 実際には気にする必要はない. 一般の m $(2^{n-1} < m \leq 2^n)$ について, ユニタリー行列 $U \in \mathbb{C}^{m\times m}$ に $2^n - m$ 個, サイズを広げて,

$$U' = \begin{bmatrix} & U & & \\ & & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

広げた部分は対角に 1 を置いてできる行列 U' も問題なくユニタリー行列.

n qbit であるが実際には 2^n 状態の内 m 状態しか取り得ないようなもの長さ m の列ベクトルで表現でき, $m \times m$ 行列 U で, 操作を記述すればよい. 実際には余白を 0 で埋めた長さ 2^n の列ベクトルを $2^n \times 2^n$ 行列 U' で操作したものだとすれば問題ない.

例: 量子 NOT

量子 NOT というゲート X は $\alpha|0\rangle + \beta|1\rangle$ を $\beta|0\rangle + \alpha|1\rangle$ に写す.

即ち

$$\begin{bmatrix} \beta \\ \alpha \end{bmatrix} = X \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

となれば良いが, これは

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

であってユニタリ行列になっている.

EXACT and THRESHOLD algorithm

紹介する論文

- “Exact quantum query complexity of EXACT and THRESHOLD”
- Andris Ambainis, Jānis Iraids, Juris Smotrovs
- arxiv.org/abs/1302.1235

与えられた n bit (qbit ではない) について立ってる (1 である) ビットの数を数えるアルゴリズム. 正確に述べると, 立ってるビットの数がちょうど k であるか判定する EXACT_k^n と, k 以上であるか判定する $\mathrm{THRESHOLD}_k^n$ の 2 つのアルゴリズムを与える.

notation

n qbit の基底の内, i 番目 (0-indexed) の qbit だけが立ってるもの $|0\cdots010\cdots0\rangle$ を

$$|i\rangle$$

と書くことにする $(i = 0, 1, \ldots, n-1)$.

2 つの n qbit $|i\rangle$ と $|j\rangle$ とを並べたものを

$$|i, j\rangle := |i\rangle \otimes |j\rangle$$

と書く.

更にそれが $2n$ qbit であることが紛らわしく無ければ,

$$|i\rangle := |i, i\rangle$$

と書く.

> **補足**: 実際には, 中身の表現はどうでもよくて, 要するに区別できる状態であればよい. つまり $|\cdot\rangle$ の中に書く数字は単なるラベルだとしか思ってない.

1 qbit $|x\rangle$ $(x = 0, 1)$ に対して $(-1)^x$ を $\hat{x}$ と書く $(\hat{x} = 1, -1)$.

Query (量子クエリ)

これから EXACT と THRESHOLD という 2 つのアルゴリズムを説明するが, 共に Q という操作が登場する. これは入力 $x_1, x_2, \ldots, x_n$ に依存する写像であって,

- $Q|i\rangle = \hat{x}_i|i\rangle$

で定めるものである. このように入力に依存する操作をクエリと呼ぶ.

1 回のクエリの処理 (Q の適用) のたびに, 入力 x_i を一回読む必要がある. アルゴリズムの複雑性として, クエリを処理する回数を指標とする. これを量子クエリ計算量という.

EXACT

n qbit

$$x = (x_0 x_1 \cdots x_{n-1})$$

の内, ちょうど k 個が立ってるか判定するアルゴリズムを EXACT_k^n とする.

$$\mathrm{EXACT}_k^n : \{0,1\}^n \to \{\mathrm{true}, \mathrm{false}\}$$

今から述べる彼らのアルゴリズムでは $2n$ qbit を用意する. 取り得る状態は $|i\rangle$ と $|i,j\rangle$ $(0 \le i,j < n; i < j)$ の $n(n+1)/2$ 個だけとし, 初め $|0\rangle$ であるとする. 従って, 長さ $n(n+1)/2$ の複素ベクトルで状態は表現される.

さていきなり一般の EXACT_k^n を考えるのは難しいので, まずは EXACT_k^{2k} の場合を考える.

EXACT_k^{2k}

全体が偶数ビットで, 内のちょうど半分のビットが立ってるかを判定する. このことは, ビットそれぞれを $x \mapsto \hat{x}$ としたときのその和 $\sum_i \hat{x}_i$ がゼロになることと等しいことを利用する.

$$\mathrm{EXACT}_k^{2k}(x) = \mathrm{true} \iff \sum_i \hat{x_i} = 0$$

次の 3 つの操作を用いる:

1. $U_1|0\rangle = \frac{1}{\sqrt{2k}} \sum_{i=0}^{2k-1} |i\rangle$
2. クエリ Q
3. $U_2|i\rangle = \frac{1}{\sqrt{2k}} \left(\sum_{i<j} |i,j\rangle - \sum_{i>j} |j,i\rangle + |0\rangle \right)$

ここで未定義な値 (e.g. $U_1|1\rangle$) はどう定義してもいいので U_1, U_2 をユニタリー行列になるようにする.

$|0\rangle$ にこれらを順に通す:

$$
\begin{array}{c}
|0\rangle \\
\Big\downarrow U_1 \\
\frac{1}{\sqrt{2k}} \sum_{i=0}^{2k-1} |i\rangle \\
\Big\downarrow Q \\
\frac{1}{\sqrt{2k}} \sum_{i=0}^{2k-1} \hat{x}_i |i\rangle \\
\Big\downarrow U_2 \\
\frac{1}{2k} \Big((\hat{x}_i - \hat{x}_j)|i,j\rangle + \sum_{i=0}^{2k-1} \hat{x}_i |0\rangle \Big)
\end{array}
$$

で, 最後の量子を観測したときに得られうる状態は

1. $|i,j\rangle$ $(i<j)$
2. $|0\rangle$

の 2 種類がある.

もし $|0\rangle$ を観測したならば, $\sum_i \hat{x}_i$ がゼロでないことがわかる. なぜなら, $|0\rangle$ を観測する確率は $(\frac{1}{2k}\sum \hat{x_i})^2)$ であるから. 従って,

$$\mathrm{EXACT}_k^{2k}(x) = \mathrm{false}$$

であることがわかる.

次に $|i,j\rangle$ を観測したときを考えると, この係数について $\hat{x}_i - \hat{x}_j \neq 0$ であることがわかる. 即ち, ビット x_i とビット x_j とが異なることを示してる. 今, EXACT_k^{2k} はビットが立っているものの数と立っていないものの数が **等しい**かどうかにだけ興味があるので, 次が言える.

$$\mathrm{EXACT}_k^{2k}(\{x_0, x_1, \ldots, x_{n-1}\}) = \mathrm{EXACT}_{k-1}^{2k-2}(\{x_0, x_1, \ldots, x_{n-1}\} \setminus \{x_i, x_j\}).$$

(ビットの列を集合に書き換えているので註意.)

このことはビットに関する帰納法を示唆している. その基底状態として,

$$\mathrm{EXACT}_0^0\ \{\} = \mathrm{true}$$

がある.

帰納部分は, (false であれば) 運が良ければさっさと終わるが, 最悪 (true なら必ずそうで) $\frac{2k}{2}$ 回繰り返す必要がある.

EXACT_k^n

入力 $x = (x_0 \cdots x_{n-1})$ に余計にビットを付け足せば EXACT_k^{2k} に出来る. 具体的に, $\mathrm{EXACT}_k^n(x)$ は次に等しい:

- case $n = 2k$
  - $\mathrm{EXACT}_k^{2k}(x)$
- case $n > 2k$
  - $\mathrm{EXACT}_{n-k}^{2n-2k}(x \mathbin{+\!\!+} (1 \dots 1))$
  - $n - 2k$ 個の 1 bit 列を連結
- case $n < 2k$
  - $\mathrm{EXACT}_k^{2k}(x \mathbin{+\!\!+} (0 \dots 0))$
  - $2k - n$ 個の 0 bit 列を連結

クエリ計算量

EXACT_k^{2k} のクエリ計算量は, 再帰の回数なので, 最悪 k. したがって, EXACT_k^n のクエリ計算量は, 最悪

$$\max\{k, n-k\}$$

となる.

THRESHOLD

n bit

$$x = (x_0 x_1 \cdots x_{n-1})$$

の内, k 個 **以上**が立ってるか判定するアルゴリズムを

$$\mathrm{THRESHOLD}_k^n \colon \{0,1\}^n \to \{\mathrm{true}, \mathrm{false}\}$$

とする.

$\mathrm{THRESHOLD}_{k+1}^{2k+1}$

まず初めに $\mathrm{THRESHOLD}_{k+1}^{2k+1}$ を考える. これは即ち過半数ビットが立ってるかを判定する手続きである.

入力は $x_0, x_1, \dots, x_{2k}$ の $2k+1$ ビット. これに関して

- $S_0 = \{i \mid 0 \le i < 2k+1, x_i = 0\}$
- $S_1 = \{i \mid 0 \le i < 2k+1, x_i = 1\}$

とする. それぞれのサイズ (要素数) を $\#S_0, \#S_1$ と書くことにして, 全体は奇数ビットなので, 必ず $\#S_0 \neq \#S_1$. 今 $\#S_0 > \#S_1$ とする. 逆の場合も同様であるので省略する.

次のことが言える:

- when $i \in S_0$,
  - when $\#S_0 = \#S_1 + 1$
    - $\sum_{j \neq i} \hat{x_j} = 0$

  - when $\#S_0 > \#S_1 + 1$
    - $\mathrm{THRESHOLD}_{k+1}^{2k+1}(x) = \mathrm{THRESHOLD}_{k-1}^{2k-1}(x \setminus \{x_i, x_j\})$ $(\forall j \neq i)$
- when $i \in S_1$
  - $\mathrm{THRESHOLD}_{k+1}^{2k+1}(x) = \mathrm{THRESHOLD}_{k-1}^{2k-1}(x \setminus \{x_i, x_j\})$ $(\forall j \neq i)$

THRESHOLD では次の 3 つの操作を用いる:

1. U_1
   - EXACT のときと同様
2. Q
   - EXACT のときと同様
3. U_3
   - $U_3|i\rangle = \frac{\sqrt{2k+1}}{2k}\left(\sum_{i<j}|i,j\rangle - \sum_{i>j}|j,i\rangle + \frac{1}{2k}|j\rangle\right)$
   - ユニタリ変換になるように

EXACT と同様に $|0\rangle$ から初めてこれらに順に通す.

$$
\begin{array}{c}
|0\rangle \\
\Big\downarrow U_1 \\
\frac{1}{\sqrt{2k}}\sum_{i=0}^{2k-1}|i\rangle \\
\Big\downarrow Q \\
\frac{1}{\sqrt{2k}}\sum_{i=0}^{2k-1}\hat{x}_i|i\rangle \\
\Big\downarrow U_3 \\
\frac{\sqrt{2k-1}}{2k\sqrt{2k+1}}\sum_{i<j}(\hat{x}_i - \hat{x}_j)|i,j\rangle + \frac{1}{2k\sqrt{2k+1}}\sum_{i=0}^{2k}\sum_{i\neq j}\hat{x}_i|j\rangle
\end{array}
$$

これを測定すると

1. $|i,j\rangle$ または
2. $|j\rangle$

のいずれかを得る.

1. $|i,j\rangle$ を得た時,

$\hat{x}_i - \hat{x}_j \neq 0$ であるから, $x_i \neq x_j$. 従って

$$\mathrm{THRESHOLD}_{k+1}^{2k+1}(x) = \mathrm{THRESHOLD}_{k+1}^{2k+1}(x \setminus \{x_i, x_j\})$$

2. $|j\rangle$ を得た時,

$\sum_{i\neq j}\hat{x}_i \neq 0$. 先ほどの性質を思い出せば,

$$\mathrm{THRESHOLD}_{k+1}^{2k+1}(x) = \mathrm{THRESHOLD}_{k+1}^{2k+1}(x \setminus \{x_i, x_j\})$$

以上から, ちょうど k 回, 再帰的に U_1, Q, U_2 を適用することで

$$\mathrm{THRESHOLD}_0^1(x_0) = x_0$$

というわけで $\mathrm{THRESHOLD}_{k+1}^{2k+1}$ はクエリ計算量 $k+1$ で解ける.

$\mathsf{THRESHOLD}_k^n$
EXACT と同じ感じでやる.

クエリ計算量は,

$$\max\{k+1, n-k+1\}.$$