SuperCon 2018 本選問題

Aug. 20, 2018

本選問題

量子コンピュータをシミュレートしてください

背景

量子コンピュータは

- 素因数分解 ($p \times q \longmapsto (p, q)$) が効率的に解ける。今使われている暗号はほとんどすべて解読できる。
- n 個の箱に 1 つ当たりがあるときに、$\sqrt{n}$ 回箱を開けて、当たりを見つけることができる。

IBM が 50 量子ビット (2017 年 11 月)、Google が 72 量子ビット (2018 年 5 月) の量子コンピュータを作ったと発表。

ただし、現在の量子コンピュータはあまり長い時間計算できない。また、エラーレートはそこそこ高い。

本選問題

30 量子ビットの量子コンピュータをシミュレートしてください

※ Google は 50 量子ビット深さ 40 の量子回路は現在最高のスパコンでもシミュレートできないと主張している。

量子状態

量子状態は測定するまで中身が分からないブラックボックス

量子ビットは測定すると確率的に 0 か 1 が得られる

量子ビット

ビット `0`, `1`

量子ビット

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

より一般に、

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \qquad (|\alpha|^2 + |\beta|^2 = 1).$$

この $|\psi\rangle$ を測定すると測定結果 `0` が得られる確率が $|\alpha|^2$
測定結果 `1` が得られる確率が $|\beta|^2$

2 量子ビット

$$|00\rangle := \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \quad |01\rangle := \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \quad |10\rangle := \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \quad |11\rangle := \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

より一般に、

$$|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle = \begin{bmatrix}\alpha\\\beta\\\gamma\\\delta\end{bmatrix}$$

ただし $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$ を満たす。
測定結果が `00` 〜11 の確率が $|\alpha|^2$ 〜$|\delta|^2$ である。

n 量子ビット

n 量子ビット

$$|0\cdots0\rangle := \begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}, \quad |0\cdots01\rangle := \begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}, \quad \cdots \quad |1\cdots1\rangle := \begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix}.$$

より一般に、

$$|\psi\rangle = \sum_{\mathsf{x}\in\{0,1\}^{\mathsf{n}}} \alpha_{\mathsf{x}} |\mathsf{x}\rangle = \begin{bmatrix}\alpha_{0\ldots0}\\\vdots\\\alpha_{1\ldots1}\end{bmatrix}$$

ただし $\sum_{\mathsf{x}\in\{0,1\}^{\mathsf{n}}} |\alpha_{\mathsf{x}}|^2 = 1$ を満たす。
測定結果が x となる確率が $|\alpha_{\mathsf{x}}|^2$ である。

量子ゲート: いくつかの量子ビットを変化させる
(ブラックボックスから別のブラックボックスへ変化)

量子ゲートは「線形変換」
量子ゲート U が $x \in \{0,1\}^n$ について $|x\rangle$ を $U(|x\rangle)$ に変化させる
とすると、一般の重ね合わせ状態

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

は

$$U(|\psi\rangle) = \sum_{x \in \{0,1\}^n} \alpha_x U(|x\rangle)$$

に変化させる。よって$\underline{U(|x\rangle)\text{ だけ定義すればよい}}$。

パウリ X 変換

ただの 0,1 反転

$$|0\rangle \longmapsto |1\rangle$$
$$|1\rangle \longmapsto |0\rangle$$

回路図

$$|\mathsf{x}\rangle \;\; \text{——}\boxed{\mathsf{X}}\text{——} \;\; |\bar{\mathsf{x}}\rangle$$

ここで、$\bar{\mathsf{x}} := 1 - \mathsf{x}$ である。

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \longmapsto \alpha\,|1\rangle + \beta\,|0\rangle$$

パウリ Z 変換

直感的には理解できない

$$
\begin{aligned}
|0\rangle &\longmapsto |0\rangle \\
|1\rangle &\longmapsto -|1\rangle
\end{aligned}
$$

回路図

$$|\mathsf{x}\rangle \;\; — \boxed{\mathsf{Z}} — \;\; (-1)^{\mathsf{x}} |\mathsf{x}\rangle$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \longmapsto \alpha|0\rangle - \beta|1\rangle$$

アダマール変換

重ね合わせ状態を作りだす

$$|0\rangle \longmapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|1\rangle \longmapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

回路図

$$|\mathsf{x}\rangle \;\text{———}\boxed{\mathsf{H}}\text{———}\; \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{\mathsf{x}}|1\rangle)$$

$$\begin{aligned}|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \longmapsto \quad & \alpha\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \beta\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ = \quad & \frac{1}{\sqrt{2}}(\alpha + \beta)|0\rangle + \frac{1}{\sqrt{2}}(\alpha - \beta)|1\rangle\end{aligned}$$

n 量子ビットに 1 量子ビットゲートを適用

$i \in \{0, 1, \ldots n-1\}$ について

パウリ X 変換を i 番目の量子ビットに適用すると

$$|x_{n-1}x_{n-2}\cdots x_1\rangle \longmapsto |x_{n-1}x_{n-2}\cdots x_{i+1}\,\overline{x_i}\,x_{i-1}\cdots x_1\rangle$$

パウリ Z 変換を i 番目の量子ビットに適用すると

$$|x_{n-1}x_{n-2}\cdots x_1\rangle \longmapsto (-1)^{x_i}\,|x_{n-1}x_{n-2}\cdots x_1\rangle$$

アダマール変換を i 番目の量子ビットに適用すると

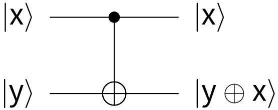
$$\begin{aligned}|x_{n-1}x_{n-2}\cdots x_1\rangle \longmapsto \frac{1}{\sqrt{2}}\big(&|x_{n-1}x_{n-2}\cdots x_{i+1}\,0\,x_{i-1}\cdots x_1\rangle \\ &+ (-1)^{x_i}\,|x_{n-1}x_{n-2}\cdots x_{i+1}\,1\,x_{i-1}\cdots x_1\rangle\big)\end{aligned}$$

コントロール X 変換

$|x\rangle$ を「コントロール量子ビット」、$|y\rangle$ を「ターゲット量子ビット」とするコントロール X 変換は $|yx\rangle$ を次のように変化させる。

$$
\begin{aligned}
|00\rangle &\longmapsto |00\rangle \\
|01\rangle &\longmapsto |11\rangle \\
|10\rangle &\longmapsto |10\rangle \\
|11\rangle &\longmapsto |01\rangle
\end{aligned}
$$

回路図による表現は次のようになる。



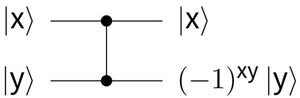
ここで $\oplus$ は排他的論理和 ($x \oplus y = 0 \iff x = y$ で定義される $\{0, 1\}$ 上の二項演算) である (C 言語でいう ^)。

コントロール Z 変換

コントロール Z 変換は $|yx\rangle$ を次のように変化させる。

$$
\begin{aligned}
|00\rangle &\longmapsto |00\rangle \\
|01\rangle &\longmapsto |01\rangle \\
|10\rangle &\longmapsto |10\rangle \\
|11\rangle &\longmapsto -|11\rangle
\end{aligned}
$$

回路図による表現は次のようになる。



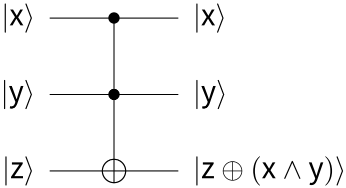
x と y について対称で、どちらがコントロール量子ビットでどちらがターゲット量子ビットという区別はない。

トフォリ変換

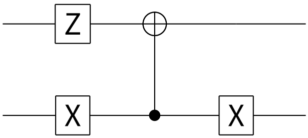
$|x\rangle$ と $|y\rangle$ を「コントロール量子ビット」、$|z\rangle$ を「ターゲット量子ビット」とするトフォリ変換は $|zyx\rangle$ を次のように変化させる。

$$
\begin{aligned}
|000\rangle &\longmapsto |000\rangle\,, & \qquad |001\rangle &\longmapsto |001\rangle \\
|010\rangle &\longmapsto |010\rangle\,, & \qquad |011\rangle &\longmapsto |111\rangle \\
|100\rangle &\longmapsto |100\rangle\,, & \qquad |101\rangle &\longmapsto |101\rangle \\
|110\rangle &\longmapsto |110\rangle\,, & \qquad |111\rangle &\longmapsto |011\rangle
\end{aligned}
$$

回路図による表現は次のようになる。



量子回路の例



状態によらず、
<u>異なる量子ビットへの操作は順番を入れ替えてもよい</u>。

- 0 番目の量子ビットに Z を適用する
- 1 番目の量子ビットに X を適用する

どちらを先にしてもよいので、上の回路図は曖昧さがない。

本選問題: 概要

概要

問題の入力として量子回路を与える。$|00\cdots0\rangle$ をその量子回路への入力とした場合の量子回路の出力を測定した時に、得られる確率が一番高い測定結果とその確率を問題の解とする。

入力

$$
\begin{array}{ll}
\mathrm{N} & \mathrm{K} \\
\mathrm{G}_1 & \mathrm{O}_1 \\
\mathrm{G}_2 & \mathrm{O}_2 \\
& \vdots \\
\mathrm{G}_\mathrm{K} & \mathrm{O}_\mathrm{K}
\end{array}
$$

N は量子ビットの数 (30 以下)
K は量子ゲートの数 (2000 以下)
G_i は量子ゲートの種類
O_i は量子ゲートが適用される量子ビットのインデックス

入力

$$
\begin{array}{ll}
N & K \\
G_1 & O_1 \\
G_2 & O_2 \\
 & \vdots \\
G_K & O_K
\end{array}
$$

$$25 \leq N \leq 30$$

$$500 \leq K \leq 2000$$

$$G_i \in \{\, X, Z, H, CX, CZ, CCX \,\}, \qquad \forall i \in \{1, 2, \ldots, K\},$$

ただし H の総数は 124 以下

$$O_i = \begin{cases} a, & \text{if} \quad G_i \in \{\, X, Z, H \,\} \\ a\ b, & \text{if} \quad G_i \in \{\, CX, CZ \,\} \\ a\ b\ c, & \text{if} \quad G_i = CCX \end{cases}, \qquad \begin{array}{l} 0 \leq a, b, c \leq N-1, \\ a, b, c \text{ は相異なる} \\ \forall i \in \{1, 2, \ldots, K\} \end{array}$$

本選問題: 出力

出力

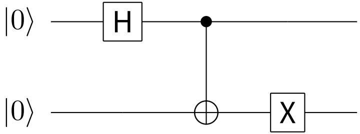
測定確率最大の測定結果の一つの十進数表現
その測定確率

測定確率最大の測定結果が複数ある場合はどれを出力してもよい。
測定確率は相対誤差 10^{-5} 未満を正解とする。

入出力例

```
2 3
H 0
CX 0 1
X 1
```

回路図は次のようになる。



$$|00\rangle \longmapsto \frac{1}{\sqrt{2}}(|00\rangle+|01\rangle) \longmapsto \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle) \longmapsto \frac{1}{\sqrt{2}}(|10\rangle+|01\rangle)$$

よって出力は

```
1
0.5
```

もしくは

```
2
0.5
```

評価方法

プログラムのコンパイルは

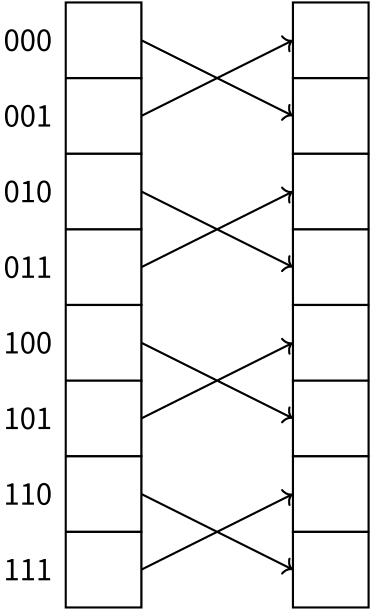
```
nvcc program.cu -O2 -arch=sm_60
```

で行う。速度を評価するための入力は配布の gen.cppを用いて生成する。$25 \leq \mathsf{N} \leq 30$, $500 \leq \mathsf{K} \leq 2000$ の範囲で入力を生成し、計算結果と実行時間を評価する。速度差が判別できるくらい十分多くの入力を用いて評価する。順位付けは

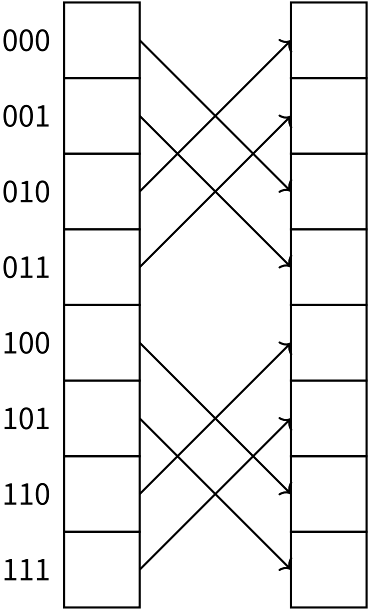
- 計算結果が正しいものが多い方が上位
- 計算結果が正しいものが同数の場合は実行時間の総和が小さい方が上位

というルールに従って決める。上位は全問正解するものと期待している。結局のところ $\mathsf{N} = 30$ の場合が一番時間がかかるので、そこで一番差がつくと考えている。

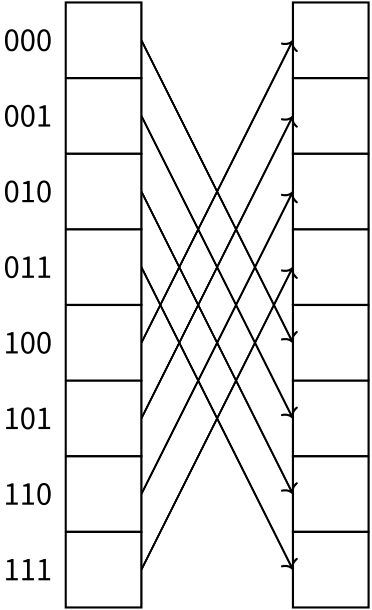
ベクトルの変化 パウリ X 変換を 0 番目に



ベクトルの変化 パウリ X 変換を 1 番目に



ベクトルの変化　パウリ X 変換を 2 番目に



CPU プログラム

$$|x_{n-1}x_{n-2}\cdots x_1\rangle \longmapsto |x_{n-1}x_{n-2}\cdots x_1\rangle$$

```
void X(float *p, float *q, int N, int a) {
  int t = 1 << a;
  int i;
  for(i = 0; i < N; i++){
    q[i] = p[i^t];
  }
}
```

CPU プログラム cpu.c を参考にせよ。
ただし、cpu.c は数値誤差については考慮していない。

資料には書いていないルールと注意点

- GPU ライブラリの使用は禁止 (Thrust など)。
- `float` のベクトルの最大値を GPU で高速に計算する関数はこちらから与えるので使ってよい。ほとんど改良の余地はないと考えているが改良してもよい。