Wyklad13

4 maja 2011
19:25

Kwantowa Transformata Fouriera i jej konsekwencje

1. Wstęp

Algorytm Shora (1993)

Rozkład liczby n bitowej na czynniki pierwsze w czasie n^3

Klasyczne problem bardzo trudny ale jednocześnie b. ważny (RSA).

Jeśli N liczba N bitowa $N < 2^n$

najlepsze metody:

- sprawdzamy czy dzieli się przez wszystkie liczby mniejsze od N (w zasadzie $\sqrt{N}$)

  mamy $\sim 2^{\frac{n}{2}}$ operacji

Czas obliczeń rośnie wykładniczo.

- Najlepsze algorytmy klasyczne mają złożoność:

  $\sim 2^{\sqrt[3]{n}}$ sygnały mi dowody wielomian

Algorytm dość skomplikowany, ale podstawowy @Ukmat to kwantowa transformata Fouriera

2. Transformata Fouriera klasycznie

- Klasyczna dyskretna trans Fouriera

$x_0, \ldots, x_{N-1}$

$\downarrow F$

$y_0 \ldots y_{N-1}$

$$y_k = \frac{1}{\sqrt{N}} \sum_j e^{\frac{2\pi i j k}{N}} x_j$$

Liczba operacji N^2 ilości 2^{2n}

Liczba operacji N^2, inaczej 2^{2n}
(jeśli N liczba n bitowa)
FFT ma trochę przyspieszenie $N \log N = 2^n n$
wciąż wykładniczo z liczbą bitów mierzących dane

3. Kwantowa trans. Fouriera
definiujemy op. unitarny U_F na stanach n qubitowych

$$|j\rangle \xrightarrow{U_F} \frac{1}{\sqrt{N}} \sum_k e^{\frac{2\pi i jk}{N}} |k\rangle \qquad N=2^n$$

Można sprawdzić że jest unitarna.

Jeśli zapiszemy nasze dane jako superpozycję:

$$\sum_j x_j |j\rangle \xrightarrow{U_F} \sum_j \frac{1}{\sqrt{N}} \sum_k x_j e^{\frac{2\pi i jk}{N}} |k\rangle =$$

$$= \sum_k y_k |k\rangle$$

gdzie $y_k = \frac{1}{\sqrt{N}} \sum_j x_j e^{\frac{2\pi i jk}{N}}$ to liczy nam na raz całą trans. Fouriera.

Uwagi:

— nie oznacza to że można odczytać wszystkie y_k na raz, czyli ten kw. paralelizm trzeba jakoś sprytnie użyć dalej (okazuje się to możliwe właśnie w alg. Shora)

— U_F też trzeba zbudować z el. bramek kwantowych i pytanie jak to się skaluje z n. Okazuje się że b. dobrze bo $\sim n^2$

[illegible]

Czyli mamy zysk w porównaniu z klasycznym FFT jak

n^2 do $n 2^n$?!

Wykładniczy zysk.

4. Co ma trans. Fouriera do rozkładu na czynniki pierwsze?

Transf. Fouriera pozwala znajdować okres funkcji!

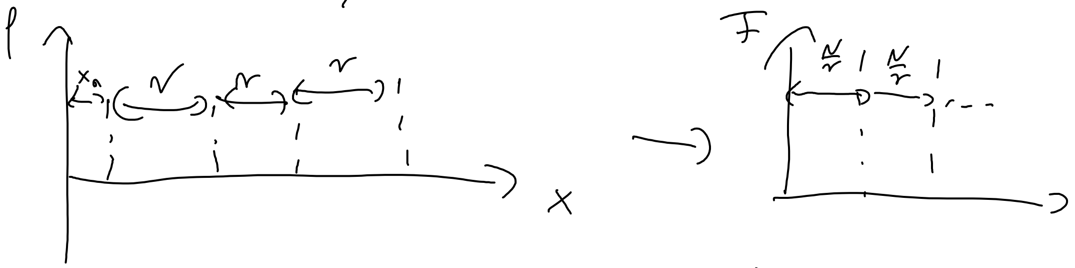
Rozważmy funkcję $f: Z_N \to Z$

(↑ całkowite mod N)

Szukamy jej okresu $f(x+r) = f(x)$

Liczymy dla wszystkich argumentów

$$|f\rangle = \frac{1}{\sqrt{N}} \cdot \sum_{x=0}^{N-1} |x\rangle |f(x)\rangle$$



Mierzymy drugi rejestr jeśli wynikiem y to

$\to$ $N = k \cdot r$, $\underbrace{\frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} |x_0 + k \cdot r\rangle}_{|\psi_{x_0,r}\rangle} |y\rangle$, $f(x_0) = y_0$

Mając stan $|\psi_{x_0,r}\rangle$ chcemy stwierdzić jakie jest r? Mierzenie x nic nie da dostaniemy $x_0 + kr$ i nic nie wiemy, ale

możemy zrobić trans. Fouriera

$$F|\psi_{x_0,r}\rangle = \frac{1}{\sqrt{N}} \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} \sum_{c=0}^{N} e^{\frac{2\pi i\, l (x_0 + kr)}{N}} |l\rangle =$$

$$= \frac{1}{\sqrt{N}} \sum_{c=0}^{N} e^{\frac{2\pi i\, l x_0}{N}} |l\rangle \cdot \underbrace{\frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} e^{\frac{2\pi i\, l \cdot r \cdot k}{N}}}_{\sqrt{K} \text{ jeśli } l \cdot r = s \cdot N,\ l = s \cdot \frac{N}{r}} =$$

$$= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{\frac{2\pi i\, s\, x_0}{r}} \left| s \cdot \frac{N}{r} \right\rangle$$

Teraz mierzymy i dostajemy

$\lambda = s \cdot \frac{N}{r}$ dla jakiegoś $s = 0, \ldots, r-1$

Czy stąd możemy wyznaczyć r?

Jeśli $GCD(s, r) = 1$ to tak bo wiemy $\frac{\lambda}{N} = \frac{s}{r}$ więc sprowadzamy $\frac{\lambda}{N}$ do nieskracalnej postaci mamy r.

Jeśli nie to niestety nie. Ale to jest mało prawdopodobne, czyli możemy wyznaczyć r

Fakt z teorii liczb:

Liczba liczb pierwszych mniejszych od r równa jest $\frac{r}{\log r}$, czyli prawdopodobieństwo że wylosowane s jest $GCD(s, r) = 1$ jest co najmniej $\frac{1}{\log r}$.

Jeśli powtórzymy procedurę $\log N$ razy to mamy szanse $\sim 1 - \left(1 - \frac{1}{\log r}\right)^{\log N} \simeq 1 - \left(1 - \frac{1}{\log r}\right)^{\log r \frac{\log N}{\log r}} =$

$$= 1 - e^{-\frac{\log N}{\log r}} \longrightarrow 1$$

5 Co nam wyznaczenie okresu pomaga?

do rozkładu na czynniki pierwsze ?

N – liczba, której czynników pierwszych szukamy

Losujemy liczbę $a < N$ i sprawdzamy

$GCD(a, N)$ ⇧ Jeśli $GCD > 1$ to już mamy czynnik!

Jeśli $GCD(a, N) = 1$ (zupełnie niedużej) to:

Wtedy z Tw Eulera ⇧ istnieje r:

$$a^r = 1 \mod N \qquad (r - \text{rząd } a)$$

Jeśli r jest parzyste:

$$a^r - 1 = 0 \mod N$$

$$\underbrace{\left(a^{\frac{r}{2}} - 1\right)}_{\alpha}\underbrace{\left(a^{\frac{r}{2}} + 1\right)}_{\beta} = k \cdot N \qquad \alpha \cdot \beta = k \cdot N$$

czyli α lub β muszą mieć wspólne dzielniki z N

(chyba że α, β – wielokrotność N)

{ Fakt z teorii liczb: Dla $a < N$, t.że $GCD(a, N) = 1$
wybranego losowo prawd. że rząd r ($a^r = 1 \mod N$)
jest parzysty i $a^{\frac{r}{2}} \pm 1$ nie są wielokrotnością $N \geq \frac{1}{2}$

Liczymy $GCD(\alpha, N)$, $GCD(\beta, N)$ dostajemy czynnik pierwszy N

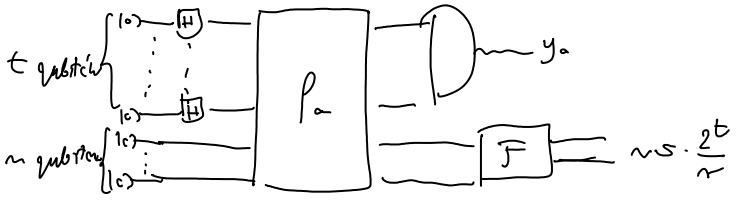
Zauważmy, że najtrudniejsze jest wyznaczenie r.

Niech

$$f(x) = a^x \mod N$$

Czym jest rząd r? Okresem funkcji f ?

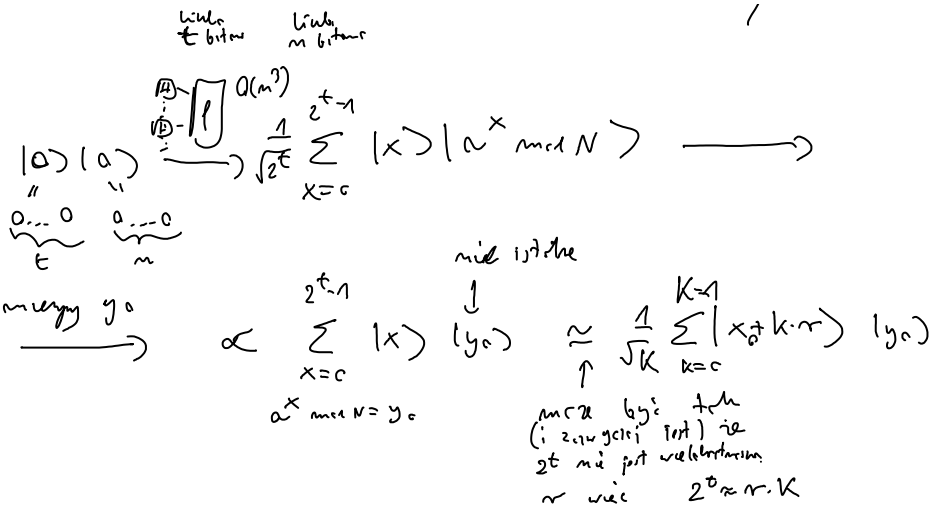
$$f(x + r) = a^x \cdot a^r \mod N = a^x \mod N$$



$N < 2^n$

$$U_a : |x\rangle|0\rangle = |x\rangle|a^x \mod N\rangle$$

{ wymaga n^3 bramek



przybliżmy t bitowe ułamki $\frac{s}{r}$, jeśli t

odpowiednio duże (chcemy się że $t \approx 2m$) to

wystarczy aby wyznaczyć r. (continued fraction algorithm)

Uwaga: Widzimy jest tej implementacji funkcji f_a,

poprzez elem. bramki kwantowe. Na szczęście to

też jest $O(m^3)$

Czyli cała procedura jest $O(m^3)$ ☆ c