Rozwiązanie zadania F 941.
Po włączeniu zielonego światła samochody nie ruszają jednocześnie. Najpierw rusza pierwszy rząd samochodów, które stoją bezpośrednio przy sygnalizatorze, potem kolejno ruszają następne rzędy – wzdłuż korka rozchodzi się rodzaj „fali". Niech w czasie T_z, gdy włączone jest zielone światło, obok sygnalizatora przejeżdża część korka o długości L. Na czas T_z składa się czas potrzebny na to, aby „fala" przebyła drogę L i czas potrzebny, aby samochód drogę L przejechał. Jeżeli v jest prędkością samochodu (bez uwzględniania jego rozpędzania się), a u prędkością rozchodzenia się „fali", to

$$T_z = \frac{L}{v} + \frac{L}{u} = L\left(\frac{1}{v} + \frac{1}{u}\right).$$

Jeżeli czas, na który jest włączone czerwone światło wynosi T_c, to średnia prędkość poruszania się w korku wynosi $V_S = L/(T_z + T_c)$. Podstawiając, znajdujemy

$$V_S = \frac{T_z}{T_z + T_c} \cdot \left(\frac{1}{v} + \frac{1}{u}\right)^{-1},$$

a stąd dla $T_z = T_c$ otrzymujemy $u = 2vV_S/(v - 2V_S) = 6\,\mathrm{m/s}$. Po podwojeniu czasu T_z prędkość samochodu w korku wyniesie:

$$V_{2S} = \frac{2T_z}{2T_z + T_c} \cdot \left(\frac{1}{v} + \frac{1}{u}\right)^{-1} = \\ = \frac{2(T_z + T_c)}{2T_z + T_c} V_S = \frac{4}{3} V_S = 2\frac{\mathrm{m}}{\mathrm{s}}.$$

Rozwiązanie zadania F 942.
Skoro prędkość kolumny wynosi u, a odległość przednich zderzaków kolejnych ciężarówek wynosi l to $t_1 = l/(u - v_1)$ i $t_2 = l/(v_2 - u)$, gdzie $(u - v_1)$ i $(v_2 - u)$ są względnymi prędkościami samochodu osobowego i kolumny w przypadku (1) i (2). Stąd $(u - v_1)t_1 = l$ oraz $(v_2 - u)t_2 = l$. Rozwiązując ten układ równań dostajemy:

$$u = \frac{v_1 t_1 + v_2 t_2}{t_1 + t_2},$$

$$l = \frac{t_1 t_2 (v_2 - v_1)}{t_1 + t_2}.$$

Kolejne ciężarówki będą mijały stojący samochód co

$$t = \frac{l}{u} = \frac{t_1 t_2 (v_2 - v_1)}{v_1 t_1 + v_2 t_2} = 5\,\mathrm{s}.$$

Odpowiedź na zadanie ze strony 19:

$$p_1 = -3, \quad p_2 = 5,$$
$$q_1 = -4, \quad q_2 = 7.$$

Algorytm faktoryzacji Shora

Wojciech CZERWIŃSKI

W 1994 roku Peter Shor, pracujący wówczas w Bell Labs w New Jersey, pokazał, jak przy użyciu hipotetycznego komputera kwantowego rozłożyć w czasie wielomianowym dowolną liczbę naturalną na czynniki pierwsze. W tamtym czasie algorytmy kwantowe dopiero raczkowały. To właśnie odkrycie Shora spowodowało wielki rozwój tej dziedziny. Społeczność informatyków zrozumiała, że gdyby udało się zbudować komputer kwantowy rozsądnej wielkości, to świat stałby się istotnie inny. Nie jest bowiem znany żaden algorytm dla problemu faktoryzacji, czyli rozkładu na dzielniki pierwsze, który działa w czasie wielomianowym na komputerze klasycznym. Co więcej, nawet nie znaleziono algorytmu losowego, który z dużym prawdopodobieństwem w zazwyczaj niedługim czasie faktoryzuje liczbę: nie jest po prostu znana zupełnie żadna rozsądna heurystyka... W 1994, ale też i teraz, w 2017 roku, po prostu nie umiemy rozkładać szybko liczb na czynniki pierwsze. A na trudności faktoryzacji opiera się m.in. kryptologia, najbardziej znany kryptosystem RSA dałby się łatwo łamać, gdybyśmy umieli szybko rozkładać liczby na czynniki pierwsze. Drugim najbardziej popularnym problemem, na którego trudności opiera się wiele w kryptologii, jest problem logarytmu dyskretnego (dla danych $a, b \in \mathbb{N}$ i liczby pierwszej p znajdź k takie, że $a^k \equiv b \bmod p$). Warto wiedzieć, że w tym samym artykule Shor udowodnił również, że komputer kwantowy umie rozwiązywać problem logarytmu dyskretnego w czasie wielomianowym. Algorytm Shora nie tylko zainspirował intensywne badania w tej dziedzinie, ale chyba do tej pory jest najbardziej znanym i celebrowanym algorytmem kwantowym.

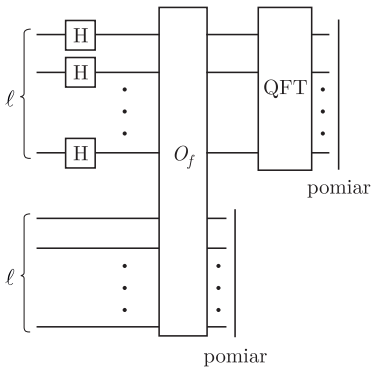
W tym artykule postaramy się zrozumieć najistotniejsze idee w algorytmie Shora. Niektóre szczegóły techniczne pominiemy, gdyż tłumaczenie wszystkiego zajęłoby raczej kilkanaście stron niż kilka.

Sprecyzujmy problem. Dostajemy liczbę $n \in \mathbb{N}$. W naszych rozważaniach skupimy się na przypadku, gdy n jest iloczynem dwóch liczb pierwszych, tj. $n = p_1 p_2$. Ten przypadek jest również trudny (nie ma dla niego żadnych szybko działających heurystyk), a algorytm Shora w ogólności działa prawie identycznie, jak dla tego przypadku. Nasz cel to znaleźć liczby p_1 i p_2. Po pierwsze powiedzmy sobie od razu, że algorytm Shora jest oparty na losowości. Uruchomiony wiele razy z pewnym dużym (bliskim 1) prawdopodobieństwem znajdzie rozkład $n = p_1 p_2$. Myślmy, że zarówno p_1, jak i p_2 mają po 100 cyfr, wtedy będziemy mieli odpowiednie wyobrażenie o tym, co się da, a czego nie da się szybko zrobić.

Pierwszy krok, niemający jeszcze żadnego związku z kwantami, to redukcja faktoryzacji do problemu *rzędu elementu* modulo n. Problem ten, dla danych $x, n \in \mathbb{N}$, pyta o najmniejsze naturalne $r > 0$ takie, że $x^r \equiv 1 \bmod n$ (takie r nazywamy *rzędem* x modulo n). Przy założeniu, że umiemy w czasie wielomianowym znajdować rząd elementu (wszędzie tu używana jest losowość, więc przestaniemy się na niej skupiać, a czasem nawet o niej wspominać), pokażemy, jak rozłożyć n na czynniki w czasie wielomianowym. Rozważmy $n = p_1 p_2$ i wylosujmy liczbę x ze zbioru $\{1, \ldots, n-1\}$. Jeśli $\mathrm{nwd}(x, n) \neq 1$ (co możemy szybko sprawdzić algorytmem Euklidesa), to świetnie, bo wtedy $\mathrm{nwd}(x, n) = p_1$ albo $\mathrm{nwd}(x, n) = p_2$ i znaleźliśmy rozkład. Ale to się zdarza rzadko. Załóżmy więc, że $\mathrm{nwd}(x, n) = 1$. Można wykazać (nie bardzo trudno, szczegóły pominiemy), że dla co najmniej jednej czwartej wylosowanych x zachodzą następujące dwa warunki: 1) r, czyli rząd x, jest parzysty, 2) $x^{\frac{r}{2}} \not\equiv \pm 1 \bmod n$. Dla takiego x mamy $n \mid (x^r - 1) = (x^{\frac{r}{2}} - 1)(x^{\frac{r}{2}} + 1)$. Jednak skoro n nie dzieli żadnego z dwóch nawiasów, to musi być tak, że p_1 dzieli jeden z nich, a p_2 drugi. Zatem $\mathrm{nwd}(n, x^{\frac{r}{2}} - 1)$ jest równe albo p_1, albo p_2. Łatwo je obliczyć algorytmem Euklidesa, a tym samym znaleźć rozkład n. A więc wystarczy skupić się na znajdowaniu rzędu liczby x modulo n, co zrobimy przy użyciu algorytmu kwantowego.

Ustalmy pewne q, które należy do przedziału $(n^2, 2n^2]$ oraz jest potęgą dwójki, niech $q = 2^\ell$. Nasz algorytm na wejściu będzie miał 2ℓ drutów, czyli będzie operował na 2ℓ kubitach (albo jeszcze inaczej: stan pamięci może być opisany

przez wektor długości 1 z $\mathbb{C}^{2^{2\ell}}$). Te 2ℓ drutów podzielimy na dwa segmenty po ℓ drutów. Zaczynamy od stanu pamięci równego 0 na wszystkich kubitach. Czyli, formalnie rzecz biorąc, jest to stan $|0\dots0\rangle$, gdzie ciąg zer ma długość 2ℓ. My jednak na potrzeby naszego algorytmu będziemy o nim myśleli jako o $|0^\ell\rangle \otimes |0^\ell\rangle$, co będziemy zapisywać w skrócie jako $|0^\ell\rangle|0^\ell\rangle$.



Cały obwód kwantowy realizujący algorytm Shora przedstawiony jest na rysunku. Wchodzi do niego 2ℓ drutów po lewej, do których po kolei aplikowane są bramki kwantowe i na końcu wykonywany jest pomiar. Fakt, że algorytm jest wielomianowy, oznacza, że w obwodzie jest wielomianowo wiele podstawowych bramek (czyli bramek Hadamarda H, obrotu T i kontrolowanej negacji CNOT, z których składamy wszystkie macierze unitarne, potrzebne do obliczeń). Teraz szczegółowo opiszemy, co dzieje się po kolei (od lewej).

Najpierw robimy to, co często robią na początek algorytmy kwantowe, czyli zamieniamy stan „same zera” na superpozycję wszystkich możliwych stanów bazowych, każdy z równym prawdopodobieństwem. Tyle, że my teraz zrobimy to tylko na pierwszych ℓ kubitach, tj. w pierwszym segmencie. Aby to zrobić, używamy, jak zawsze w takim przypadku, bramek Hadamarda

$$\mathrm{H} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}.$$

Bramka H przekształca $|0\rangle$ na $\mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, czyli na superpozycję $|0\rangle$ i $|1\rangle$ z równym prawdopodobieństwem. Jeśli zastosujemy bramkę H do każdego z pierwszych ℓ kubitów, to stan $|0^\ell\rangle|0^\ell\rangle$ zostanie przekształcony na stan $\sum_{a=0}^{q-1}(\frac{1}{\sqrt{2}})^\ell|a\rangle|0^\ell\rangle = \sum_{a=0}^{q-1}\frac{1}{\sqrt{q}}|a\rangle|0^\ell\rangle$, gdzie przez $|a\rangle$ rozumiemy stan określony przez binarną reprezentację a, np. dla $\ell = 4$ przez $|9\rangle = |1001\rangle$. Formalnie rzecz biorąc, powyżej przyłożyliśmy do aktualnego stanu przekształcenie będące produktem ℓ macierzy Hadamarda H i ℓ macierzy identycznościowych I. Jednak warto patrzeć na to intuicyjnie, jako na przyłożenie bramki Hadamarda do każdego kubitu oddzielnie, bo takie jest właśnie znaczenie produktu tensorowego.

Następnie w algorytmie przykładamy do wszystkich drutów bramkę, która liczy funkcję $f : \mathbb{C}^{2^{2\ell}} \to \mathbb{C}^{2^{2\ell}}$ (czyli z 2ℓ kubitów w 2ℓ kubitów) taką, że

$$f(|a\rangle|0^\ell\rangle) = |a\rangle|x^a \bmod n\rangle.$$

Przypomnijmy, że x to nasza wylosowana liczba ze zbioru $\{1,\dots,n-1\}$. Na rysunku obwód obliczający funkcję f oznaczamy przez O_f. Sprawdzenie, że obliczenie funkcji f jest unitarne oraz że da się je zrealizować obwodem o wielomianowej liczbie małych bramek, nie jest specjalnie trudne. Tutaj jednak pominiemy szczegóły. A więc po przejściu przez bramkę O_f stan jest następującą superpozycją:

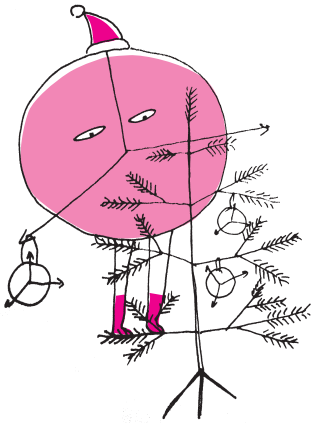
$$\sum_{a=0}^{q-1}\frac{1}{\sqrt{q}}|a\rangle|x^a \bmod n\rangle.$$

Teraz wykonujemy pomiar na drugim segmencie, czyli na drugich ℓ kubitach. W wyniku pomiaru zmierzona zostaje jakaś (nie wiemy z góry jaka!) wartość $|x^s \bmod n\rangle$. Przy czym nie wiadomo wcale, czy na pierwszych ℓ kubitach jest wartość s. Może być również tak, że jest tam wartość $s+r$, gdyż $x^{s+r} = x^s \cdot x^r \equiv x^s \cdot 1 \equiv x^s \bmod n$. Podobnie może być tam wartość $s+2r, s+3r, \dots$ Tak jak po każdym pomiarze, teraz stan układu jest superpozycją tych stanów bazowych sprzed pomiaru, które są zgodne z pomiarem.

W dalszej części rozważa się dwa przypadki. Pierwszy, gdy $r \mid q$, jest łatwiejszy, a drugi, gdy $r \nmid q$ trudniejszy. My przyjrzyjmy się pierwszemu, bo idea jest w obu przypadkach podobna, tylko w drugim jest więcej szczegółów technicznych. W tym przypadku z pomiarem $x^s \bmod n$ (dla $0 \leqslant s < r$) zgodne są wartości $s, s+r, s+2r, \dots, s+(q-r)$ na pierwszych ℓ kubitach. Zatem po pomiarze stan układu na pierwszych ℓ kubitach jest postaci

$$\frac{1}{\sqrt{q/r}}\sum_{j=0}^{(q/r)-1}|s+jr\rangle.$$

Widać teraz, że r jest jakoś związane ze stanem układu. Pytanie tylko, jak je z niego wydobyć. Jeśli zmierzymy po prostu wartość tych kubitów, to otrzymamy



pewną liczbę $s + jr$, która będzie jakąś liczbą ze zbioru $\{0, \ldots, q-1\}$, wiele nam nie powie. Nie znamy przecież s, żeby móc obliczyć jr, a tym bardziej nie znamy j, żeby obliczyć z jr wartość r. Musimy więc postępować inaczej.

Tu w sukurs przychodzi nam dziedzina, która wielu Czytelnikom zapewne wcale nie kojarzy się z faktoryzacją liczb pierwszych. Przyjrzyjmy się jeszcze raz naszemu pytaniu, z nieco innej strony. Możemy pomyśleć o naszej superpozycji jako o ciągu wartości $|a\rangle$, dla których przy większości jest współczynnik 0, ale dla niektórych niezerowy współczynnik $\frac{1}{\sqrt{q/r}}$. Te wartości o niezerowych współczynnikach powtarzają się co r i chcemy odkryć, z jakim okresem to robią. Wróćmy na chwilę do świata niekwantowego i zastanówmy się, co należy robić w takich sytuacjach, jak znaleźć okres pewnego okresowego zjawiska. Zauważmy, że nasze ucho robi to przez cały czas. Dźwięk, który słyszymy, rozkłada się bowiem na wiele składowych o różnych częstotliwościach. Ucho właśnie rozkłada dźwięk na składowe, które wyglądają jak sinusy i kosinusy. To, co robi, nazywa się w matematyce transformatą Fouriera. Okazuje się, że każdą funkcję okresową da się rozłożyć na nieskończoną sumę sinusów i kosinusów. Podobnie robi się, gdy mamy sygnał, który nie jest ciągły, lecz dyskretny. Tylko, że wtedy rozkładamy na skończoną sumę próbkowań kosinusów. Czytelnik Zapoznany Z Algebrą Liniową może sobie wyobrażać oba przekształcenia jako wyrażanie funkcji okresowej (bądź jej próbkowania) po prostu w innej bazie, złożonej z sinusów i kosinusów (bądź ich próbkowań). Ciekawostką może być, że ta sama transformata jest wykorzystywana w innych miejscach informatyki, np. w formatach jpeg lub mpeg.

A więc w świecie kwantowym, jeśli chcemy odnaleźć coś w stylu okresu w naszym stanie, też powinniśmy zastosować transformatę Fouriera, tylko że kwantową. Szczęśliwie rzeczywiście istnieje kwantowa transformata Fouriera (QFT – *quantum Fourier transform*), która okazuje się unitarna i daje się zaimplementować za pomocą wielomianowej liczby podstawowych bramek kwantowych. Przyłożenie jej do naszej konfiguracji $\frac{1}{\sqrt{q/r}} \sum_{j=0}^{(q/r)-1} |s + jr\rangle$ na wyjściu daje współczynniki przy odpowiednich okresach. Dostajemy pewną superpozycję $\sum_{j=0}^{q-1} c_j |j\rangle$. Okazuje się, że dla przypadku gdy $q \mid r$ współczynniki c_j są niezerowe jedynie dla j będących wielokrotnościami q/r. A więc możemy teraz wykonać pomiar i jesteśmy pewni, że otrzymaliśmy jakąś wielokrotność q/r. Gdy wykonamy wiele (ale wielomianowo wiele) takich pomiarów i weźmiemy minimum albo nwd, to obliczymy z dużym prawdopodobieństwem q/r. A zatem poznamy również i rząd r, co kończy naszą opowieść.

Czytelnikom Zainteresowanym Szczegółami polecamy oryginalny artykuł Petera Shora dostępny pod adresem: `arxiv.org/abs/quant-ph/9508027`.

Kwantowe wyżarzanie „klasycznej” optymalizacji

Konrad JAŁOWIECKI, Bartłomiej GARDAS, Jerzy DAJKA, Marcin MIERZEJEWSKI

Wstęp

Komputer D-Wave, opisywany w tym artykule, nie realizuje standardowego modelu obliczeń kwantowych ze stron 1–3. Realizuje tak zwany model adiabatycznych obliczeń kwantowych, który – choć jest koncepcyjnie zupełnie inny – to jest (wielomianowo) równoważny obliczeniowo modelowi standardowemu.

Wydaje się, że moc, szybkość obliczeniowa współczesnych komputerów, bazujących na krzemie, osiąga swoje plateau wynikłe z ograniczeń natury materiałowej. Jednocześnie w wielu dziedzinach życia codziennego, poczynając od prób unikania korków w planowanej podróży, poprzez minimalizację kosztochłonności produkcji aż po liczne zaawansowane zagadnienia badawcze z zakresu teorii sterowania, staramy się optymalizować nasze postępowanie. Wobec wspomnianych ograniczeń sprzętowych pozostaje nam poszukiwanie nowych algorytmów dla optymalizacji lub zupełnie nowych paradygmatów obliczeniowych – być może kwantowych?