Šiuolaikiniai kvantiniai algoritmai

R. Čiegis

April 3, 2020

Matematika visada vystėsi pagal pačios matematikos dėsnius ir vidinę logiką.

Tačiau Matematika jautė, sprendė, katalizavo, keitė mūsų realybę (o ir pati keitėsi) atsiliepdama į žmonių viltis, poreikius ir apetitus.

Kai kalbama apie matematikos vidinio vystymosi labiausiai netikėtus rezultatus, populiarus yra teiginys, kad tokie fundamentalūs matematikos skyriai, kaip abstrakčioji algebra ir skaičių teorija sukūrė matematinį šiuolaikinės kriptografijos aparatą daug, daug anksčiau, nei atsirado poreikis tiems algoritmams.

Deja, tai graži, bet neteisinga legenda. Šiuos nuostabius kriptografinius algoritmus (juos aptarėme seminare prieš tris metus) sukūrė talentingi teorinės informatikos specialistai apie 1975 metus.

O klasikinis bendrosios algebros rezultatas apie diskrečiojo logaritmavimo uždavinį buvo tik panaudotas Diffie-Hellman algoritme, kaip tinkamas įrankis (vienas iš galimų), realizuojantis vienpusę funkciją.

RSA algoritmas taip pat buvo sukurtas 1975-1977 metais.

Jo praktinei realizacijai autoriai panaudojo klasikinius skaičių teorijos rezultatus (bet ir vėl, tai buvo tik vienas iš galimų pasirinkimų pagrindinės įdėjos įgyvendinimui).

Seminarų cikle mums bus svarbesnė kita matematikos vystymosi šaka.

Skaičiai yra bazinė matematikos sąvoka, be jų negalime skaičiuoti, vėliau spręsti lygčių, taigi analizuoti, prognozuoti, būti stipresniais už kitus.

Kiek skaičių užtenka, jei norime gauti atsakymus į visus matematikai užduodamus klausimus (beje sunkiausius klausimus sugalvoja pati matematika, bet klausinėja ir žmonės, gamta, Visata)?

Labai ilgai užteko tik racionaliųjų skaičių, netgi norėjosi tikėti, kad daugiau ir kitokių skaičių nereikia (oi kaip arti teisingo atsakymo buvo senovės graikai, ir kaip ilgai dar teko matematikai vystytis, kol sužinojome PASLAPTĮ).

O kaip tada spręsti lygtį

$$x^2 = 2.$$

Akivaizdu (ne tik mums, bet ir senovės graikams), kad vienetinio kvadrato įstrižainė egzistuoja, tačiau, koks yra jos ilgis?

Matematikoje buvo žengtas labai svarbus žingsnis. Racionaliųjų skaičių aibė PRAPLĖSTA iki realiųjų skaičių aibės.

Ne pakeista, bet papildyta!

Techniškai tai galima padaryti pvz. nagrinėjant racionaliųjų skaičių Koši sekas – kai kurios iš jų konverguoja į racionaliuosius skaičius, bet kitos artėja prie kažko, kas jau nėra racionalusis skaičius. Visas tokias ribas ir įtraukiame į praplėstąją skaičių aibę – gauname realiuosius skaičius.

Realiųjų skaičių aibė jau yra pilna, bet kuri Koši seka konverguoja į realųjį skaičių.

Bendrosios algebros teorijoje tai dar nėra kūrybos pabaiga.

Ieškokime n-tosios eilės polinomo šaknų, tai yra spręskime lygtį

$$x^n + c_1 x^{n-1} + \ldots + c_{n-1} x + c_n = 0,$$

kai lygties koeficientai yra realieji skaičiai. Kiek tokia lygtis turi sprendinių, kurie yra irgi realieji skaičiai?

1 pavyzdys.

$$x^2 - 3x + 2 = 0.$$

2 pavyzdys.

$$x^2 + 1 = 0.$$

3 pavyzdys.

$$x^3 - x^2 + x - 1 = 0.$$

Apibrėžiame kompleksinius skaičius (praplėčiame realiųjų skaičių aibę).

Algebros teorija garantuoja, kad tokia praplėsta algebra būtinai turi egzistuoti.

Pagrindinė algebros teorema

Jeigu polinomo koeficientai yra kompleksiniai skaičiai, tai polinominė lygtis visada turi lygiai n sprendinius.

Kol kas "kėdžių" tik daugėja ir jos vis labiau "išsilaksto" (Ilfas ir Petrovas).

Fundamentalus matematikos žingsnis – griežtai apibrėžiamos aproksimavimo ir ribos sąvokos.

Vystoma diferencialinio ir integralinio skaičiavimo teorija, matematinės fizikos lygčių teorija.

Fizika, chemija, inžinerija tampa esmingai susietos su matematika (abipusiu ryšiu).

Bet čia svarbiausia yra tokia fundamentali įdėja.
Reikia išspręsti uždavinį

$$Pu = f, \quad u \in A.$$

Aproksimavimo teorija garantuoja, kad norimu tikslumu ε sprendinį u galime aproksimuoti elementu $U \in A_h$

$$|u - U| \leq \varepsilon, \quad A_h \subset A,$$

jei h yra pakankamai mažas parametras.

Pavyzdžiui racionalieji skaičiai sudaro tirštą realiųjų skaičių poaibį.

Sprendinio $u \in A$ artinį $U \in A_h$ randame spręsdami kitą uždavinį mažesnėje aibėje

$$P_h U = F, \quad U \in A_h.$$

Sprendinio $u \in A$ niekada ir neskačiuojame.

Prasidėjo audringas Skaičiavimo matematikos vystymosi laikotarpis, jį stebime (ir dalyvaujame jame patys) iki šiol.

1. Užtenka nedidelio, baigtinio racionaliųjų skaičių poaibio A_h ir virtualiosios tikrovės jau negalime atskirti nuo mūsų realybės.

Taigi visų racionaliųjų skaičių aibė yra net per didelis rinkinys daugelio (beveik visų) svarbiausių šios dienos matematikos uždavinių sprendimui.

2. Spręsti uždavinį tampa ekvivalentu skaičiuoti sprendinio artinį. Toks darbas formalizuojamas (A. Tiuringo mašina, Enigma istorija). Kompiuteriai tampa kasdieniniu įrankiu.

3. Lygiagretieji algoritmai.
Ar dirbtinis intelektas jau protingesnis už mus?
Dirbtinis intelektas, robotai, skaitmeninės technologijos, didieji duomenys – tai jau realybė dabar ir čia.

4. Dar didesnes mūsų svajones norint paversti realybe reikia esmingai didesnių skaičiavimo pajėgumų. Vienas iš potencialių šaltinių – kvantiniai kompiuteriai. Turime pasiruošti šiai revoliucijai jau dabar.

Įprastiniai kompiuteriai (CPU, lygiagretieji, GPU)

Kompiuteriai veikia remdamiesi šiais principais:

- Dvejetaine logika: elementarus elementas gali būti tik dvejose stabiliose būsenose – 0 ir 1.

Įprastiniai kompiuteriai (CPU, lygiagretieji, GPU)

Kompiuteriai veikia remdamiesi šiais principais:

- Dvejetaine logika: elementarus elementas gali būti tik dvejose stabiliose būsenose – 0 ir 1.
- Toks elementas (bitas) užkoduoja vieną bitą informacijos.

Įprastiniai kompiuteriai (CPU, lygiagretieji, GPU)

Kompiuteriai veikia remdamiesi šiais principais:

- Dvejetaine logika: elementarus elementas gali būti tik dvejose stabiliose būsenose – 0 ir 1.
- Toks elementas (bitas) užkoduoja vieną bitą informacijos.
- Naudodami logines operacijas (gates) NOT ir & galime realizuoti bet kokį kombinatorikos algoritmą.

- Bito būseną galima matuoti jos nepakeičiant (švelnus matavimas).

- Bito būseną galima matuoti jos nepakeičiant (švelnus matavimas).
- Bitą galima kopijuoti kiek norima kartų.

- Bito būseną galima matuoti jos nepakeičiant (švelnus matavimas).
- Bitą galima kopijuoti kiek norima kartų.
- n atskirų bitų sistemos bendras laisvės laipsnių skaičius yra lygus $2n$

- Lygiagretumas yra proporcingas procesų/procesorių skaičiui.

- Lygiagretumas yra proporcingas procesų/procesorių skaičiui.
- Turime skirti algoritmo lygiagretumo laipsnį ir lygiagrečiojo kompiuterio procesorių skaičių.

- Lygiagretumas yra proporcingas procesų/procesorių skaičiui.
- Turime skirti algoritmo lygiagretumo laipsnį ir lygiagrečiojo kompiuterio procesorių skaičių.
- Uždavinį, kurio lygiagretumo laipsnis P_1, sprendžiant lygiagrečiuoju kompiuteriu, kuris turi P_2 procesorių, spartinimo koeficientas yra nedidesnis už

$$\min(P_1, P_2).$$

- Lygiagretumas yra proporcingas procesų/procesorių skaičiui.
- Turime skirti algoritmo lygiagretumo laipsnį ir lygiagrečiojo kompiuterio procesorių skaičių.
- Uždavinį, kurio lygiagretumo laipsnis P_1, sprendžiant lygiagrečiuoju kompiuteriu, kuris turi P_2 procesorių, spartinimo koeficientas yra nedidesnis už

$$\min(P_1, P_2).$$

- Algoritmų sudėtingumo teorija pateikia NP hard sudėtingumo uždavinius – kaip juos spręsti?

Kvantiniai algoritmai

- Pagrindinis elementas yra kvantinis bitas – kubitas (qubit).

Kvantiniai algoritmai

- Pagrindinis elementas yra kvantinis bitas – kubitas (qubit).
- Kubitas yra tiesinės vektorinės erdvės virš kompleksinių skaičių elementas $|\psi\rangle \in \mathbb{C}^2$.

  Jį užrašome kaip tiesinę kombinaciją (superpoziciją) dviejų bazinių būsenų (Dirako *ket* – vektorių):

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle ,$$

  čia $\alpha_j \in \mathbb{C}$ ir tenkina normavimo sąlygą:

$$|\alpha_0|^2 + |\alpha_1|^2 = 1.$$

Bazinius *ket*–vektorius $|0\rangle$ ir $|1\rangle$ dažnai patogu užrašyti ir kaip įprastinius vektorius–stulpelius:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

n kubitų sistema

n kubitų sistema apibrėžiama panaudojant tenzorinę bazinių vektorių sandaugą, taigi laisvės laipsnių skaičius yra lygus 2^n.
Bazinių vektorių sistema yra:

$$\left\{ |\psi_0\rangle \otimes |\psi_1\rangle \otimes \cdots \otimes |\psi_{n-1}\rangle \right\}, \quad \psi_k \in \{0, 1\}.$$

Sistemos būseną $|\psi\rangle$ išreiškiame bazinių *ket*-vektorių tiesine kombinacija

$$|\psi\rangle = \sum_{j=0}^{2^n-1} \alpha_j |j\rangle, \quad \alpha_j \in \mathbb{C},$$

čia $|j\rangle = \left|\psi_0^j \psi_1^j \cdots \psi_{n-1}^j\right\rangle$.

Elementas vienu metu yra visose bazinėse būsenose $|j\rangle$.

Jeigu atliksime matavimą, tai su tikimybe $|\alpha_k|^2$ gausime reikšmę reikšmę, atitinkančią vektorių $|k\rangle$.

Po matavimo pasikeičia elemento būsena

$$|\psi\rangle = |k\rangle\,.$$

Aišku, kad elemento būsena nepriklauso nuo pasirinktos bazės, galime imti ir kitą bazę:

$$|\psi\rangle = \sum_{j=0}^{2^n-1} \widetilde{\alpha}_j \left|\tilde{j}\right\rangle, \quad \widetilde{\alpha}_j \in \mathbb{C},$$

keičiasi tik tikimybė išmatuoti vieną arba kitą rezultatą.

Bazinius *ket*-vektorius patogu užrašyti ir vektorių-stulpelių forma, pvz.:

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

Jeigu bazinių vektorių sistema yra fiksuota, tai sistemos būseną užrašysime ir vektorių-stulpelių forma

$$|\psi\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix}$$

Bet turime neužmiršti, kad ji priklauso nuo pasirinktos bazinių vektorių sistemos.

$\langle\psi|$ yra *bra*-vektorius, kuris yra dualus *ket*-vektoriui $|\psi\rangle$.

Bra-vektorių patogu vaizduoti vektoriumi-eilute

$$\langle\psi| = (\alpha_0^*, \alpha_1^*, \ldots, \alpha_{2^n-1}^*).$$

$\langle\psi|\phi\rangle$ apibrėžia skaliarinę sandaugą (Hilberto tiesinė erdvė):

$$\langle\psi|\phi\rangle = \sum_{j=0}^{2^n-1} \alpha_j^* \beta_j.$$

Baziniai *ket*-vektoriai visada yra ortonormuoti

$$\langle j|k\rangle = \delta_{jk}.$$

Jeigu dvi būsenos yra išreikštos tiesine tos pačios ortonormuotos sistemos superpozicija

$$|\psi\rangle = \sum_{j=0}^{2^n-1} \alpha_j |j\rangle, \quad |\phi\rangle = \sum_{j=0}^{2^n-1} \beta_j |j\rangle,$$

tai

$$\langle\psi|\phi\rangle = \sum_{j=0}^{2^n-1} \alpha_j^* \beta_j.$$

Lygiagretumas (geroji žinia)

Tegul U yra tiesinis operatorius, tada

$$U|\psi\rangle = \sum_{j=0}^{2^n-1} \alpha_j U|j\rangle .$$

Per vieną taktą apskaičiuojame 2^n operatoriaus U vaizdus su visais baziniais vektoriais.

Skaičiuojame bazinių vektorių vaizdų tiesinę kombinaciją, o ne tiesinės kombinacijos vaizdą.

Uždavinys P.

$$\min_{0 \le j < 2^n} f(|j\rangle).$$

Nuoseklusis algoritmas

$$
\begin{aligned}
& s = \infty; \;\; j_M = -1; \\
& for \; (\; j = 0; \; j < 2^n; \; j++)\{ \\
& \quad s_1 = f(|j\rangle); \\
& \quad if(\; s_1 < s \;)\{ \; s = s_1; \; j_M = j; \; \} \\
& \}
\end{aligned}
$$

Skaičiavimo laikas

$$T_0 = c2^n.$$

Lygiagretusis algoritmas

Turime p procesorių.

Kiekvienam procesoriui skiriame užduočių bloką ir jis vykdo pateiktąjį algoritmą.

$$T_p = \frac{T_0}{p}.$$

Kvantinis algoritmas

Pasirenkame pradinį artinį-būseną

$$|\psi\rangle = \sum_{j=0}^{2^n-1} \alpha_j |j\rangle, \quad \alpha_j \in \mathbb{C},$$

Skaičiuojame

$$U|\psi\rangle = \sum_{j=0}^{2^n-1} \alpha_j U|j\rangle.$$

Per vieną iteraciją apskaičiuojame visas $f(|j\rangle)$ reikšmes, bet šią informaciją saugome tik tiesinės kombinacijos (kvantinės būsenos) formatu.

Matuodami rezultatą sužinome tik kurią nors vieną reikšmę $f(|k\rangle)$

Po matavimo visa likusi informacija yra prarandama.

Apie tiesinius operatorius, realizuojančius duotosios funkcijos skaičiavimą, šiek tiek vėliau...

Kvantinių algoritmų ribojimai. Operatoriai

Nagrinėkime tiesinį operatorių M:

$$M|\psi\rangle = |\phi\rangle := \sum_{j=0}^{2^n-1} \beta_j |j\rangle .$$

S1. Kvantiniuose algoritmuose svarbūs ermitiniai (*Hermitian*) operatoriai $M^\dagger = M$.
Ermitiškai jungtinis operatorius $M^\dagger$ apibrėžiamas taip:

$$\langle\psi| M |\phi\rangle = \langle\phi| M^\dagger |\psi\rangle^* .$$

Tiesinius operatorius galime susieti su matricomis

$$M = \begin{pmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \\ m_{30} & m_{31} & m_{32} & m_{33} \end{pmatrix},$$

matricos koeficientai skaičiuojami taip

$$m_{kj} = \langle k | M | j \rangle .$$

Tada

$$M^\dagger = (M^T)^*.$$

Matricos koeficientų reikšmės priklauso nuo pasirinktos bazinių vektorių sistemos!

S2. Kvantiniuose algoritmuose visos transformacijos turi būti apibrėžiamos naudojant unitariuosius operatorius

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

Tada ermitiniams unitariesiams operatoriams

$$U^{-1} = U.$$

Operatoriaus tikrinės reikšmės $|\lambda_j| = 1$.

Svarbi tokių operatorių savybė, kad $UU = I$. Du kartus atlikę tą pačią transformaciją vėl gauname pradinę būseną.

Kvantiniai vartai (gates)

Pauli matricos (spino operatoriai x, y ir z kryptimis)

NOT operatorius X:

$$X: \quad |0\rangle \to |1\rangle = m_{00}|0\rangle + m_{10}|1\rangle,$$
$$|1\rangle \to |0\rangle = m_{01}|0\rangle + m_{11}|1\rangle,$$

matricinė forma $m_{kj} = \langle k|X|j\rangle$, $k, j = 0, 1$:

$$X = \begin{pmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{pmatrix} \quad \to \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Operatorius Y:

$$Y: \quad \begin{aligned} |0\rangle &\to \; i\,|1\rangle\,, \\ |1\rangle &\to -i\,|0\rangle\,, \end{aligned}$$

matricinė forma:

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},$$

Kompleksinis jungtinis operatorius Z:

$$Z: \quad \begin{aligned} |0\rangle &\to |0\rangle, \\ |1\rangle &\to -|1\rangle, \end{aligned}$$

matricinė forma:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Matavimų atlikimas

Imkime ermitinį operatorių L, kuris apibrėžia mus dominantį dydį.

- Surandame operatoriaus L tikrines būsenas (*ket*-vektorius):

$$L\left|\psi_j\right\rangle = \lambda_j \left|\psi_j\right\rangle, \quad j = 0, \ldots, 2^n - 1.$$

Matavimų atlikimas

Imkime ermitinį operatorių L, kuris apibrėžia mus dominantį dydį.

- Surandame operatoriaus L tikrines būsenas (*ket*-vektorius):

$$L\left|\psi_j\right\rangle = \lambda_j \left|\psi_j\right\rangle, \quad j = 0, \ldots, 2^n - 1.$$

- Būseną $|\psi\rangle$ užrašome tikrinių vektorių bazėje

$$|\psi\rangle = \sum_{j=0}^{2^n-1} \alpha_j \left|\psi_j\right\rangle.$$

Matavimų atlikimas

Imkime ermitinį operatorių L, kuris apibrėžia mus dominantį dydį.

- Surandame operatoriaus L tikrines būsenas (*ket*-vektorius):

$$L\left|\psi_j\right\rangle = \lambda_j \left|\psi_j\right\rangle, \quad j = 0, \ldots, 2^n - 1.$$

- Būseną $|\psi\rangle$ užrašome tikrinių vektorių bazėje

$$|\psi\rangle = \sum_{j=0}^{2^n-1} \alpha_j \left|\psi_j\right\rangle.$$

- Atliekame matavimą, su tikimybe $|\alpha_k|^2$ gauname rezultatą λ_k, o sistemos būsena pasikeičia į

$$|\psi\rangle = \left|\psi_k\right\rangle.$$

Susietos būsenos (entaglement)

Nagrinėkime dviejų kubitų sistemos būseną

$$|\psi\rangle = \frac{1}{\sqrt{2}}\big(\,|00\rangle + |01\rangle\,\big).$$

Ją galime užrašyti kaip dviejų atskirų kubitų tenzorinę sandaugą

$$|\psi\rangle = |0\rangle \otimes \frac{1}{\sqrt{2}}\big(\,|0\rangle + |1\rangle\,\big).$$

Abu kubitai yra nepriklausomi, vieno iš jų reikšmės matavimas nepakeičia kito kubito būsenos.

Nagrinėkime dviejų kubitų sistemos EPR būseną (Albert Einstein, Boris Podolsky, Nathan Rosen pavyzdys)

$$|\psi\rangle = \frac{1}{\sqrt{2}}\big(\,|00\rangle + |11\rangle\,\big).$$

Abu kubitai yra surišti, matuojant vieną kubitą, tuo pat metu matuojamas ir kitas kubitas.

Kvantiniai vartai (gates)

Kontroliuojamas NOT operatorius C_{NOT} (dviejų kubitų sistemai):

$$\begin{aligned} C_{NOT}: \quad & |00\rangle \to |00\rangle \\ & |01\rangle \to |01\rangle \\ & |10\rangle \to |11\rangle \\ & |11\rangle \to |10\rangle \end{aligned}$$

Matricinė operatoriaus forma:

$$C_{NOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Galime C_{NOT} užrašyti ir tokia patogia forma

$$C_{NOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$$

Tada šių vartų matricinę formą gauname tiesiogiai

$$C_{NOT} = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Hadamard'o vartai:

$$H: \qquad |0\rangle \to \frac{1}{\sqrt{2}}\big(\,|0\rangle + |1\rangle\,\big)$$
$$|1\rangle \to \frac{1}{\sqrt{2}}\big(\,|0\rangle - |1\rangle\,\big).$$

Matricinė operatoriaus forma:

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

Pavyzdžiai: EPR būsenos generavimas

Pradinė dviejų kubitų sistemos būsena:

$$|\psi_0\rangle = |00\rangle\,.$$

Pavyzdžiai: EPR būsenos generavimas

Pradinė dviejų kubitų sistemos būsena:

$$|\psi_0\rangle = |00\rangle\,.$$

Pirmam kubitui pritaikome Hadamard transformaciją

$$|\psi_1\rangle = H \otimes I\ |00\rangle = \frac{1}{\sqrt{2}}\big(\,|00\rangle + |10\rangle\,\big).$$

Pavyzdžiai: EPR būsenos generavimas

Pradinė dviejų kubitų sistemos būsena:

$$|\psi_0\rangle = |00\rangle\,.$$

Pirmam kubitui pritaikome Hadamard transformaciją

$$|\psi_1\rangle = H \otimes I\ |00\rangle = \frac{1}{\sqrt{2}}\big(\,|00\rangle + |10\rangle\,\big).$$

Abiems kubitams pritaikome C_{NOT} transformaciją

$$|\psi_2\rangle = C_{NOT}\,|\psi_1\rangle = \frac{1}{\sqrt{2}}\big(\,|00\rangle + |11\rangle\,\big).$$

Visų būsenų superpozicijos generavimas

$$H^{\otimes n} = H \otimes H \otimes \cdots \otimes H.$$

Visų būsenų superpozicijos generavimas

$$H^{\otimes n} = H \otimes H \otimes \cdots \otimes H.$$

$$H|0\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big).$$

Visų būsenų superpozicijos generavimas

$$H^{\otimes n} = H \otimes H \otimes \cdots \otimes H.$$

$$H|0\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big).$$

$$\begin{aligned} H^{\otimes 2}|00\rangle &= \frac{1}{2}\big(|0\rangle + |1\rangle\big) \otimes \big(|0\rangle + |1\rangle\big) \\ &= \frac{1}{2}\big(|00\rangle + |01\rangle + |10\rangle + |11\rangle\big). \end{aligned}$$

Visų būsenų superpozicijos generavimas

$$H^{\otimes n} = H \otimes H \otimes \cdots \otimes H.$$

$$H\,|0\rangle = \frac{1}{\sqrt{2}}\big(\,|0\rangle + |1\rangle\,\big).$$

$$\begin{aligned} H^{\otimes 2}\,|00\rangle &= \frac{1}{2}\big(\,|0\rangle + |1\rangle\,\big) \otimes \big(\,|0\rangle + |1\rangle\,\big) \\ &= \frac{1}{2}\big(\,|00\rangle + |01\rangle + |10\rangle + |11\rangle\,\big). \end{aligned}$$

$$H^{\otimes n}\,|00\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{j\in\{0,1\}^n} |j\rangle\,.$$

Funkcijos reikšmių skaičiavimo kvantinis operatorius

$$f : \quad \{0,1\}^n \to \quad \{0,1\}^m$$

Bulio funkcijos reikšmių skaičiavimas

Galime sudaryti keturias skirtingas vieno kontamojo Bulio funkcijas. Nagrinėkime vieną iš jų:

$$\begin{aligned} f_1: \quad 0 &\to 0, \\ 1 &\to 0, \end{aligned}$$

jos kvantinio analogo matricinė forma

$$M_{f_1} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix},$$

M nėra unitarioji.

Nagrinėkime 2 kubitų operatorių

$$U_{f_1}: \quad |x, y\rangle \rightarrow |x, y \oplus f_1(x)\rangle .$$

Tada

$$U_{f_1}: \quad |x, 0\rangle \rightarrow |x, f_1(x)\rangle .$$

Bendruoju atveju

$$U_f: \quad |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle ,$$

kur x yra n kubitų, o y yra m kubitų.
Turime savybę

$$U_f U_f = I,$$

nes $f(x) \oplus f(x) = 0$.

Patikrinkime, kad Bulio funkcijos f_1 operatorius U_{f_1} yra simetrinis ir unitarusis.

$$\begin{aligned} |0,0\rangle &\to |0,0\rangle, \quad |0,1\rangle \to |0,1\rangle, \\ |1,0\rangle &\to |1,0\rangle, \quad |1,1\rangle \to |1,1\rangle, \end{aligned}$$

operatoriaus matrica:

$$U_{f_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$U_{f_1} = I \otimes I.$$

Pratybos

$$f_2: \quad 0 \to 1,$$
$$1 \to 0,$$

$$U_{f_2} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$f_3: \quad 0 \to 0,$$
$$1 \to 1,$$

$$U_{f_3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$U_{f_3} = C_{NOT}.$$

$$f_4: \quad 0 \to 1,$$
$$1 \to 1,$$

$$U_{f_4} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$U_{f_4} = I \otimes X.$$

Kubitų negalima kopijuoti

Tarkime, kad U yra tokia funkcija, jog kiekvienam kubitui a:

$$U: \quad |a,0\rangle \ \rightarrow |a,a\rangle$$

Tada imkime kubitą $c = \alpha|0\rangle + \beta|1\rangle$.

$$U: \ |00\rangle \ \rightarrow \ |00\rangle, \quad |10\rangle \ \rightarrow \ |11\rangle.$$

Kadangi U yra tiesinis operatorius, tai

$$U: \ |c,0\rangle \ \rightarrow \ \alpha|00\rangle + \beta|11\rangle.$$

O turėtume gauti:

$$U: \ |c,0\rangle \ \rightarrow \ |c,c\rangle = \alpha^2|00\rangle + \alpha\beta(|01\rangle + |10\rangle) + \beta^2|11\rangle.$$

Užduotis – sudarykite operatorių, realizuojantį Bulio logikos operaciją/funkciją AND

Taip gausime labai svarbius Toffoli vartus!
Jų vienų užtenka, jei norime realizuoti bet kokią kombinatorinę loginę grandinę.

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

Sugrupuojame pagal pirmą kubitą

$$
\begin{aligned}
|000\rangle &\to |000\rangle, & |001\rangle &\to |001\rangle,\\
|010\rangle &\to |010\rangle, & |011\rangle &\to |011\rangle,
\end{aligned}
$$

$$|0\rangle\langle 0| \otimes I \otimes I$$

$$|100\rangle \rightarrow |100\rangle, \quad |101\rangle \rightarrow |101\rangle,$$
$$|110\rangle \rightarrow |111\rangle, \quad |111\rangle \rightarrow |110\rangle.$$

$$|1\rangle\langle 1| \otimes C_{NOT}$$

$$T = \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & X \end{pmatrix},$$

$$T : |11z> \rightarrow |11\bar{z}>$$

Teleportacijos galimybė

Realizuokite kvantinį operatorių (vartus), kuris sukeičia du kubitus vietomis

$$SWAP: \ |q_1 q_2\rangle \ \rightarrow \ |q_2 q_1\rangle$$

Ši transformacija yra leistina, nes visą laiką egzistuoja tik po vieną kubito kopiją.

Tai vis dar ne teleportacija, bet esminiai jos bruožai jau matosi. Klausimas ar kubitai pasikeitė vietomis, ar pasikeitė tik jų būsenos?

$$q_1 = a\,|0\rangle + b\,|1\rangle\,, \quad q_2 = c\,|0\rangle + d\,|1\rangle\,.$$

Tada

$$|q_1 q_2\rangle = |q1\rangle \otimes |q2\rangle = ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle\,,$$
$$|q_2 q_1\rangle = |q2\rangle \otimes |q1\rangle = ac\,|00\rangle + ad\,|10\rangle + bc\,|01\rangle + bd\,|11\rangle\,.$$

Gauname operatoriaus *SWAP* apibrėžimo sąlygas

$$|00\rangle \to |00\rangle\,, \quad |01\rangle \to |10\rangle\,, \quad |10\rangle \to |01\rangle\,, \quad |11\rangle \to |11\rangle\,.$$

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Teleportacija

Paruošiame du kubitus, kurie yra susiję EPR sąryšiu:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle \right).$$

Vieną kubitą atiduodame Alice, o kitą Bobui.

Po to jie išvažiuoja gyventi į skirtingus miestus. Alice ir Bobas gali bendrauti tik naudodami paprastą tradicinį ryšį (pvz. internetą).

Alice turi vertingą kubitą

$$|\psi_0\rangle = a\,|0\rangle + b\,|1\rangle\,,$$

kurį ji nori persiųsti Bobui.

Alice ir pati nežino koeficientų a, b reikšmių.

Prisiminkime, kad kubitų negalima kopijuoti.

Taigi turime trijų kubitų sistemą:

$$|\psi_0\rangle \otimes |\psi\rangle = \frac{1}{\sqrt{2}}\big(a\,|000\rangle + a\,|011\rangle + b\,|100\rangle + b\,|111\rangle\big).$$

Taigi turime trijų kubitų sistemą:

$$|\psi_0\rangle \otimes |\psi\rangle = \frac{1}{\sqrt{2}}\left(a\,|000\rangle + a\,|011\rangle + b\,|100\rangle + b\,|111\rangle\right).$$

Alice atlieka dvi transformacijas (su savais kubitais):

$$C_{NOT} \otimes I\,|\psi_0\rangle \otimes |\psi\rangle = \frac{1}{\sqrt{2}}\left(a\,|000\rangle + a\,|011\rangle + b\,|110\rangle + b\,|101\rangle\right) = |\psi_1\rangle\,.$$

Taigi turime trijų kubitų sistemą:

$$|\psi_0\rangle \otimes |\psi\rangle = \frac{1}{\sqrt{2}}\big(a\,|000\rangle + a\,|011\rangle + b\,|100\rangle + b\,|111\rangle\big).$$

Alice atlieka dvi transformacijas (su savais kubitais):

$$C_{NOT} \otimes I\,|\psi_0\rangle \otimes |\psi\rangle = \frac{1}{\sqrt{2}}\big(a\,|000\rangle + a\,|011\rangle + b\,|110\rangle + b\,|101\rangle\big) = |\psi_1\rangle\,.$$

ir

$$H \otimes I \otimes I\ |\psi_1\rangle = \frac{1}{2}\big(a\,|000\rangle + a\,|100\rangle + a\,|011\rangle + a\,|111\rangle$$
$$b\,|010\rangle - b\,|110\rangle + b\,|001\rangle - b\,|101\rangle\big) = |\psi_2\rangle\,.$$

Sugrupuojame gautosios trijų kubitų sistemos būsenos narius:

$$|\psi_2\rangle = \frac{1}{2}\Big(|00\rangle \otimes \big(a\,|0\rangle + b\,|1\rangle\big) + |01\rangle \otimes \big(a\,|1\rangle + b\,|0\rangle\big) + |10\rangle \otimes \big(a\,|0\rangle - b\,|1\rangle\big) + |11\rangle \otimes \big(a\,|1\rangle - b\,|0\rangle\big)\Big).$$

Teleportacijai pasiruošta! Alice du kubitai yra keturių galimų bazinių būsenų superpozicijoje.

Alice matuoja savo turimų qubitų reikšmę ir su vienoda tikimybe gauna kurį nors vieną rezultatą 0, 1, 2 arba 3.

Alice atlikus matavimą, Bobo kubito būsena irgi pasikeičia, kubitas projektuojamas į matavimo rezultatą atitinkančią būseną.

Teleportacija įvyko!

Alice nusiunčia matavimo rezultatą Bobui (atviruoju kanalu).

Alice nusiunčia matavimo rezultatą Bobui (atviruoju kanalu).

Bobas transformuoja savo turimą kubitą į reikalingą būseną:

$$
\begin{array}{llll}
00 & a|0\rangle + b|1\rangle & I & a|0\rangle + b|1\rangle, \\
01 & a|1\rangle + b|0\rangle & X & a|0\rangle + b|1\rangle, \\
10 & a|0\rangle - b|1\rangle & Z & a|0\rangle + b|1\rangle, \\
11 & a|1\rangle - b|0\rangle & Y & -i(a|0\rangle + b|1\rangle).
\end{array}
$$

Po teleportacijos Bobas saugo Alice kubito kopiją. Bet tai neprieštarauja įrodytai teoremai apie kubito kopijavimo negalimumą.

Alice nebeturi savo kubito, ji ir nesužinojo, kokia buvo jo pradinė būsena.

Taigi ir po teleportacijos egzistuoja vienintelė kubito kopija.

Deutsch algoritmas

Nagrinėkime Bulio funkciją f. Turime tik juodąją dėžę – funkcijos programinę realizaciją. Reikia apskaičiuoti

$$f(0) \oplus f(1).$$

Diskretusis algoritmas skaičiuoja dvi funkcijos reikšmes ir tada jas sumuojame.

Parodysime, kad kvantiniame algoritme užtenka vieną kartą pritaikyti funkcijos kvantinį operatorių.

Imkime du kubitus

$$|\psi_0\rangle = |01\rangle\,.$$

Abiems kubitams atlikime Hadamard'o transformaciją

$$H \otimes H\ |\psi_0\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) := |\psi_1\rangle\,.$$

Funkcijos f kvantinis operatorius

$$U_f : \quad |xy\rangle \quad \rightarrow \quad |x, y \oplus f(x)\rangle\,.$$

$$U_f : \quad |x\,0\rangle \quad \to \quad |x, f(x)\rangle\,,$$
$$U_f : \quad |x\,1\rangle \quad \to \quad \left|x, \overline{f(x)}\right\rangle.$$

Tada gauname, kad

$$U_f : \quad |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad \to \quad |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \quad \text{jei } f(x) = 0,$$
$$U_f : \quad |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad \to \quad |x\rangle \otimes \frac{|1\rangle - |0\rangle}{\sqrt{2}}, \quad \text{jei } f(x) = 1.$$

$$U_f : \quad |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad \to \quad (-1)^{f(x)}\,|x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}},$$

Skaičiuojame $|\psi_2\rangle = U_f |\psi_1\rangle$

$$|\psi_2\rangle = \begin{cases} \pm\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}, & \text{if } f(0) = f(1), \\ \pm\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}, & \text{if } f(0) \neq f(1) \end{cases}.$$

Dabar pirmajam kubitui pritaikome Hadamard'o transformaciją

$$|\psi_3\rangle = H \otimes I\, |\psi_2\rangle\,.$$

$$|\psi_3\rangle = \begin{cases} \pm\,|0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}, & \text{if } f(0) = f(1), \\ \pm\,|1\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}, & \text{if } f(0) \neq f(1) \end{cases}.$$

Išmatuojame pirmojo kubito reikšmę.

Deutsch-Jozsa algoritmas

Uždavinį pasunkiname: Bulio funkcija $f(x)$ yra apibrėžta visiems $X = \{0, 1, \ldots, 2^n - 1\}$.
Taip pat žinome, kad $f(x)$ yra arba konstanta, arba subalansuota.
Subalansuota funkcija lygi nuliui 2^{n-1} taške ir lygi vienetui kituose 2^{n-1} taškuose.

Reikia nustatyti, kokią turime funkciją, jeigu žinome tik jos realizaciją (deterministinę f arba kvantinę U_f).
Ankstesniame pavyzdyje $n = 1$.

Deterministiniu algoritmu blogiausiu atveju turėsime apskaičiuoti $2^{n-1} + 1$ funkcijos reikšmę.

Ką sužinojome naudodami Deutsch algoritmą?

1. Kvantinio algoritmo lygiagretumas gaunamas skaičiuojant funkcijos reikšmes abiejuose taškuose $|0\rangle$ ir $|1\rangle$ tuo pačiu metu. Duomenų kubitą $|0\rangle$ praleidome pro Hadamard'o vartus ir gavome abiejų bazinių būsenų tiesinę kombinaciją

$$H|0\rangle \quad \rightarrow \quad \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big).$$

2. Rezultato kubitą pasirinkome $|1\rangle$ ir jį taip pat praleidome pro Hadamard'o vartus

$$H|1\rangle \quad \rightarrow \quad \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big).$$

Toks pasirinkimas gali būti siejamas su U_f operatoriaus tikrinių funkcijų panaudojimu

$$U_f: \quad |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad \to \quad (-1)^{f(x)} \left(|x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right),$$

Skaičiuojame $|\psi_2\rangle = U_f |\psi_1\rangle$, kai $\psi_1 = \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle)$

$$|\psi_2\rangle = \begin{cases} \pm \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}, & \text{kai} \quad f(0) = f(1), \\ \pm \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}, & \text{kai} \quad f(0) \neq f(1) \end{cases}.$$

Tiek pirmojo, tiek antrojo kubito matavimai neatsako į mūsų klausimą, kokią turime funkciją.

Pirmąjį kubitą vėl praleidžiame pro Hadamard'o vartus, gauname būseną

$$|\psi_3\rangle = \begin{cases} \pm|0\rangle \otimes \dfrac{|0\rangle - |1\rangle}{\sqrt{2}}, & \text{kai } f(0) = f(1), \\ \pm|1\rangle \otimes \dfrac{|0\rangle - |1\rangle}{\sqrt{2}}, & \text{kai } f(0) \neq f(1) \end{cases}.$$

Išmatuojame pirmojo kubito reikšmę (jis buvo skirtas pradiniams duomenims) ir sužinome funkcijos tipą

Dabar pritaikykime šio algoritmo principus apibendrintam uždaviniui. Gausime naują algoritmą.

Mūsų svarbiausias tikslas – parodyti, kad kiekvienam naujam uždaviniui nereikės esmingai kitokio algoritmo.

Aišku, kad sukonstruosime tik formalų algoritmą. Jo teisingumą dar turėsime ištirti ir tik tada galėsime padaryti išvadą, ar toks kvantinis algoritmas išsprendža duotąjį uždavinį.

1. Kvantinio algoritmo lygiagretumas gaunamas skaičiuojant funkcijos reikšmes visuose 2^n taškuose tuo pačiu metu. Duomenų kubitus $|0\ldots0\rangle$ praleidžiame pro Hadamard'o vartus ir gauname visų bazinių būsenų tiesinę kombinaciją

$$H^{\otimes n}|0\ldots0\rangle \;\rightarrow\; \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle = \left|\widetilde{\psi}_1\right\rangle.$$

2. Rezultato kubitą pasirenkame $|1\rangle$ ir jį taip pat praleidžiame pro Hadamard'o vartus

$$H|1\rangle \;\rightarrow\; \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big).$$

3. Skaičiuojame $|\psi_2\rangle = U_f |\psi_1\rangle$, kai

$$|\psi_1\rangle = \left|\widetilde{\psi}_1\right\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Gauname bazinių būsenų vaizdų superpoziciją

$$|\psi_2\rangle = \sum_{x\in\{0,1\}^n} \frac{(-1)^{f(x)}|x\rangle}{\sqrt{2^n}} \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

4. Gautosios sistemos duomenų kubitus praleidžiame pro Hadamard'o vartus

$$H^{\otimes n} \otimes I \quad |\psi_2\rangle \quad \rightarrow \quad |\psi_3\rangle .$$

Pirmiausia apskaičiuosime vienos būsenos vaizdą. Priminsime, kad

$$H|x\rangle = \frac{1}{\sqrt{2}} \sum_{z\in\{0,1\}} (-1)^{xz} |z\rangle .$$

Tada gauname

$$H^{\otimes n}|x_1 \dots x_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{z_1 \dots z_n} (-1)^{x_1 z_1 + \dots + x_n z_n} |z_1 \dots z_n\rangle$$

Šią lygybę užrašysime kompaktiškiau

$$H^{\otimes n}|x\rangle = \frac{\sum_z (-1)^{x\cdot z} |z\rangle}{\sqrt{2^n}}.$$

Atlikę visas transformacijas gauname tokią būseną

$$|\psi_3\rangle = \sum_z \sum_x \frac{(-1)^{x\cdot z + f(x)} |z\rangle}{\sqrt{2^n}} \otimes \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right].$$

5. Išmatuojame duomenų kubitus.

Nagrinėkime pirmąjį atvejį, kai f yra konstanta. Tada gauname, kad būsenos $|0\cdots 0\rangle$ amplitudė yra lygi vienetui. Reiškia matuodami gausime visų kubitų reikšmes lygias nuliui.

Dabar nagrinėkime antrąjį atvejį, kai f yra subalansuota. Tada skaičiuojant būsenos $|0\cdots 0\rangle$ amplitudę 1 ir -1 kompensuojasi ir gauname, jog šios būsenos amplitudė yra lygi nuliui. Taigi atliekant matavimus būtinai gausime, kad bent vieno kubito reikšmė yra lygi vienetui.

Informacijos paieškos algoritmas

- Turime sąrašą argumento reikšmių $X = \{0, 1, \ldots, 2^n - 1\}$. Pažymėkime $N = 2^n$.

Informacijos paieškos algoritmas

- Turime sąrašą argumento reikšmių $X = \{0, 1, \ldots, 2^n - 1\}$. Pažymėkime $N = 2^n$.
- Turime funkciją $P(x) \in \{0, 1\}, \quad x \in X$.
  Reikia rasti tokius $x \in X_1 \subset X$, kai $P(x) = 1$.

Informacijos paieškos algoritmas

- Turime sąrašą argumento reikšmių $X = \{0, 1, \ldots, 2^n - 1\}$. Pažymėkime $N = 2^n$.
- Turime funkciją $P(x) \in \{0, 1\}, \quad x \in X$.
  Reikia rasti tokius $x \in X_1 \subset X$, kai $P(x) = 1$.
- Nesutvarkytose aibėse paieškos algoritmų sudėtingumas yra $\mathcal{O}(N)$.

Apibrėžiame unitarųjį operatorių $U_P(x)$, skirtą funkcijos $P(x)$ reikšmių skaičiavimui

$$U_P(x,\phi): \quad |x,\phi\rangle \quad \rightarrow \quad |x,\phi \oplus P(x)\rangle\,,$$

čia $|x,P(x)\rangle = |x\rangle \otimes |P(x)\rangle$.

- Generuojame pradinių duomenų vektorių

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \, .$$

- Generuojame pradinių duomenų vektorių

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \, .$$

- Apskaičiuojame operatoriaus U_P vaizdą – tai tik vienas kvantinio algoritmo veiksmas

$$U_P : \quad |\psi, 0\rangle \quad \rightarrow \quad \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x, P(x)\rangle \, .$$

Pasiruošimas matavimo operacijai

Turime kvantinę būseną

$$|\psi\rangle = \sum_{x=0}^{N-1} \alpha_x |x\rangle, \quad a_x \geq 0,\ 0 \leq x < N.$$

Pakeičiame amplitudžių ženklus:

$$\alpha_x = \begin{cases} \alpha_x, & \text{jeigu } P(x) = 0, \\ -\alpha_x, & \text{jeigu } P(x) = 1. \end{cases}$$

Atliekame inversijos transformaciją vidurkio atžvilgiu

$$A = \frac{1}{N}\sum_{x=0}^{N-1} \alpha_x, \quad \alpha_x := 2A - \alpha_x, \quad x = 0, \ldots, N-1.$$

Tarkime, kad $P(x_k) = 1$, o visuose kituose taškuose $P(x_j) = 0$.
Taip pat tarkime, kad pradiniu momentu $\alpha_x > 0$ yra vienodi.

Atlikus vieną iteraciją $\alpha_k \approx 3\alpha_j$, po antros iteracijos koeficientas α_k dar padidėja kitų koeficientų atžvilgiu $\alpha_k \approx 5\alpha_j$.

Atlikus $\frac{\pi}{8}\sqrt{N}$ iteracijų $\alpha_k^2 \approx \sum_{j\neq k} \alpha_j^2$.

Po $\frac{\pi}{4}\sqrt{N}$ iteracijų $\alpha_k^2 \approx 1 - \frac{1}{N}$.

Toliau didinant iteracijų skaičių koeficientų pasiskirstymas vėl artėja prie pradinio tolygaus pasiskirstymo.

Nagrinėkime pavyzdį: $n = 6,\ N = 64$. Tada gauname tokias tikimybes, kad išmatuosime reikalingą kubitą

$$\begin{aligned}
\text{it} = 0 \quad & p = 0.015625, \\
\text{it} = 1 \quad & p = 0.134827, \\
\text{it} = 2 \quad & p = 0.343895, \\
\text{it} = 3 \quad & p = 0.59138, \\
\text{it} = 6 \quad & p = 0.996586, \\
\text{it} = 12 \quad & p = 7.05e - 05.
\end{aligned}$$

Matuojame paskutinį kubitą. Jeigu gauname vienetą, tai turime poerdvį

$$\frac{1}{\sqrt{2^k}} \sum_{i=1}^{k} |x_i, 1\rangle .$$

Tada išmatavę likusius n kubitus gauname vieną iš sprendinių x_i.

Jeigu matuodami paskutinį kubitą gauname nulinę reikšmę, tai pakartojame skaičiavimus.

?

Tikėjausi klausimo – kodėl turime $\sqrt{N}$ kartų atlikti kvantinės funkcijos U_P transformaciją, kai jau po pirmo karto apskaičiavome funkcijos $P(x)$ reikšmes visuose N taškų?

Ženklų pakeitimo transformacija

Imkime rezultato kontrolinį kubitą tokį

$$|\phi\rangle = H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Pažymėkime X_0 aibę taškų, kur $P(x) = 0$, ir X_1 aibę taškų, kur $P(x) = 1$.

Skaičiuokime funkcijos vaizdą

$$U_P(|\psi,\phi\rangle) = \sum_{x=0}^{N-1} \alpha_x U_P |x,\phi\rangle = \sum_{x\in X_0} \alpha_x U_P |x,\phi\rangle + \sum_{x\in X_1} \alpha_x U_P |x,\phi\rangle$$

$$= \frac{1}{\sqrt{2}}\Big(\sum_{x\in X_0} \alpha_x |x, 0\oplus 0\rangle - \sum_{x\in X_0} \alpha_x |x, 1\oplus 0\rangle + \sum_{x\in X_1} \alpha_x |x, 0\oplus 1\rangle$$

$$- \sum_{x\in X_1} \alpha_x |x, 1\oplus 1\rangle \Big) = \sum_{x\in X_0} \alpha_x |x,\phi\rangle + \sum_{x\in X_1} (-\alpha_x) |x,\phi\rangle .$$

Inversijos transformacija

$$D: \quad \sum_{x=0}^{N-1} \alpha_x |x\rangle \quad \to \quad \sum_{x=0}^{N-1} (2A - \alpha_x) |x\rangle .$$

Tai atlieka operatorius

$$D = \begin{pmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \cdots & \frac{2}{N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} - 1 \end{pmatrix}$$

Nesunku patikrinti, kad D yra unitarusis:

$$D = D^\dagger, \quad DD = I.$$

Transformaciją D realizuojame taip:

$$D = H^{\otimes n} R H^{\otimes n}, \qquad n = \log N,$$

$$R = Z \otimes I \otimes \cdots \otimes I.$$

Priminsime, kad Z yra Pauli opertaorius

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Įrodysime D operatoriaus skaičiavimo teisingumą.

Išreiškiame R kaip dviejų operatorių tiesinę kombinaciją $R = R' - I$, kur

$$R' = \begin{pmatrix} 2 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & 0 \end{pmatrix}.$$

Tada gauname

$$H^{\otimes n} R H^{\otimes n} = H^{\otimes n}(R' - I)H^{\otimes n} = H^{\otimes n} R' H^{\otimes n} - I = D.$$

Specialieji skyriai.

1. Pasikeitimo raktais kvantinis algoritmas.
2. Greitoji Furjė transformacija.
3. Skaičių faktorizavimas. Shor algoritmas. RSA saugumas.

Skaičių faktorizavimas. Shor algoritmas

Reikia išskaidyti duotąjį skaičių N į pirminius daugiklius $N = pq$.

Skaičių faktorizavimas. Shor algoritmas

Reikia išskaidyti duotąjį skaičių N į pirminius daugiklius $N = pq$.

1. Pasirenkame skaičių a, tarkime, kad $\gcd(a, N) = 1$.
2. Apibrėžiame funkciją $f(x) = a^x \bmod N$.
3. Randame funkcijos $f(x)$ periodą $f(x + r) = f(x)$.
4. Jeigu $r = 2l - 1$, tai grįžtame į 1 žingsnį.
5. Jeigu $a^{r/2} \equiv -1 \pmod N$, tai grįžtame į 1 žingsnį.
6. Skaičiuojame skaičiaus N pirminį daugiklį $\gcd(a^{r/2} \pm 1, N)$.

Skaičių faktorizavimas. Shor algoritmas

Reikia išskaidyti duotąjį skaičių N į pirminius daugiklius $N = pq$.

1. Pasirenkame skaičių a, tarkime, kad $\gcd(a, N) = 1$.
2. Apibrėžiame funkciją $f(x) = a^x \bmod N$.
3. Randame funkcijos $f(x)$ periodą $f(x + r) = f(x)$.
4. Jeigu $r = 2l - 1$, tai grįžtame į 1 žingsnį.
5. Jeigu $a^{r/2} \equiv -1 \pmod N$, tai grįžtame į 1 žingsnį.
6. Skaičiuojame skaičiaus N pirminį daugiklį $\gcd(a^{r/2} \pm 1, N)$.

$N = 15, \quad a = 2, \quad r = 4, \quad \gcd(2^2 \pm 1, 15).$

1. Randame tokį $M = 2^m$, kad $N^2 \leq M < 2N^2$.

1. Randame tokį $M = 2^m$, kad $N^2 \leq M < 2N^2$.
2. Paruošiame pradinį *ket*-vektorių

$$|\psi\rangle = \frac{1}{M} \sum_{x=0}^{M-1} |x\rangle \, .$$

1. Randame tokį $M = 2^m$, kad $N^2 \leq M < 2N^2$.
2. Paruošiame pradinį *ket*-vektorių

$$|\psi\rangle = \frac{1}{M} \sum_{x=0}^{M-1} |x\rangle \,.$$

3. Skaičiuojame funkcijos $f(x)$ reikšmes kiekviename taške

$$U_f(|\psi, 0\rangle) = \frac{1}{M} \sum_{x=0}^{M-1} |x, f(x)\rangle \,.$$

4. Išmatuojame funkcijos $f(x)$ reikšmę, gauname $f(\widetilde{x})$, o sistemą pervedame į būseną

$$|\psi\rangle = C \sum_{x=0}^{M-1} g(x)\,|x, f(\widetilde{x})\rangle\,, \quad g(x) = \begin{cases} 1, & \text{jeigu } f(x) = f(\widetilde{x}), \\ 0, & \text{jeigu } f(x) \neq f(\widetilde{x}). \end{cases}$$

4. Išmatuojame funkcijos $f(x)$ reikšmę, gauname $f(\widetilde{x})$, o sistemą pervedame į būseną

$$|\psi\rangle = C \sum_{x=0}^{M-1} g(x)\,|x, f(\widetilde{x})\rangle\,, \quad g(x) = \begin{cases} 1, & \text{jeigu } f(x) = f(\widetilde{x}), \\ 0, & \text{jeigu } f(x) \neq f(\widetilde{x}). \end{cases}$$

$g(x)$ yra periodinė funkcija $g(x+r) = g(x)$, bet galime atlikti tik vieną matavimą.

5. Pritaikome kvantinę Furje transformaciją

$$\sum_{x=0}^{M-1} g(x)\,|x\rangle \quad \rightarrow \quad \sum_{y=0}^{M-1} G(y)\,|y\rangle\,.$$

Funkcijos spektras $G(y)$ klasterizuojasi apie $j\frac{2^m}{r}$ dažnius.

5. Pritaikome kvantinę Furje transformaciją

$$\sum_{x=0}^{M-1} g(x)\,|x\rangle \quad \rightarrow \quad \sum_{y=0}^{M-1} G(y)\,|y\rangle\,.$$

Funkcijos spektras $G(y)$ klasterizuojasi apie $j\frac{2^m}{r}$ dažnius.

6. Išmatuojame duotąją būseną, gauname $y \approx j\frac{2^m}{r}$.

Skaičiuojame trupmeną $\frac{y}{2^m}$ ir panaudodami grandininių trupmenų aproksimaciją randame kandidatą periodui r (skaičiavimai atliekami CPU).

Kvantinio algoritmo sudėtingumas $(\log N)^3$.