Matematika vakar, šiandien ir ... visada, o kur kvantiniai algoritmai?

R. Čiegis

Vilniaus Gedimino technikos universitetas
e-mail: rc@vgtu.lt

Rugsėjo 10 d., 2019, Vilnius

Matematika visada vystėsi (taip buvo, yra ir bus) 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:

$$\left\{ |\psi_0\rangle \otimes |\psi_1\rangle \otimes \cdots \otimes |\psi_{2^n-1}\rangle \right\}, \quad \psi_k \in \{0, 1\}.$$

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 $|j\rangle = \left|\psi_0^j \psi_1^j \cdots \psi_{2^n-1}^j\right\rangle$.

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

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

$\langle\psi|$ 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:

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

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. Priminsime, kad

$$2^{10} = 1024, \quad 2^{20} > 10^6, \quad 2^{30} > 10^9.$$

Antroji žinia (blogesnė):
Gauname tik šių reikšmių tiesinę kombinaciją.

Kvantinių algoritmų ribojimai. Operatoriai

Nagrinėkime tiesinį operatorių M:

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

Kvantiniuose algoritmuose leistini tik 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

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

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 yra realios ir $|\lambda_j| = 1$.

Svarbi tokių operatorių savybė, kad $UU = I$.

Kvantiniai vartai (gates)

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

NOT operatorius X:

$$X: \qquad |0\rangle \to |1\rangle = \beta_0^0 |0\rangle + \beta_1^0 |1\rangle,$$
$$|1\rangle \to |0\rangle = \beta_0^1 |0\rangle + \beta_1^1 |1\rangle,$$

matricinė forma $\langle k| X |j\rangle$, $k, j = 0, 1$:

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

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.

Kas galima, kas ne?

- Nežinomos kubito būsenos $|\psi\rangle = a|0\rangle + b|1\rangle$ negalima klonuoti.

  O teleportacija? Argi ji negalima?

Kas galima, kas ne?

- Nežinomos kubito būsenos $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ negalima klonuoti.

  O teleportacija? Argi ji negalima?

- Iš nežinomos kubito būsenos $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ galima gauti susietų n dalelių sistemą:

$$|\psi\rangle = a\,|00\ldots0\rangle + b\,|11\ldots1\rangle\,.$$

  Visos dalelės elgiasi vienodai pasirinktoje bazinių *ket*–vektorių sistemoje.

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:

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

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

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

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

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

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

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

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

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

Teleportacija

Alice turi kubitą $|\psi_0\rangle = a|0\rangle + b|1\rangle$, kurį nori persiųsti Bobui.
Alice ir pati nežino a, b reikšmių.

Teleportacija

Alice turi kubitą $|\psi_0\rangle = a\,|0\rangle + b\,|1\rangle$, kurį nori persiųsti Bobui. Alice ir pati nežino a, b reikšmių.

Paruošiame dar du kubitus, kurie yra susiję EPR sąryšiu:

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

Vieną kubitą atiduodame Alice, o kitą Bobui.

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!

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

Alice atlikus matavimą, Bobo kubitas irgi pasikeičia, jis projektuojamas į atitinkamą būseną.

Teleportacija įvyko!

Alice nusiunčia išmatuotą raktą Bobui (atviruoju kanalu).

Alice nusiunčia išmatuotą raktą Bobui (atviruoju kanalu).

Bobas transformuoja savo turimą kubitą į reikalingą būseną:

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

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 pradinį *ket*-vektorių

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

- Generuojame pradinį *ket*-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 :-)

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

Turime *ket*-vektorių

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

- Pakeičiame amplitudžių ženklus:

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

Turime *ket*-vektorių

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

- 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

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

- Tarkime, kad $P(x_k) = 1$, o visuose kituose taškuose $P(x_j) = 0$. Atlikus vieną iteraciją $\alpha_k \approx 3\alpha_j$, po antros iteracijos α_k dar padidėja kitų koeficientų atžvilgiu $\alpha_k \approx 5\alpha_j$.

- Tarkime, kad $P(x_k) = 1$, o visuose kituose taškuose $P(x_j) = 0$. Atlikus vieną iteraciją $\alpha_k \approx 3\alpha_j$, po antros iteracijos α_k dar padidėja kitų koeficientų atžvilgiu $\alpha_k \approx 5\alpha_j$.
- Atlikus $\frac{\pi}{8}\sqrt{N}$ iteracijų $\alpha_k^2 \approx \sum_j \alpha_j^2$, po $\frac{\pi}{4}\sqrt{N}$ iteracijų $\alpha_k^2 \approx 1 - \frac{1}{N}$.

- Tarkime, kad $P(x_k) = 1$, o visuose kituose taškuose $P(x_j) = 0$. Atlikus vieną iteraciją $\alpha_k \approx 3\alpha_j$, po antros iteracijos α_k dar padidėja kitų koeficientų atžvilgiu $\alpha_k \approx 5\alpha_j$.
- Atlikus $\frac{\pi}{8}\sqrt{N}$ iteracijų $\alpha_k^2 \approx \sum_j \alpha_j^2$, 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.

Ženklų pakeitimo transformacija

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

Ženklų pakeitimo transformacija

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

Atlikime transformaciją

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

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

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

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

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

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

Transformaciją D realizuojame taip:

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

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

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 $(\log N)^3$.

Ačiū už dėmesį!