7 Codeknacken, superschnelle Datensuche und sicheres Cloud-Computing

Werden Quantencomputer einmal allgegenwärtige Superrechner sein oder ein Nischenprodukt für einige Spezialanwendungen?

»Die Technik zog sich Siebenmeilenstiefel an. Das Bewusstsein hat normale Schrittlänge.«

Heinrich Wiesner, Schweizer Schriftsteller

Düsterer als das Magazin »Focus« kann man die Bedrohung »Quantencomputer« wohl kaum zeichnen: »US-Monsterrechner droht, die Welt ins Chaos zu stürzen«,[57] überschrieb das Blatt einen Artikel über die Absicht der NSA, einen Quantenrechner zu bauen. Himmelhoch jauchzend geht es aber auch, indem Journalisten, wie oft geschehen, vom »Wunderrechner« schreiben. In jedem Fall rückt mancher Journalist rechnende Quanten in die Nähe einer wahlweise zerstörerischen oder konstruktiven Allmacht.

Es geht aber auch in die andere Richtung. Mit Blick auf die bislang enttäuschende Quantentechnik der Firma D-Wave Systems ist oft von einem »Flop« die Rede.

Na, was denn nun? Kann einem denn niemand sagen, ob Quantencomputer die Welt umkrempeln oder eine Laborkuriosi-

tät bleiben werden? Genau das will ich in diesem Kapitel versuchen.

Es lässt sich bemäkeln, dass 20 Jahre nach den Entdeckungen des Shor- und des Grover-Algorithmus trotz intensiver Forschung kaum weitere »Partituren« für das Quantenorchester von vergleichbarer Schlagkraft komponiert wurden. Dies könnte ein Indiz dafür sein, dass die Grenzen des Quantencomputers sehr eng gesteckt sind. Doch Peter Shor selbst ist anderer Meinung, wie wir noch sehen werden.

Um einen »Wunderrechner« zu erfinden, muss man sich jedenfalls gewaltig ins Zeug legen, denn die Konkurrenz durch den klassischen Computer ist enorm. Das Konzept des Von-Neumann-Rechners kann im Prinzip alles, was *irgendeine Art von Computer* kann. Er ist eine Universalmaschine.

»Universell« bedeutet allerdings nicht, dass der klassische Rechner *alles* berechnen kann, er hat seine Grenzen, wie schon Alan Turing in den 1930er-Jahren bewies. Der britische Mathematiker fragte sich, was sich überhaupt berechnen lässt und was nicht. Vor dem so genannten Halteproblem muss jeder Computer kapitulieren, zeigte Turing. Demnach gibt es kein Computerprogramm, das ein anderes Programm daraufhin testet, ob dieses endlich oder unendlich lange braucht, bis es ein Ergebnis ausspuckt. Was akademisch klingt, begegnet vielen Menschen tagtäglich: abstürzende Software. Die Entwickler haben kein Testprogramm, das ihnen vorab sagt, ob neu entwickelte Software aufgrund eines Programmierfehlers in eine Endlosschleife geraten kann oder nicht. Das müssen die Nutzer leidvoll in der Praxis testen.

Einen Quantencomputer als »Wunderrechner« zu bezeichnen ist insofern irreführend, als er die gleichen Grenzen hat wie ein klassischer Computer. Zumindest, was die Art von Problemen angeht, die er lösen kann. Denn er ist eben auch nur ein Computer und kann nicht mehr können als eine universelle Maschine. Auch er wird am Halteproblem scheitern. Wer auf künftige Absturzsicherheit hofft, sei hiermit schon mal enttäuscht.

Es gibt allerdings Aufgaben, die normale Rechner zwar in einer endlich langen Zeitspanne *lösen* können, die aber *unakzeptabel* lang ist. Fünf Milliarden Jahre sind eben auch eine »endliche Zeitspanne«. Dies sind jene Probleme, die exponentiell mit ihrer Größe wachsen. Das Problem des Handlungsreisenden vervielfacht seine Komplexität mit jeder hinzukommenden Stadt. Und eine Zahl in ihre Primfaktoren zu zerlegen (was der RSA-Verschlüsselung zugrundeliegt) wird mit jeder zusätzlichen Dezimalstelle der Zahl um etwa den Faktor zehn zeitaufwendiger.

Um dieses Problems Herr zu werden, müsste man einen klassischen Rechner im gleichen Maß ausbauen, wofür man letztlich ganze Kontinente mit Prozessoren bedecken müsste. Diese Art Probleme sind also für klassische Rechner nicht handhabbar.

Der Quantencomputer unterscheidet sich wohltuend: Fügt man ein Qubit zu einem Quantenregister hinzu, passiert etwas Ähnliches wie beim Hinzufügen einer Stadt zum Problem des Handlungsreisenden: Die Zahl der parallel gespeicherten Werte verdoppelt sich. Die Rechenkapazität wächst somit genauso schnell wie die Komplexität des Problems, nämlich exponentiell. Der Quantencomputer sollte daher die hochkomplexen Aufgaben auch dann noch in wenigen Schritten lösen können, wenn ein normaler Rechner dafür Jahrtausende oder länger bräuchte.

So keimte nach Peter Shors Coup im Jahr 1994 die Hoffnung, der Quantencomputer würde *alle* diese aufsässigen Probleme handstreichartig niederstrecken. »Diese Hoffnung hielt ungefähr sechs Monate an«, kommentiert Seth Lloyd. Inzwischen ist eine deutliche Ernüchterung eingetreten.

Der Quantencomputer rechnet nicht *pauschal* schneller als ein normaler Rechner. »In der öffentlichen Wahrnehmung herrscht ein großes Missverständnis«, meint Scott Aaronson, der das Potenzial und die Grenzen von Quantencomputern erforscht. »Die Leute denken, dieser Rechner gehe durch eine Überlagerung aller möglichen Antworten und irgendwie schreie die korrekte Antwort laut über alle anderen hinweg, um sich bemerkbar zu machen«, sagt der Informatiker.

Im vorletzten Kapitel haben wir aber gelernt, dass vielmehr eine Orchestrierung nötig ist, um die richtige Antwort herauszufiltern. In Aaronsons Worten: »Die Antworten werden im Quantencomputer in Form von Wellen dargestellt und Wellen können sich gegenseitig auslöschen oder verstärken«, erklärt er. »Die Kunst ist, die Interferenz so zu steuern, dass sich die Wege, die zu den falschen Antworten führen, gegenseitig auslöschen und am Ende nur der Weg zur richtigen Antwort übrig bleibt.«

Ob das gelingt oder nicht, hängt von der Art des Problems ab. Es gibt sozusagen wohlorganisierte Profiorchester, bei denen sich der Dirigent leicht tut, den Musikern eine berührende Interpretation von Beethoven oder Tschaikowski zu entlocken. Das sind die für Quantencomputer lösbaren Aufgaben. Andererseits existieren Probleme, die eher einem frisch zusammengewürfelten Schülerorchester gleichen, aus dem selbst ein Daniel Barenboim keine sonderlich wohlklingende Symphonie herauskitzeln könnte.

Klingt abstrakt? O. K., sehen wir uns ein paar Beispiele für solche Orchestrierungen an. Fangen wir mit dem Grover-Algorithmus an, der in Datenbanken sucht.

Shiva und der Hütchenspieler

Stellen Sie sich vor, Ihr Smartphone zeigt Ihnen die unbekannte Nummer eines entgangenen Anrufs. Sie würden gerne wissen, wer da angerufen hat. Google schweigt sich aber darüber aus. Es bleibt nur das Telefonbuch. Wenn es sich um eine, sagen wir, Münchner Nummer handelt, haben Sie einen echten Scheißjob vor sich.

Denn eine bessere Methode, als Eintrag für Eintrag zu prüfen, gibt es nicht. Was dumm ist, wenn man sich entscheidet, von A bis Z Nummer für Nummer zu prüfen und sich schließlich herausstellt, dass die Person »Zwenger« heißt. Dann hat man eine Million Nummern gecheckt! Nur selten wird der Suchende dieses Pech haben, aber durchschnittlich fallen bei dieser mühseligen Methode immerhin 500.000 Prüfungen an.

Ein Quantencomputer hingegen würde, zumindest theoretisch, mit 1000 Rechenschritten auskommen, also 500 Mal weniger. Je größer die Datenbank ist, desto gewaltiger wird auch der Unterschied zwischen der Quantensuchmethode und dem herkömmlichen Abklappern. Bei zehn Milliarden Einträgen müsste ein klassischer Rechner im Schnitt fünf Milliarden davon prüfen, während ein Quantencomputer mit 100.000 Rechenschritten auskäme. Hier ist das Missverhältnis schon 50.000 : 1. Allgemeingültig ausgedrückt: Die Zahl der Anfragen steigt mit der klassischen Methode proportional zur Anzahl der Datenbankeinträge, mit dem Quantencomputer nur proportional zur Quadratwurzel dieser Anzahl.

Dass eine Google-Anfrage trotz riesiger Datenmengen nur Bruchteile von Sekunden dauert, liegt allerdings nicht etwa daran, dass im kalifornischen Mountainview schon Quantenserver stünden. Vielmehr gibt es ein mächtiges Hilfsmittel, um mit klassischen Suchmethoden schnell fündig zu werden: der so genannte Index. In unserem Szenario würde ein Index die Telefonnummern *in sortierter Reihenfolge* enthalten, sodass der Computer die Nummer schnell findet. Neben der Nummer stünde ein Verweis auf den Namen »Zwenger«.

Der Datenbankindex ähnelt dem Schlagwortregister am Ende eines Buchs.

Die Erstellung eines Index nimmt also bestimmte Suchanfragen vorweg, der Aufwand wird vorverlagert. Google liefert so fix, weil es schon zuvor Ressourcen in den Aufbau von Indizes gesteckt hat. Internet-Suchmaschinen erstellen laufend Indizes und verbrauchen damit immer mehr Energie und Speicherplatz.

Je mehr die Datenflut zu einem gewaltigen Daten-Tsunami anwächst, desto wünschenswerter wäre ein Computer, der auch in *unsortierten* Datenbanken schnell fündig wird, also ohne aufwendig erzeugte Indizes auskommt. Wie massiv die Datenberge anschwellen, sei hier nur anhand des größten Teilchenbeschleunigers der Welt, des Large Hadron Collider am europäischen Kernforschungszentrums CERN bei Genf, veranschaulicht. Dieser produ-

ziert jedes Jahr 15 Millionen Gigabytes an Messdaten, womit man mehrere Hunderttausend DVDs füllen könnte. Forscher durchsuchen die Daten nach hypothetischen und, falls existent, extrem seltenen Auffälligkeiten. Sie fahnden nach Abweichungen von der Regel, die auf neue Physik jenseits der Grenzen der bekannten Naturgesetze hinweisen.

Mit jeder neuen Datenquelle, jedem neuen Heuhaufen in der Welt, mit jeder darin versteckten Nadel wächst das Bedürfnis nach schnellen Suchmöglichkeiten.

Kann ein Quantencomputer die gewünschte Blitz-Suchmaschine liefern? Wir wissen: Er verarbeitet alle Lösungsmöglichkeiten parallel. Übertragen auf die Namenssuche: Er kann das Suchkriterium, die Telefonnummer, simultan mit allen Einträgen vergleichen. Der Quantenrechner ähnelt ein bisschen dem hinduistischen Gott Shiva mit seinen vielen Armen. Stellen wir uns vor, dass Shiva mit jedem seiner Arme in einer anderen Parallelwelt hantiert. Er könnte damit z.B. einen Hütchenspieler überlisten, indem er simultan unter alle drei Hütchen guckt. Doch jemandem, der sich Shiva zu Diensten machen will, bringt das nichts, solange der Gott ihm nicht mitteilt, unter welchem Hütchen die Nuss denn nun liegt.

Ein Quantencomputer ist ähnlich schweigsam. Das Quantenregister enthält zwar jede Menge Information, nämlich mit welchen Wahrscheinlichkeiten bei einer Messung welcher der vielen gespeicherten Werte herauskommt. Aber man kann eben immer nur *einen* dieser Werte auslesen, wobei die Information über die Wahrscheinlichkeit der anderen Messwerte verlorengeht.

Was fängt man aber mit einem Rechner an, der zwar »weiß«, unter welchem Hütchen das Ergebnis steckt, beim Messen aber *irgendein* Hütchen aufdeckt?

Man muss sich etwas einfallen lassen, um auf Anhieb das richtige Hütchen zu raten. Das gelingt, indem man die Wahrscheinlichkeiten geschickt manipuliert. Der indisch-amerikanische Physiker Lov Grover entwickelte 1996 an den Bell Labs in Murray Hill einen Algorithmus dafür, eben den Grover-Algorithmus. Dieser

zeigt auf einfache Weise, wie Quantencomputer einmal arbeiten sollen. Deshalb sehen wir uns das mal genauer an.

Stellen Sie sich ein Quantenregister mit zwei Qubits vor. Es speichert vier binäre Werte, nämlich 00, 01, 10 und 11.

Auf das Register wendet man das Hadamard-Gatter an, sodass eine Superposition der vier Zahlen entsteht. Jeder der Werte ist gleichberechtigt, das heißt, er würde bei einer Messung mit gleicher Wahrscheinlichkeit herauskommen wie die drei anderen, was in Abbildung 6–1 als gleich lange Striche dargestellt ist.

Das Ziel von Grovers Algorithmus ist es, die Messwahrscheinlichkeit für den richtigen Wert zu vergrößern und die aller anderen möglichst auf null zu reduzieren. Dann wird beim Auslesen der Richtige angezeigt.

Den gesuchten Wert findet man mit einem Suchfilter. Stellen wir uns diesen als eine Black Box vor, die die Phase um 180 Grad dreht, wenn sie den gesuchten Eintrag prüft, und bei falschen Einträgen nichts verändert. Das manipuliert die Superposition wie in Abbildung 6–1 b dargestellt. Der Witz ist, dass der Suchfilter alle vier Einträge gleichzeitig prüft, da diese in der Superposition ja parallel existieren. Es ist wie mit Shiva und dem Hütchenspieler. (Wie gesagt, kommt es uns hier nicht so sehr darauf an, wie die Suche selbst funktioniert, sondern wie der Quantencomputer uns das Suchergebnis mitteilt.)

Die Phase der Wellenfunktion des richtigen Werts, angenommen, es ist 01, wird also um 180 Grad gedreht, während alle anderen unangetastet bleiben. Allerdings ist damit noch nichts gewonnen. Die Operation ist vergleichbar mit einer Drehung des Pfeils in der Blochkugel aus Abbildung 3–7 entlang deren Äquator. An der Wahrscheinlichkeit, diesen Wert zu messen, ändert das erst einmal nichts. Nennen wir die Messwahrscheinlichkeit wie weiter oben »Amplitude«.

Doch Grover fand eine Operation, die auf einem Quantenregister durchführbar ist und das Problem im Handumdrehen löst: nämlich die einzelnen Amplituden am Mittelwert aller Amplituden zu spiegeln. Auch dies ist graphisch dargestellt (Abb. 6–1 c). Die

Spiegelung führt dazu, dass die Amplitude des Eintrags 01 auf den Wert 1 erhöht und die Amplituden der anderen Einträge jeweils auf den Wert 0 reduziert werden. Wenn jetzt gemessen wird, lautet das Ergebnis mit Sicherheit »B«. Bingo!

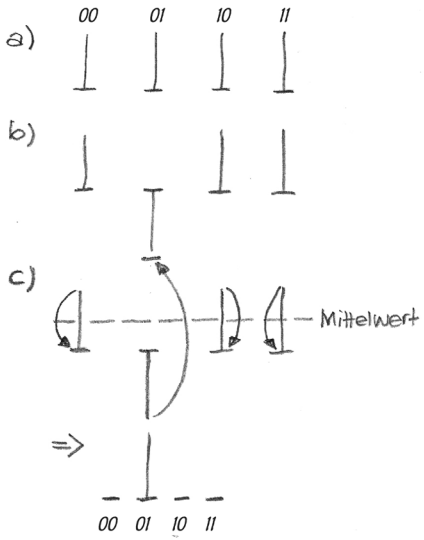

Abb. 7–1 Graphische Darstellung des Grover-Algorithmus

a: Am Anfang haben alle Einträge die gleiche Amplitude.

b: Drehen der Phase des gesuchten Eintrags

c: Spiegeln der Amplituden am Mittelwert aller vier Amplituden

In nur *zwei* Schritten kommt der Quantencomputer zu einem Ergebnis – was bei vier Einträgen allerdings auch klassisch gelingt. Erst bei deutlich mehr Datenbankeinträgen zeigt der Grover-Algorithmus seinen Geschwindigkeitsvorteil. Allerdings reichen zwei Operationen dann nicht mehr. Denn nach dem Drehen des richtigen Eintrags weicht der Mittelwert umso weniger von den Amplituden der restlichen Einträge ab, je mehr Einträge es gibt. Das Spiegeln am Mittelwert reduziert deren Amplituden also kaum, sodass sehr wahrscheinlich ein falscher Eintrag gemessen wird.

Wird die Prozedur aus Drehen und Spiegeln aber mehrmals wiederholt, schrumpfen die Amplituden der falschen Einträge bei jedem Schritt und die des richtigen wächst entsprechend, bis es praktisch sicher ist, das richtige Ergebnis zu messen. Eine einfache

Rechnung zeigt, dass die Anzahl der nötigen Wiederholungen der Wurzel aus der Anzahl der Einträge entspricht.

Auch der Quantencomputer kommt also nicht in einem Schritt zum Ergebnis. Man muss die Wahrscheinlichkeitswellen in seinem Innern so geschickt überlagern, dass die der falschen Ergebnisse ausgelöscht werden und die der richtigen verstärkt. Dafür muss eine gewisse Anzahl von Operationen ausgeführt werden.

Harmonien im Universum der Zahlen

Als zweites Beispiel betrachten wir Shors Algorithmus, bei dem die besondere Rolle der Verschränkung auf faszinierende Weise zutage tritt. Der Job lautet: Finde die Primfaktoren einer großen Zahl, also jene Teiler dieser Zahl, die sich nicht weiter zerlegen lassen. Bei der Primzahl 15 sind das die Teiler 5 und 3 und bei der Primzahl 21 lauten sie 7 und 3.

In Presseartikeln steht oft, Shors Verfahren probiere alle möglichen Teiler parallel aus. Das ist das Missverständnis, das Scott Aaronson anprangert. Der Quantencomputer kann das genauso wenig, wie ein Dirigent es schaffen würde, die kreischenden Möwen an einer Steilküste dazu zu bringen, Beethovens Neunte zu trällern. Vielmehr hat das Faktorisierungsproblem eine schwache Flanke, die ein Quantencomputer ausnutzen kann.

Die folgende Erklärung soll nur plausibel machen, wo diese Achillesferse liegt und wie ein Quantenrechner hineinstößt. Eine detaillierte mathematische Beschreibung findet sich zum Beispiel im ausgezeichneten Buch »Quantencomputing verstehen« von Matthias Homeister. Ich gebe zwei Erklärungen: eine, die nahe am mathematischen Algorithmus ist mit ein paar Formeln, und eine bildliche, die sich im Kasten »Der Wanderer und Quanten-Reinhold« findet. Wer mit Zahlen und Formeln nicht so gut kann, darf abkürzen, die zwei folgenden Absätze lesen und dann direkt zum Kasten springen.

Ein Kunstgriff, den Mathematiker gerne machen, um widerborstige Probleme zu lösen, besteht darin, die Aufgabe auf eine

andere zu »reduzieren«, die leichter zu knacken ist. Dazu braucht es aber irgendeine Art von mathematischer Struktur hinter dem Problem. Um es kurz zu machen: Eine solche gibt es bei der uns interessierenden Frage. Die Faktorisierung lässt sich auf das *Finden von Perioden* zurückführen. Was ist damit gemeint? Stellen Sie sich Außerirdische vor, die auf der Erde landen und Beobachtungen machen. Ziemlich schnell würden sie bemerken, dass es einen Tag-Nacht-Rhythmus von 24 Stunden gibt. Etwas länger würden sie benötigen, um die Periodizität der Mondphasen zu finden und noch länger, um die der Jahreszeiten festzustellen. Sie hätten durch Beobachtung mehrere Perioden gefunden.

Welche Art von Perioden sucht man nun beim Faktorisieren? Es geht um regelmäßige Widerholungen innerhalb von Zahlenfolgen. Dazu sehen wir uns eine spezielle Zahlenfolge an, nämlich die der Potenzen von zwei, also 2^1, 2^2, 2^3 … Hier ist jede Zahl das Doppelte der vorangegangenen.

2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, …

Nun teilen wir jede dieser Zahlen durch 15 und sehen uns jeweils den Rest dieser Division an. In der Mathematik bezeichnet man diese Operation als »Modulo«, geschrieben als

$2^x \bmod 15$.

Die Divisionsreste ergeben ebenfalls eine Folge:

2, 4, 8, 1, 2, 4, 8, 1, 2, 4, …

Wir sehen: Nach vier Zahlen wiederholt sich das Ganze. Die Folge ist periodisch. Machen wir das gleiche mit 21 als Teiler, ergibt sich eine Periode von sechs Zahlen:

2, 4, 8, 16, 11, 1, 2, 4, 8, 16, …

Was hat das jetzt mit der Primfaktorenzerlegung zu tun? Die Antwort gab das Mathematik-Genie Leonhard Euler schon im 18. Jahrhundert. Angenommen, die Zahl N hat die Primfaktoren p und q. Dann sehen wir uns diese Zahlenfolge an:

$x \bmod N$, $x^2 \bmod N$, $x^3 \bmod N$, $x^4 \bmod N$, …

Auch diese Folge wird eine Periodizität haben. Der Clou: die Periode ist ein Teiler von $(p\text{-}1)(q\text{-}1)$. Kurz gesagt: *Sie hängt mit den gesuchten Primfaktoren zusammen*. Nun kann man die Perioden für mehrere dieser Folgen ausrechnen (indem man x variiert) und

lernt so immer mehr über die Primfaktoren, bis man p und q selbst ermittelt hat.

Man kann das Faktorisierungsproblem also lösen, indem man Perioden bestimmt. Wenn N klein ist, geht das auch ohne Quantenrechner. Weil aber die Wiederholungen fast so lang werden können wie die Zahl selbst, bekommt man ein Problem, wenn N ein paar hundert Dezimalstellen hat wie bei RSA üblich. Einen normalen Computer überfordert das, denn er müsste mehr Divisionsreste ausrechnen, als es Partikel im All gibt. Er müsste quasi wie zuvor schon bei der Datenbanksuche eine Zahl nach der anderen abklappern, bis er feststellt, wo die Folge sich wiederholt. Um es in der obigen Metapher mit den Außerirdischen zu sagen: Mit einem normalen Rechner kann man den Tag-Nacht-Rhythmus erkennen, vielleicht gerade noch die Mondphasen, aber auf keinen Fall den Wechsel der Jahreszeiten.

Hätten wir aber einen Quantencomputer: Ha! Dann sähe die Welt völlig anders aus. Angenommen, wir wollen eine Bank knacken oder digitale Signaturen fälschen. Dazu müssen wir eine 600-stellige Zahl faktorisieren, nennen wir sie N.

Alles, was wir brauchen, sind zwei Quantenregister, von denen jedes N Werte speichert. Dafür braucht man insgesamt etwa 1000 Qubits. Die Sache bleibt also übersichtlich. Das eine Quantenregister speichert einfach die laufenden Nummern von 1 bis N. Aus diesen Werten lassen sich nach dem oben skizzierten Rezept eine Zahlenfolge und daraus die Folge der Divisionsreste berechnen. Der Quantencomputer macht das parallel mit allen Werten und schreibt die Ergebnisse in das zweite Register.

Und jetzt kommt der Witz. Die Ergebnisse der Berechnung sind mit ihren Ausgangswerten im ersten Quantenregister verschränkt. Sie merken sich sozusagen ihre Herkunft, ihre Eltern, wenn man so will. Nun braucht man nur noch ein paar Schritte zur Lösung.

Zunächst misst man das zweite Quantenregister. Dabei kommt *irgendeiner* der Reste heraus, was natürlich überhaupt nichts aussagt. *Aber*: Wegen der Verschränkung geschieht gleich-

zeitig auch etwas mit dem ersten Register. Angenommen, der gemessene Wert ist *X*. Wegen der Wiederholungen tritt *X* mehrmals im zweiten Register auf. Also hat *X* mehrere Eltern. Die Messung löscht nun die Amplituden aller Werte im ersten Register bis auf die der Eltern von *X*.

Die gesuchte Periode ist nun also im ersten Register enthalten, nämlich als Differenz zwischen je zwei benachbarten Werten. Diese Differenz lässt sich durch weitere Manipulationen und eine Messung am ersten Register gewinnen.

Der Shor-Algorithmus arbeitet sehr effizient. Im »Handumdrehen«, wie es oft heißt, wird aber auch ein Quantencomputer nicht die Primfaktoren einer großen Zahl finden. Die Frage ist letztlich: Wie stark wächst die Zeit, die er dafür braucht, mit der Größe der Zahl, die es zu zerlegen gilt? Die Sicherheit von RSA basiert darauf, dass man dem Fortschritt der Computertechnik leicht vorauseilen kann, indem man diese Zahl vergrößert. Der Igel wird immer schon längst da gewesen sein, wo der Hase gerade ankommt. Beim Quantencomputer wächst die Rechenzeit aber *nicht* exponentiell. Das bedeutet, dass eine Vergrößerung der Zahl zwar auch für den Quantencomputer mehr Rechenzeit bedeuten würde. Aber sie würde sich nicht potenzieren. Es bliebe bei Minuten, vielleicht eine Stunde, eine Zeit jedenfalls, die der Angreifer abwarten könnte.

Der Wanderer und Quanten-Reinhold

Ein fauler und sadistischer Landvermesser gibt einem Wanderer und einer Quanten-Version von Reinhold Messner folgende Aufgabe: Geht über diese Berglandschaft und sagt mir, wie weit die Gipfel auseinanderliegen. Außer einem Funkgerät, mit dem ihr mir das Ergebnis meldet, und einem Höhenmesser dürft ihr keine technischen Hilfsmittel verwenden!

→

Der Wanderer macht sich auf den Weg. Ihm ist klar, dass er keine andere Möglichkeit hat, als von Gipfel zu Gipfel zu wandern und die Schritte dazwischen zu zählen. Mit dem Höhenmesser stellt er fest, wann er jeweils den höchsten Punkt erreicht hat.

Nach gut einer Stunde kommt er am ersten Gipfel an. Er macht nur ein Wurstbrot lang Pause, stapft weiter und fängt mit dem Schrittezählen an. Bei Schritt Nummer 28644, nach weiteren drei Stunden, kommt er am zweiten Gipfel an. Zuvor hat er seine Schrittlänge gemessen, sie beträgt 50 Zentimeter. Völlig erledigt greift er zum Funkgerät und meldet dem Vermesser: »Die Gipfel liegen 28.644 Schritte auseinander, das sind 14 Kilometer und 322 Meter.«

Nun ist Quanten-Reinhold dran. Er greift den Höhenmesser und steht auf. »Willst du keinen Proviant mitnehmen, es ist ein langer Weg«, sagt der Aufgabensteller sarkastisch. »Ach was, brauch ich nicht!«, entgegnet Quanten-Reinhold. »Und das Funkgerät?« Quanten-Reinhold winkt mit einem spöttischen Lächeln im Gesicht ab.

Er macht den ersten Schritt in Richtung des ersten Gipfels. Aber was passiert da bloß? Der Landvermesser reibt sich die Augen. Quanten-Reinhold hat den ersten Schritt noch nicht auf den Boden gesetzt, da erscheint eine kilometerlange Kolonne von identischen Kopien Quanten-Reinholds, die sich auf den ersten Gipfel zieht. In der Ferne erkennt der Aufgabensteller vage, dass sich die Kolonne weiter durch das nächste Tal und zum zweiten Gipfel erstreckt.

Fast im gleichen Moment drehen sich die Quanten-Reinholds um und melden unisono, mit vielfachem, etwas überheblichem Grinsen: »14 Kilometer und 322 Meter.«

»Wie hast du … äh, ich meine, wie habt ihr das gemacht?«, fragt der Landvermesser. »Na, wie wohl«, antworten die Quanten-Reinholds. »Wir haben eine Höhenmessung vorgenommen. Dabei sind zufällig 250 Meter herausgekommen. Drei von uns stehen auf genau dieser Höhe. Weil wir miteinander verschränkt sind, kennen wir die Positionen der drei in unserer Kolonne. Sie sind jeweils 28.644 Schritte

→

bzw. 14 Kilometer und 322 Meter voneinander entfernt. Da die Landschaft regelmäßig geformt ist, entspricht dies dem Abstand der Gipfel.«

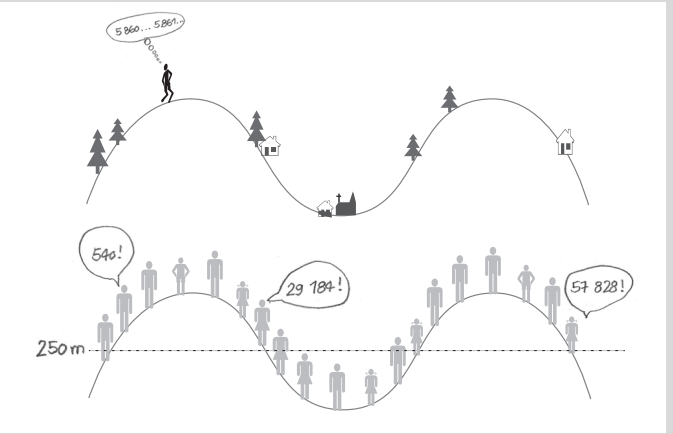

Abb. 7–2 a: Der Wanderer zählt die Schritte, um zu erfahren, wie weit die Gipfel auseinanderliegen.
b: Quanten-Reinhold ist an allen Orten gleichzeitig, das heißt, jeder Quanten-Reinhold steht für einen Schritt auf dem langen Weg. Er besetzt sehr viele unterschiedliche Höhen. Bei einer Messung kommt zufällig eine Höhe von 250 Metern heraus. Weil die Versionen von Quanten-Reinhold miteinander verschränkt sind, wissen sie, welche von ihnen bei der Höhe von 250 Meter stehen, und damit, welcher Schrittzahl diese Höhe entspricht. Damit wiederum lässt sich der Abstand berechnen.

Der Shor-Algorithmus funktioniert nur deshalb, weil sich hinter dem Faktorisierungsproblem eine mathematische Struktur verbirgt, nämlich eine Periode. Die Stärke des Quantencomputers ist es, diese im Zahlensalat gut verborgene Periode herauszufiltern.

Bemerkenswert dabei ist, dass die Lösung nicht eine der gespeicherten Zahlen ist wie zuvor beim Grover-Algorithmus, sondern eine »globale Eigenschaft« dieser Zahlen, wie Scott Aaronson