

# Kvantum algoritmusok

- Tanultuk, hogy egy kvantumregiszter mérésakor adott valószínűséggel beesik valamelyik bázisállapotába
- Hogyan használjuk akkor a kvantumszámítógépet egy **probléma megoldására**?
- Okosan megírt **kvantumalgoritmussal** el kell érni, hogy a probléma potenciális megoldását jelentő érték legyen a bázisokhoz hozzárendelve a kvantumbitek szuperpozíciójában
- Ha ekkor mérjük meg a rendszer állapotát, akkor nagy valószínűséggel olyan bázisállapotba esik, ami a probléma **megoldását reprezentálja**

# A Deutsch-Jozsa algoritmus

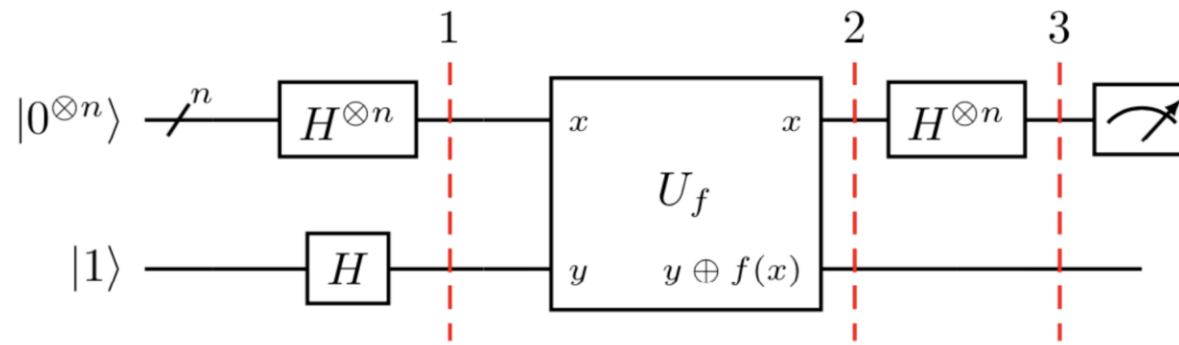
Tekintsünk egy  $f: \{0,1\}^n \rightarrow \{0,1\}$  függvényt

- $f$  lehet **kiegyensúlyozott**: a lehetséges bemenetek pontosan felére ad 0 értéket, a másik felére meg 1-et
- $f$  lehet **konstans**: az összes bemenetre 0-t vagy az összes bemenetre 1-et ad értékül

**A feladat:** döntsük el, hogy  $f$  konstans vagy kiegyensúlyozott

- Klasszikus módszerekkel, a legrosszabb esetben ehhez legalább  $2^{n-1} + 1$  függvény kiértékelés kell
- Kvantumszámítógéppel megoldható az  $f$  egyszeri kiértékelésével

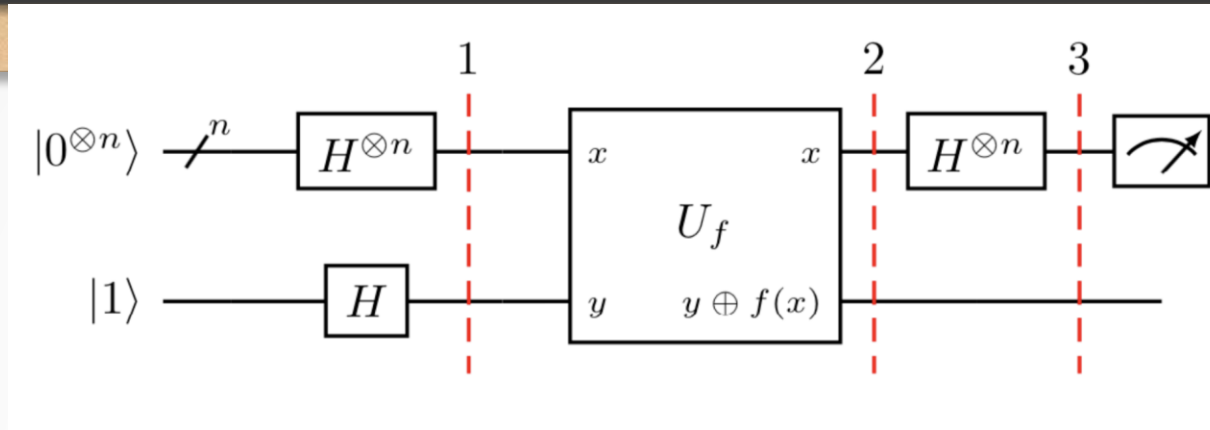
# A Deutsch-Jozsa algoritmus



1. Elkészítünk két kvantum regisztert: az egyik egy  $n$  kvantumbites, ahol minden bit a  $|0\rangle$  bázisban van, a másik 1 bites az  $|1\rangle$  bázisállapotban
2. Alkalmazzuk a Hadamard kaput mindegyik bitre
  - Azonos valószínűségű szuperpozíciót kapunk:

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle (|0\rangle - |1\rangle)$$

# A Deutsch-Jozsa algoritmus



3. Alkalmazzuk az  $U_f$  kvantumkaput, ami a bementeként kapott  $|x\rangle|y\rangle$  szuperpozícióra alkalmazza az  $f$  függvényt úgy, hogy a kapu kimenete az  $|x\rangle|y \oplus f(x)\rangle$  szuperpozíció

- A következő szuperpozíciót kapjuk:

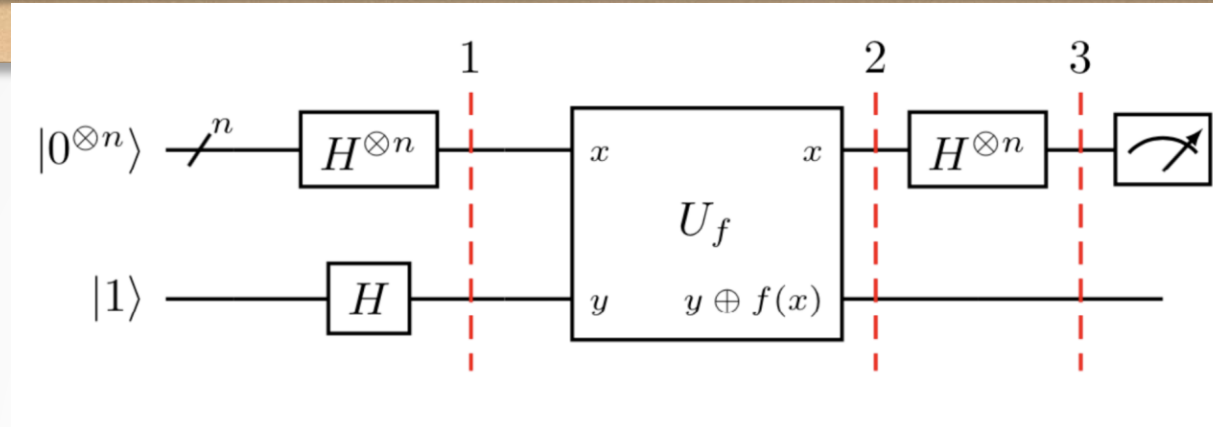
$$\frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle(|0\rangle \oplus f(x) - |1\rangle \oplus f(x)) = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle(|0\rangle - |1\rangle)$$

$\oplus$  a bináris összeadás  
művelete modulo 2

- Ezen a ponton az utolsó kubit nem érdekel minket, ami marad:

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

# A Deutsch-Jozsa algoritmus



4. Alkalmazzuk a Hadamard kaput az első regiszter kvantumbitjeire
5. A kapott szuperpozícióban annak valószínűsége, hogy a  $|00 \dots 0\rangle$  bázist mérjük:

$$\left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2 = \begin{cases} 1, & \text{ha } f(x) \text{ konstans} \\ 0, & \text{ha } f(x) \text{ kiegyensúlyozott} \end{cases}$$

# Kvantumszámítógép – Gyakorlati alkalmazás

- Egy gyakorlati alkalmazási terület: az **RSA feltörése**
- Az RSA (kódolási) eljárásról:
  - Központi eleme a **nyilvános és a privát kulcsok** generálása, ennek lépései
    - Két nagy prímszám  $p$  és  $q$  generálása (legalább 512 számjegyből állnak)
    - Legyen  $N = p \cdot q$
    - Legyen  $\varphi(N)$  az **Euler-függvény**
      - azaz  $\varphi(N)$  az  $N$ -nél nem nagyobb relatív prímek száma
      - Például:  $\varphi(15) = |\{1,2,4,7,8,11,13,14\}| = 8$

# Az RSA eljárás

- $\varphi(N)$  kiszámítható a  $p$  és  $q$  ismeretében:
  - $\varphi(N) = (p - 1) \cdot (q - 1)$
  - Például legyen  $p = 3$  és  $q = 5$ ,
  - Akkor  $p \cdot q = 15$  és láttuk, hogy  $\varphi(15) = 8$
  - Másrészt  $8 = 2 \cdot 4 = (p - 1)(q - 1)$
- Választani kell egy nyilvános és egy privát kulcsot a kódoláshoz / dekódoláshoz
- Legyen a **nyilvános kulcs**  $(N, e)$ , ahol  $1 \leq e < \varphi(N)$  egy  $\varphi(N)$ -nel relatív prím, jellemzően 65537
- Legyen a **privát kulcs**  $(N, d)$ , ahol  $d$  szám, az  $e$  multiplikatív inverze
  - azaz  $d \cdot e \equiv 1 \pmod{\varphi(N)}$

# Az RSA eljárás

- **Példa:**
  - Legyen  $p = 61$  és  $q = 53$ ,
  - Ekkor  $N = pq = 3233$  és  $\varphi(N) = (p - 1)(q - 1) = 3120$
  - Legyen  $e = 17$  és  $d = 2753$ ; ez helyes mert  $2753 \cdot 17 = 46801 \equiv 1 \pmod{3120}$
  - Nyilvános kulcs:  $N = 3233, e = 17$ 
    - $m = 123$  elkódolása:  $c = 123^{17} \pmod{3233} = 855$
  - Privát kulcs:  $N = 3233, d = 2753$ 
    - $c = 855$  dekódolása:  $m = 855^{2753} \pmod{3233} = 123$
- **Nehéz feltörni**, mert nehéz meghatározni egy (nagyon nagy) szám prímtényezőss felbontását
- **Peter Shor, 1994**: kvantumalgoritmus, ami képes a prímfaktorizációt hatékonyan elvégezni

123018668453011775513049495838496272  
077285356959533479219732245215172640  
050726365751874520219978646938995647  
494277406384592519255732630345373154  
826850791702612214291346167042921431  
160222124047927473779408066535141959  
7459856902143413  
felbontásához (768bit) 2009-ben  
**2000 év gépidő kellett!**

# Shor algoritmus

A FAKTORIZÁCIÓ visszavezethető a **periodicitás** meghatározására:

- Legyen  $N = p \cdot q$
- Válasszunk **véletlenszerűen** egy  $a < N$  számot
- Ha  $a$  **nem osztja**  $N$ -t, számoljuk ki az alábbi sorozat **periodicitását**:

$$a^1 \bmod N$$

$$a^2 \bmod N$$

$$a^3 \bmod N$$

...

$$a^N \bmod N$$

- Ha az  $r$  periodicitásra bizonyos feltételek teljesülnek, akkor

- $p = \text{lnko}(a^{\frac{r}{2}} - 1, N)$  és

- $q = \text{lnko}(a^{\frac{r}{2}} + 1, N)$

A periodicitást **hatékonyan** ki lehet számítani kvantumszámítógéppel

Példa:

$$N = 15 \quad a = 7$$

- $7^1 \bmod 15 = 7$

- $7^2 \bmod 15 = 4$

- $7^3 \bmod 15 = 13$

- $7^4 \bmod 15 = 1$

- $7^5 \bmod 15 = 7$

- ...

$$r = 4$$

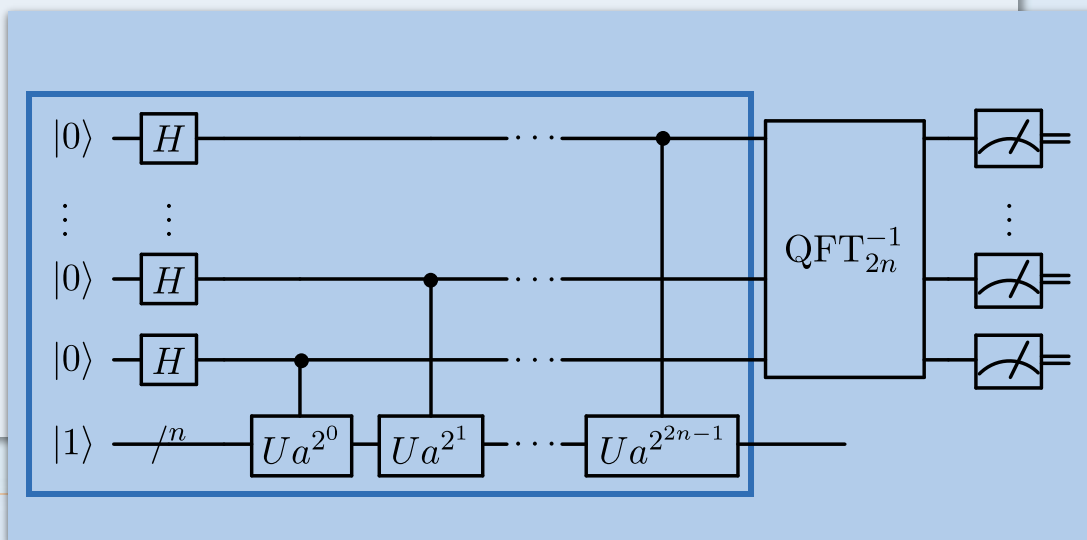
$$p = \text{lnko}(48, 15) = 3$$

$$q = \text{lnko}(50, 15) = 5$$

# Shor algoritmus

- Hozzunk létre egy **kvantum regiszter** párt a kitevő-maradék párokkal és állítsuk bázisok egyenlő szuperpozíciójába:

$$|1, a^1 \bmod N\rangle + |2, a^2 \bmod N\rangle + |3, a^3 \bmod N\rangle + \dots + |N, a^N \bmod N\rangle$$

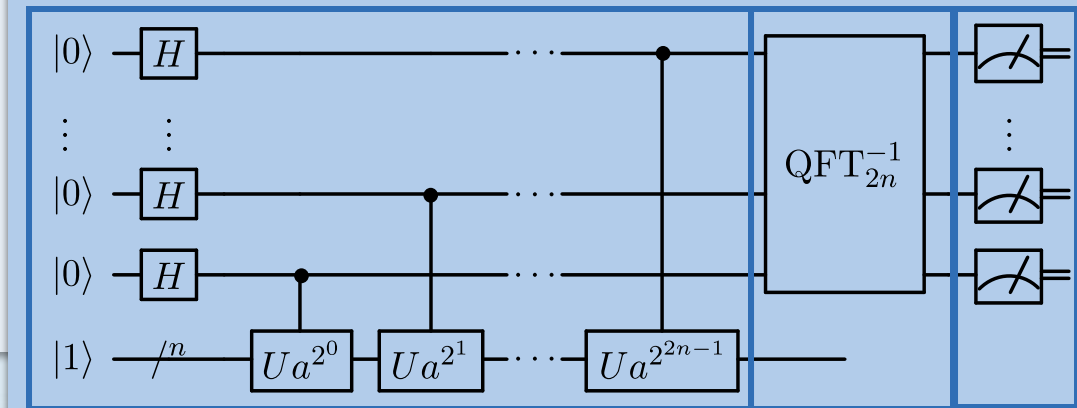


# Shor algoritmus

- Hozzunk létre egy **kvantum regiszter** párt a kitevő-maradék párokkal és állítsuk bázisok egyenlő szuperpozíciójába:

$$|1,7\rangle + |2,4\rangle + |3,13\rangle + |4,1\rangle + \dots$$

- Inverz kvantum Fourier transzformációval** megnöveljük annak a bázisnak a valószínűségét, ami a periodicitáshoz tartozik
- Méréskor** a rendszer nagy valószínűséggel a periodicitást tároló **bázis állapotba esik**
- A periodicitás segítségével **törhető** a kódolás



# Kvantumszámítás – konklúzió

Megoldhatók-e az **NP-teljes** problémák hatékonyan kvantumszámítógéppel?

- Az a sejtés, hogy nem

Kell-e félni attól, hogy kvantumszámítógéppel **feltörjük** a titkos adatainkat?

Egyelőre nem:

- AZ RSA-ban használt 2048 bites számok faktorizálásához becslések szerint **több ezer kvantumbit** kellene
- Jelenleg **256 kvantumbit** a maximum
- Közben új, **kvantum-rezisztens** kriptográfiai eljárások készülhetnek

S. Ebadi & al. Quantum phases of matter on a 256-atom programmable quantum simulator, Nature 595, 2021. július

# Kvantumszámítás – State of the Art

- 2001: 7 kubit, Shor algoritmus 15 prímfelbontására
  - 2019: A Google bejelenti, hogy elérte a kvantum-fölényt
    - 53 kubit és egy random generátor: kvantum vs klasszikus = 200 sec vs 10000 év
    - IBM: az a 10000 év inkább 2.5 nap
  - 2020 december: USTC kínai egyetem
    - 76 kubit és a „Boson sampling”: kvantum vs klasszikus = 20 sec vs 600 millió év
  - Újabb kvantumalgoritmusok megadása egy intenzíven kutatott terület
- Kapcsolat a klasszikus bonyolultsági osztályokkal (Wikipedia):

