La gracia de Shor → Busca el P de una función con la QFT

→ El resto es como un algoritmo clásico

Tiene 2 partes → Cuántico
→ Clásico

QFT → $\Theta(n^2)$ FFT → $O(n \cdot 2^n)$

Hay una mejora exponencial

Para factorizar un nº Shor lo que hace es buscar el periodo P de una función t.q:

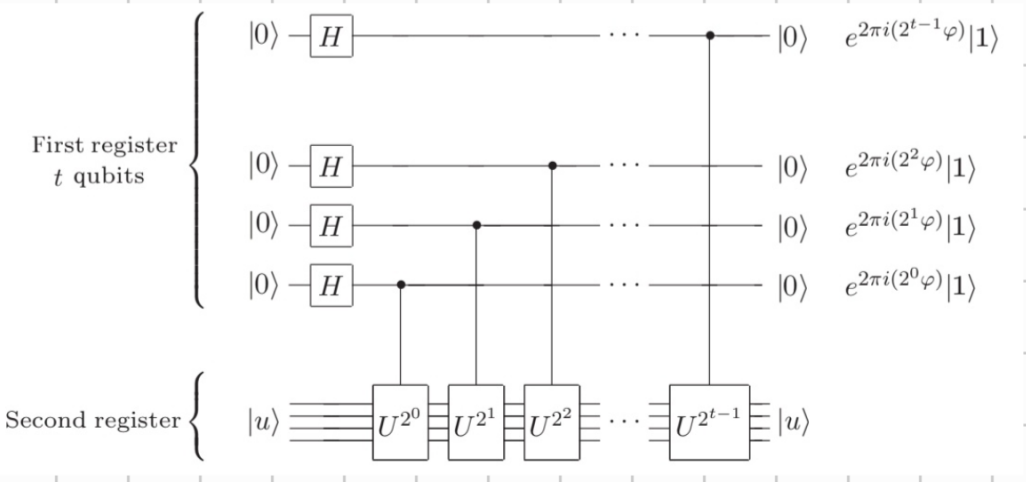
Estimar una fase

Sea $f: \{0, \dots, N-1\} \to \{0, \dots, N-1\}$ una f con periodo w

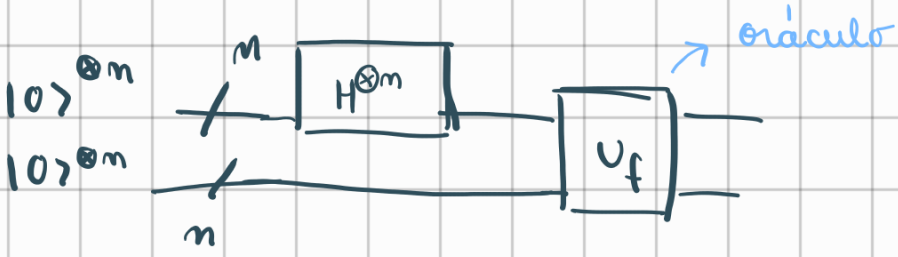
($f(x+w) = f(x) \quad \forall x$)

Tenemos n qubits, necesitamos $2^n = N$ compuertas.
Para hacer la estimación necesitamos:

- 1 oráculo que nos prepare el e.i $|v\rangle$ y efectúe las operaciones unitarias
- 2 registros → 1: contiene n qubits en $|0\rangle$
  → 2: Arranca en $|v\rangle$ y los bits necesarios para $|v\rangle$



1er paso Construir el e.i $\sum_{x=0}^{N-1} |x\rangle |f(x)\rangle$ usando



2do paso Aplicar QFT a $|x_0 + r\rangle$ con $x_0 | f(x_0) = f(x)$

3er paso Medir en la base computacional con prob.

$$\text{prob}(x) = \frac{A}{N} \left| \frac{1}{A} \sum_{j=0}^{A-1} e^{\frac{2\pi i j r y}{N}} \right|^2$$

Se favorece $\frac{yr}{N} \in \mathbb{N}$. $\frac{y}{N}$ es una fracción irreducible

$$\frac{a}{b} = \frac{c}{d} \quad \text{con} \quad \frac{c}{d}$$

Algoritmo de Shor

(1) Elegí $a < N$ al azar.

(2) $\gcd(a, N)$. Si es $\neq 1 \Rightarrow$ factorizaste N. terminaste.
Si no, seguir

(3) Calcular el período r de $f(x) = a^x \bmod N$.

$a^r = a^0 = 1 \bmod N$ porque $f(0) = f(r)$

Si r es par, $a^{r/2}$ cumple las hipótesis del teorema (X)

Si r es impar, (4)

(4) Si r es impar ó $a^{r/2} \equiv -1 \bmod N$. volvemos a (1)

(5) Por teorema, $\gcd(a^{r/2} \pm 1, N)$ son posibles factores de N.

Teorema

Si N es un número compuesto (no primo) de L bits y x es una solución no trivial de la ecuación $x^2 \equiv 1(\mod N)$ en el rango $1 \leq x \leq N$ que, además, no es ni $x \equiv 1(\mod N)$ ni $x \equiv N-1 \equiv -1(\mod N)$, entonces o $gcd(x-1, N)$ o $gcd(x+1, N)$ es un factor no trivial de N. Dicho factor puede encontrarse en $O(L)$ operaciones.