80-quantum-bernstein-vazirani-algorithm-03

Quantum Computing

Algoritmo di Bernstein-Vazirani

1

1

problema della stringa nascosta

- è dato un algoritmo che calcola $f: \{0,1\}^n \rightarrow \{0,1\}$
- l'algoritmo è implementato in una *black box*
  - non si può ispezionarne l'implementazione
  - un circuito che *non può essere aperto*
- è noto solo che $f(x) = u \cdot x$ per una qualche stringa di n bit *nascosta* (non nota) $u \in \{0,1\}^n$
  - vogliamo scoprire u
- per farlo possiamo usare l'implementazione di $f(x)$ sottoponendole diversi input
  - ci piacerebbe fare il numero minimo di tentativi

2

2

80-quantum-bernstein-vazirani-algorithm-03

problema di parità

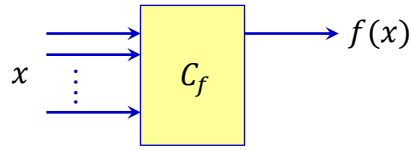
- è dato un algoritmo che calcola $f: \{0,1\}^n \rightarrow \{0,1\}$
- è noto che $f(x) = u \cdot x$ per una qualche stringa di n bit *nascosta* (non nota) $u \in \{0,1\}^n$
- l'operatore $\cdot$ fa la moltiplicazione bit per bit e poi fa l'*xor* dei bit risultato (fa la somma modulo 2)
  - in pratica u stabilisce quali sono i bit da usare per calcolare la parità di x
- es: supponiamo $n = 3$, $u = 101$, $x = x_1x_2x_3$
- abbiamo $f(x) = u \cdot x = x_1 \oplus x_3 = x_1 + x_3 (\text{mod } 2)$

3

3

un modo classico di porre il problema

- algoritmo che calcola $f: \{0,1\}^n \rightarrow \{0,1\}$, con $f(x) = u \cdot x$
- un'implementazione classica di f è una box tipo



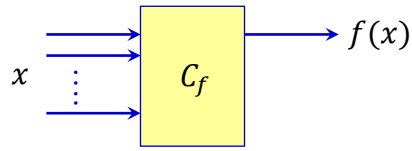
4

4

80-quantum-bernstein-vazirani-algorithm-03

un approccio esaustivo

- algoritmo che calcola $f:\{0,1\}^n \to \{0,1\}$, con $f(x) = u \cdot x$
- un'implementazione classica di f è una box tipo



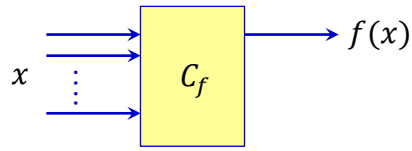
- per scoprire u proviamo con i seguenti input:
  - con $10\cdots0$ otteniamo in output u_1
  - con $01\cdots0$ otteniamo in output u_2 ….
- in pratica per scoprire u facciamo n step

5

5

nel calcolo classico si può fare meglio?

- algoritmo che calcola $f:\{0,1\}^n \to \{0,1\}$, con $f(x) = u \cdot x$
- un'implementazione classica di f è una box tipo



- nel calcolo classico non possiamo fare meglio
  - in ogni invocazione dell'algoritmo possiamo ottenere solo un bit di informazione e u ha n bit

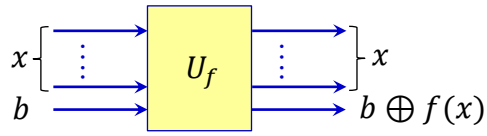
6

6

80-quantum-bernstein-vazirani-algorithm-03

l'approccio quantum computing

- algoritmo che calcola $f:\{0,1\}^n \rightarrow \{0,1\}$, con $f(x) = u \cdot x$
- consideriamo un'implementazione quantum di f



- l'algoritmo di Bernstein-Vazirani riesce a scoprire u con *una sola* invocazione di U_f

7

7

gli step dell'algoritmo

1. costruzione di una superposition $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle$
2. uso dell'operatore di Hadamard per ottenere u

8

8

supponiamo di aver eseguito lo step 1

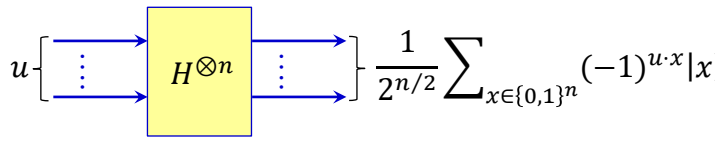
1. costruzione di una superposition
   $\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle$
2. uso dell'operatore di Hadamard per ottenere u

- nello step 1 costruiamo una superposition
  $\frac{1}{2^{n/2}}\sum_{x}(-1)^{f(x)}|x\rangle = \frac{1}{2^{n/2}}\sum_{x}(-1)^{u\cdot x}|x\rangle$
- osservazione: questa superposition è quella che si ottiene applicando l'operatore di Hadamard a u

9

9

applicazione di Hadamard a u



- applicando Hadamard a u si ottiene proprio lo stato con il quale termina lo step 1
- ma Hadamard è reversibile e l'inverso di Hadamard è ancora Hadamard
- quindi se riesco ad eseguire lo step 1, poi applico Hadamard allo stato ottenuto e poi misuro, ottengo u

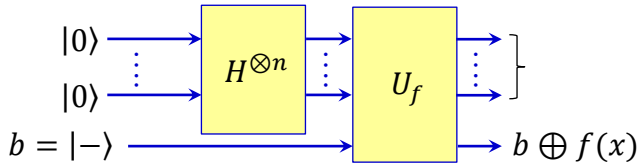
10

10

80-quantum-bernstein-vazirani-algorithm-03

come realizzare lo step 1?

1. costruzione di una superposition
   $\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle$
- usiamo questo circuito



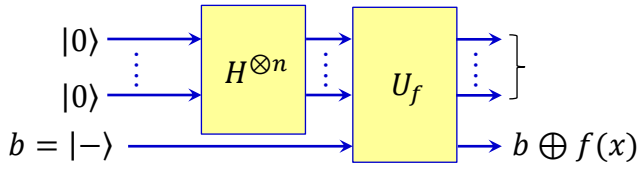
- in ingresso inseriamo n bit nello stato $|0\rangle$
- diamo al bit b il valore $|-\rangle$

11

11

un paio di richiami

1. costruzione di una superposition
   $\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle$
- usiamo questo circuito



- ricordiamo che $H^{\otimes n}|0\ldots0\rangle = \frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}|x\rangle$
- e che $|-\rangle = {}^{1}/_{\sqrt{2}}|0\rangle - {}^{1}/_{\sqrt{2}}|1\rangle$

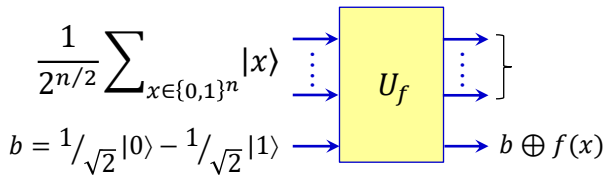
12

12

80-quantum-bernstein-vazirani-algorithm-03

osserviamo U_f

1. costruzione di una superposition
   $\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle$



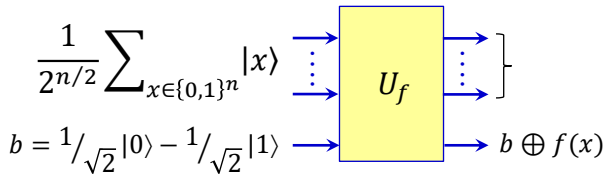
- se $f(x) = 0$, allora $b \oplus f(x) = |-\rangle$
- se $f(x) = 1$, allora $b \oplus f(x) = -|-\rangle = -{}^1/_{\sqrt{2}}|0\rangle + {}^1/_{\sqrt{2}}|1\rangle$

13

13

osserviamo U_f

1. costruzione di una superposition
   $\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle$



- quindi $b \oplus f(x) = (-1)^{f(x)}|-\rangle$

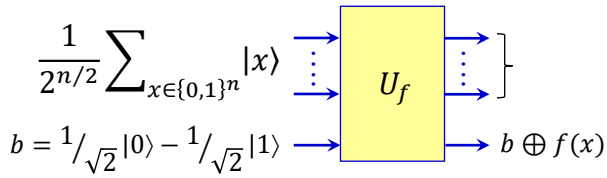
14

14

80-quantum-bernstein-vazirani-algorithm-03

osserviamo U_f

1. costruzione di una superposition
   $\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle$



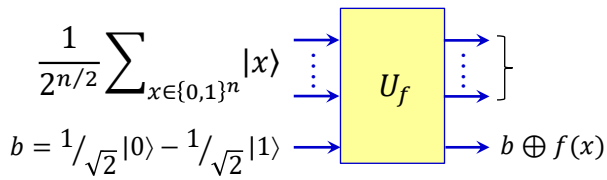
- l'input di U_f è complessivamente $\frac{1}{2^{n/2}}\sum_x |x\rangle\,|-\rangle$
- l'output è $\frac{1}{2^{n/2}}\sum_x(-1)^{f(x)}|x\rangle\,|-\rangle$
  - infatti quando $f(x) = 1$ otteniamo un segno $-$

15

15

osserviamo U_f

1. costruzione di una superposition
   $\frac{1}{2^{n/2}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle$



- l'output è $\frac{1}{2^{n/2}}\sum_x(-1)^{f(x)}|x\rangle\,|-\rangle$
- possiamo *dimenticarci* il $|-\rangle$ e fornire l'output a Hadamard

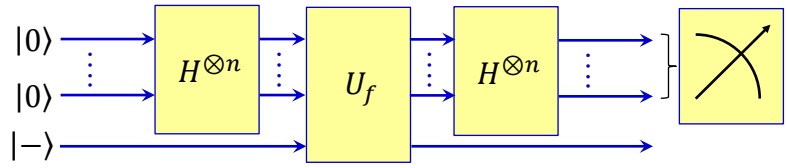
16

16

80-quantum-bernstein-vazirani-algorithm-03

l'algoritmo

- complessivamente abbiamo



17

17

intuizione algoritmica

- usiamo il potere del quantum computing per generare tutte le possibili soluzioni e un'idea algoritmica per *cancellare* quelle sbagliate

18

18

80-quantum-bernstein-vazirani-algorithm-03

esempio di esecuzione dell'algoritmo

- supponiamo $n = 2$ e $u = 01$
- eseguiamo Hadamard su $|00\rangle$ e otteniamo $|+\,+\rangle$ che è uguale a $\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$
- diamo in input a U_f lo stato $|+\,+\rangle\,|-\rangle$
- ora eseguiamo U_f e otteniamo $(\frac{1}{2}|00\rangle + \frac{1}{2}(-1)|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}(-1)\,|11\rangle)|-\rangle$

19

19

esempio di esecuzione dell'algoritmo

- ora eseguiamo Hadamard su $\frac{1}{2}|00\rangle + \frac{1}{2}(-1)|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}(-1)\,|11\rangle$
- otteniamo:

$$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}\begin{pmatrix} 1/2 \\ -1/2 \\ 1/2 \\ -1/2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

20

20

80-quantum-bernstein-vazirani-algorithm-03

esempio di esecuzione dell'algoritmo

- osserva ancora come il segno meno ottenuto relativamente al bit risultato quando $f(x) = 1$ sia usato per avere come effetto $(-1)^{f(x)}$
- nota come la misura dia il risultato corretto in modo deterministico e non probabilistico
  - è una caratteristica dell'algoritmo di Bernstein-Vazirani, non è una caratteristica generale

21

21

un po' di scetticismo

1. è vero che abbiamo fatto in tempo costante ciò che nel calcolo classico richiede tempo lineare, ma abbiamo usato un numero lineare di porte logiche
2. svelare il contenuto di una black box non è un problema così interessante

22

22