130-quantum-ricerca-di-un-elemento-in-un-array-02

Quantum Computing

Ricerca di un elemento in un array non ordinato

Introduzione all'Algoritmo di Grover

1

1

il problema della ricerca in un array

- è dato un array, non ordinato, di N elementi, indicizzati da 0 a $N-1$
- vogliamo trovare la posizione in cui è memorizzato uno specifico valore
- tutti gli elementi dell'array hanno un valore diverso dall'elemento cercato, tranne uno, ma non si sa quale sia
- in un certo istante è consentito accedere ad un elemento, ad es. all'elemento in posizione x e leggerne il valore

2

2

130-quantum-ricerca-di-un-elemento-in-un-array-02

l'approccio classico

- possiamo pensare di accedere, ad uno ad uno, a tutti gli elementi dell'array
  - nel caso peggiore dobbiamo fare N accessi
- oppure possiamo provare ad accedere a posizioni random, fino a quando troviamo l'elemento cercato
  - il valore atteso del numero di tentativi random è $\frac{N}{2}$

3

3

quantum

- è possibile fare di meglio usando il quantum computing?
- sì, è possibile usare l'algoritmo di Grover

4

4

130-quantum-ricerca-di-un-elemento-in-un-array-02

perché il problema è importante

- c'è una classe di problemi in informatica che è quella dei problemi NP-completi
- questi problemi sono importanti non solo in informatica ma anche in fisica, chimica, ecc.
- cercare una soluzione per un problema NP-completo può essere visto come un problema di ricerca

5

5

problemi NP-completi

- un problema NP-completo ha due caratteristiche distintive
  - se qualcuno propone una soluzione per il problema, si può verificare che quella sia effettivamente una soluzione in modo efficiente
  - si pensa che per trovare una soluzione sia necessario un tempo esponenziale

6

6

130-quantum-ricerca-di-un-elemento-in-un-array-02

probemi NP-completi

- il problema SAT è un tipico problema NP-completo
- un'istanza del problema SAT consiste in una formula booleana in forma normale congiuntiva
  - es. $(x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee x_4)$
- una soluzione è una scelta dei valori delle variabili per la quale la formula è vera
  - es. $x_1 = true$, $x_2 = false$, $x_3 = false$ e $x_4 = true$
- se le variabili della formula sono n, allora ci sono 2^n possibili scelte $true/false$, oppure $0/1$

7

7

perché il problema è importante

- possiamo pensare che i 2^n possibili valori delle variabili di SAT siano in corrispondenza biunivoca con gli $N = 2^n$ elementi dell'array nel quale facciamo la ricerca e che uno sia in corrispondenza con la eventuale soluzione

8

8

130-quantum-ricerca-di-un-elemento-in-un-array-02

l'algoritmo di Grover

- l'algoritmo quantum di Grover risolve il problema della ricerca in tempo $O(\sqrt{N})$
- come conseguenza l'algoritmo risolve il problema SAT in tempo $O(2^{n/2})$

9

9

formalizzazione del problema

- è data una funzione $f: \{0, \cdots, N-1\} \rightarrow \{0,1\}$, trovare x tale che $f(x) = 1$
- il caso peggiore è quello in cui c'è solo una x tale che $f(x) = 1$

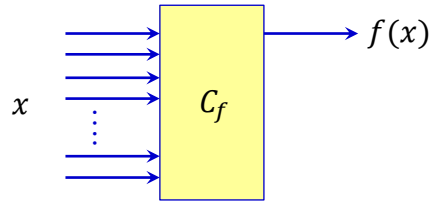
10

10

130-quantum-ricerca-di-un-elemento-in-un-array-02

in che modo è data la funzione?

- la funzione è data come un circuito, per cui l'atteggiamento classico è quello di fornire in input al circuito diversi valori di x e verificare l'output



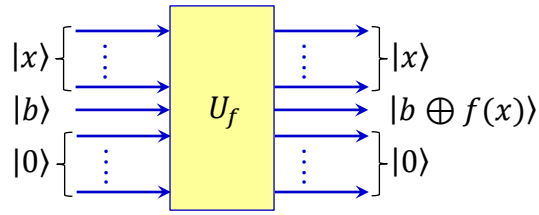
- possiamo pensare che il circuito sia quello di accesso alla memoria di un computer

11

11

in che modo è data la funzione?

- come sappiamo, l'equivalente quantum è un circuito del tipo



12

12