

# 22. Quantum Searching


Aldea Alexia, Razvan Dumitriu, Tudor Haulica, Andrei Murica,  
Sergiu Stanciu

# Grover's Algorithm: Accelerating Search with Quantum Computing


În lumea calculatoarelor clasice, găsirea unei informații dintr-un spațiu de căutare mare necesită examinarea acestui spațiu de căutare de un număr de ori proportional cu mărimea acestuia. De exemplu, dacă avem un spațiu de căutare de dimensiunea  $O(N)$ , vom avea nevoie să examinăm acest spațiu de aproximativ  $\sqrt{N}$  ori pentru a găsi informația căutată. Cu toate acestea, cuantumul poate face mult mai mult. Se pare că putem folosi un computer cuantic pentru a rezolva problema căutării după ce am examinat spațiul de căutare de aproximativ  $\pi\sqrt{N}/4$  ori! Această descoperire a fost făcută posibilă datorită algoritmului lui Grover.

Algoritmul lui Grover, numit după fizicianul american Lov Grover, este o metodă cuantică eficientă pentru căutarea într-o bază de date nesortată. Deși scopul său este adesea descris ca "căutarea într-o bază de date", este posibil să fie mai exact să-l descriem ca "inversarea unei funcții". Aproximativ vorbind, dacă avem o funcție  $y = f(x)$  care poate fi evaluată pe un computer cuantic, algoritmul lui Grover ne permite să calculăm  $x$  atunci când avem  $y$ . Inversarea unei funcții este legată de căutarea într-o bază de date deoarece am putea să creăm o funcție care produce o valoare particulară a lui  $y$  dacă  $x$  se potrivește cu o intrare dorită într-o bază de date și o altă valoare a lui  $y$  pentru alte valori ale lui  $x$ .

Pentru a înțelege mai bine algoritmul lui Grover, să explorăm un exemplu simplu. Să presupunem că avem o bază de date cu o cheie secretă de 12 cifre, iar scopul nostru este să găsim această cheie. Pe un computer clasic, ar trebui să examinăm în medie jumătate din toate cheile posibile pentru a găsi cheia corectă. În cel mai rău caz, ar trebui să examinăm toate cele  $10^{12}$  chei posibile. Cu toate acestea, algoritmul lui Grover poate găsi cheia cu mult mai puține încercări. Concret, în loc să examineze spațiul de căutare de  $10^{12}$  ori, algoritmul lui Grover va avea nevoie doar de aproximativ  $\pi\sqrt{10^{12}}/4$  ori. Acest factor de radical face o diferență semnificativă. Dacă  $N$  ar fi un trilion, așa cum este cazul în exemplul nostru, atunci un computer clasic ar trebui să examineze spațiul de căutare de un trilion de ori, în timp ce computerul cuantic va avea nevoie de mai puțin de 800 de mii de ori. Acest lucru reprezintă o îmbunătățire de peste un milion de ori.



Algoritmul lui Grover nu este doar o simplă curiozitate teoretică; are aplicații practice în domenii precum criptografia și optimizarea căutării. De exemplu, poate fi utilizat pentru căutarea rapidă a cheilor în criptografia cuantică sau pentru găsirea optimă a soluțiilor în problemele de optimizare combinatorică. În plus, algoritmul lui Grover reprezintă un exemplu important al puterii de calcul a calculatoarelor cuantice și a modului în care acestea pot revoluționa domeniul calculului și al informaticii în viitor.






## Prezentarea Codului:

Codul prezentat demonstrează procesul de creare, simulare și vizualizare a unui circuit cuantic utilizând Qiskit, un cadru popular de calcul cuantic open-source.

1. Importuri și Inițializare: Codul începe prin importul modulelor necesare din Qiskit și Qiskit Aer (un component al Qiskit pentru simulare cuantică). Apoi se inițializează un registru cuantic cu 3 qubituri și un registru clasic cu 3 biți pentru înregistrarea măsurărilor.

2. Construcția Circuitului: Circuitul cuantic este construit cu registrele cuantice și clasice specificate. Scopul este de a căuta două stări cuantice specifice,  $|101\rangle$  și  $|110\rangle$ , folosind o combinație de porți cuantice.

### 3. Aplicarea Porților Cuantice:

- Porți Hadamard: Aplicate tuturor qubiturilor pentru a-i pune într-o stare de superpoziție.
  - Porți Controlled-Z: Folosite pentru a marca stările  $|101\rangle$  și  $|110\rangle$  ca rezultate dorite.
  - Inversarea În Jurul Mediei (IAA): O secvență de porți care inversează faza stării  $|101\rangle$  și  $|110\rangle$  pentru a crește probabilitatea de a fi măsurate.
- 

4. Măsurare: Qubiturile sunt măsurate și rezultatele sunt stocate în biți clasici.
5. Simulare: Circuitul este simulat folosind simulatorul QASM din Qiskit Aer. Circuitul este transpilat pentru simulator, executat și rezultatele sunt obținute.
6. Vizualizare: Rezultatele sunt vizualizate folosind un histogramă, care arată distribuția probabilistică a stărilor măsurate.
7. Configurarea Contului IBMQ: Codul include linii comentate pentru configurarea unui cont IBM Quantum, care ar fi necesar pentru rularea circuitului pe hardware cuantic real.
8. Execuție pe Hardware Cuantic Real: Secțiunea comentată la sfârșit demonstrează cum să transpunem și să executăm circuitul pe un computer cuantic real (IBM Quantum Experience). Acesta implică încărcarea contului, obținerea unei infrastructuri, transpilarea circuitului pentru infrastructură și executarea acestuia. Rezultatele sunt apoi vizualizate în același mod ca și anterior.

