Κεφάλαιο 6

Κβαντικοί Αλγόριθμοι

|1⟩ *Ο κάθε ανόητος μπορεί να ξέρει. Το θέμα είναι να καταλαβαίνει.*
Albert Einstein

Η ζωή είναι ένα πεδίο συνείδησης, που εκφράζεται με εκατομμύρια τρόπους στο χωροχρόνο μέσω της κβαντικής διεμπλοκής
Amit Ray

6.1 Εισαγωγή

Το 1981 στο Massachusetts Institute of Technology (MIT), ο Richard Faynman διατύπωσε την πρόταση ότι οι κλασικοί υπολογιστές δεν μπορούν να προσομοιώσουν την εξέλιξη των κβαντικών συστημάτων με αποτελεσματικό τρόπο και πρότεινε ένα μοντέλο κβαντικού υπολογιστή που θα ήταν ικανός για τέτοιες προσομοιώσεις. Και με αυτό κατάφερε να περιγράψει ένα φανταστικό κβαντικό υπολογιστή, ο οποίος θα ξεπερνούσε τους κλασικούς επιτυγχάνοντας εκθετική αύξηση της υπολογιστικής ισχύος. Βέβαια, χρειάστηκαν αρκετά χρόνια, και για την ακρίβεια το 1994 όταν ο Peter Shor με τον περίφημο αλγόριθμο παραγοντοποίησης ενός φυσικού αριθμού σε γινόμενο πρώτων παραγόντων απέδειξε ότι αυτό πράγματι είναι δυνατόν. Θεωρητικά, ο αλγόριθμος του Shor είναι ικανός να σπάσει πολλά από τα κρυπτοσυστήματα που χρησιμοποιούνται σήμερα σε ώρες, αντί για εκατομμύρια χρόνια με τη χρήση κβαντι-

κών υπολογιστών και αυτό σηματοδότησε το μεγάλο πλέον ενδιαφέρον των ερευνητών για την Κβαντική Υπολογιστική και τις εφαρμογές της. Και όλα αυτά χωρίς να υπάρχει πραγματικός κβαντικός υπολογιστής. Όλα ήταν θεωρητικά! Τι θα συνέβαινε εάν.... Όλοι συνηγορούσαν ότι όταν θα είχαμε τέτοιους υπολογιστές ο κόσμος μας θα άλλαζε ριζικά. Όλες οι επιστήμες θα επωφελούνταν από αυτή την τρομακτική αύξηση της υπολογιστικής ταχύτητας. Από τη Χημεία μέχρι την Οικονομική επιστήμη και από τη Γεωλογία μέχρι τη Φαρμακευτική, όλα θα άλλαζαν προς όφελος της Ανθρωπότητας. Τελικά, ο πρώτος κβαντικός υπολογιστής με δύο qubits εμφανίστηκε το 1998, ενώ το 2017 η IBM παρουσίασε τον πρώτο κβαντικό υπολογιστή διαθέσιμο σε όλους!

Από τότε που ο Shor παρουσίασε την εργασία του, πολλοί κβαντικοί αλγόριθμοι επινοήθηκαν, κάποιοι από αυτούς χωρίς πραγματική εφαρμογή, αλλά έδωσαν το έναυσμα σε πολλούς επιστήμονες και ερευνητές να προβληματιστούν και να ασχοληθούν σοβαρά πλέον με το νέο αυτό φαινόμενο της πληροφορικής τεχνολογίας που δείχνει επαναστατικό: την Κβαντική Υπολογιστική.

Στο παρόν Κεφάλαιο θα ασχοληθούμε με κάποιους από τους πρώτους αλλά βασικούς κβαντικούς αλγόριθμους όπως είναι αυτοί των Deutsch-Jozsa, των Bernstein-Vazirani, του Simon και του Grover. Τον αλγόριθμο του Shor θα τον δούμε σε ξεχωριστό Κεφάλαιο όχι τόσο για την σπουδαιότητά του, αλλά για να συνδυαστεί με την κβαντική κρυπτογραφία.

6.2 Ο αλγόριθμος Deutsch - Jozsa

Ο αλγόριθμος Deutch-Jozsa είναι ένας κβαντικός αλγόριθμος ο οποίος προτάθηκε το 1992 από τους David Deutch και τον Richard Jozsa. Ο σκοπός του αλγόριθμου είναι να αποδείξει την υπεροχή ενός κβαντικού υπολογιστή έναντι ενός κλασικού υπολογιστή. Με άλλα λόγια, μια τεχνική που δύναται να εφαρμοστεί σε έναν κβαντικό υπολογιστή, είναι αδύνατο να εφαρμοστεί σε έναν κλασικό υπολογιστή.

Ορισμός του προβλήματος (Ο αλγόριθμος των Deutsch-Jozsa): Θεωρούμε μια συνάρτηση Boolean $f(x_0, x_1, \ldots, x_{n-1}) \to 0$ ή 1 με την ιδιότητα να είναι είτε *σταθερή* είτε *ισορροπημένη*.

- Σταθερή: Όλες οι τιμές της συνάρτησης είναι 0 ή 1,
- Ισορροπημένη: Οι μισές τιμές της συνάρτησης είναι 0 και οι άλλες μισές τιμές της είναι 1.

6.2.1 Ο αλγόριθμος Deutsch

Μία απλούστευση αυτής της συνάρτησης αποτελεί η boolean συνάρτηση $f\{0,1\} \to \{0,1\}$, (Αλγόριθμος Deutsch). Δηλαδή, η συνάρτηση είναι:

- Σταθερή: $f(0) = f(1) = 0$ ή $f(0) = f(1) = 1$,

- Ισορροπημένη: ($f(0) = 0$ και $f(1) = 1$) ή ($f(0) = 1$ και $f(1) = 0$).

Σε έναν κλασικό υπολογιστή, απαιτούνται δύο βήματα για την επίλυσή του (αρχικά υπολογίζονται οι τιμές $f(0)$ και $f(1)$ και στη συνέχεια συγκρίνονται τα αποτελέσματα). Ο αντίστοιχος κώδικας στην γλώσσα προγραμματισμού Python είναι ο ακόλουθος (Πρόγραμμα 6.1):

```
if function(0)== 0:
  if function(1) == 0:
    print("Σταθερή")
  else:
    print("Ισορροπημένη")
else:
  if function(1) == 0:
    print("Ισορροπημένη")
  else:
    print("Σταθερή")
```

Πρόγραμμα 6.1: Αλγόριθμος Deutsch: κλασική προσέγγιση.

Αντιθέτως, ο κβαντικός αλγόριθμος απαιτεί μόνο ένα βήμα. Υπενθυμίζουμε στην άλγεβρα Boole τον τρόπο που πραγματοποιείται η πρόσθεση modulo 2 και που συμβολίζεται με $\oplus$. Ισχύει $0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0$, και αυτή η πράξη μπορεί να προσομοιωθεί με την κβαντική πύλη cX όπου:

$$cX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

γιατί για παράδειγμα:

$$1 \oplus 1 \sim cX\,|11\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle$$

όπου το δεύτερο qubit είναι πλέον στην κατάσταση 0 όσο και το αποτέλεσμα της πράξης $1 \oplus 1 = 0$.

Είναι προφανές ότι $f(0) = f(1)$ είναι το ίδιο με το άθροισμα $f(0) \oplus f(1)$. Αν λοιπόν το αποτέλεσμα είναι 0, η συνάρτηση είναι σταθερή, ενώ στην αντίθετη περίπτωση η συνάρτηση είναι ισορροπημένη. Το εξαγόμενο βρέθηκε με έναν και μόνο υπολογισμό! Στη συνέχεια ετοιμάζονται οι κβαντικοί καταχωρητές του κυκλώματος. Αρχικά, ορίζουμε την κατάσταση των δύο απαιτούμενων qubits σε $|01\rangle$ όπως παριστάνεται παρακάτω (προσοχή, επειδή η αρχική κατάσταση των qubits είναι πάντα $|0\rangle$ πρέπει να εφαρμόσουμε την πύλη X για να το φέρουμε στην κατάσταση $|1\rangle$):

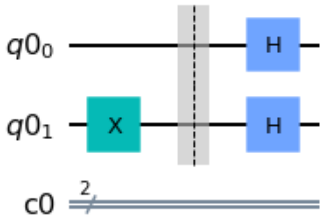
$$|01\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

Στη συνέχεια, εφαρμόζουμε την πύλη Hadamard και στα δύο qubits, δηλαδή $H \otimes H$. Αρχικά λοιπόν υπολογίζουμε το τανυστικό γινόμενο:

$$H \otimes H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix} \otimes \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\1 & -1 & 1 & -1\\1 & 1 & -1 & -1\\1 & -1 & -1 & 1\end{pmatrix}$$

και τότε το κύκλωμα γίνεται $(H \otimes H)\,|01\rangle$ όπως μαθηματικά περιγράφεται με την Σχέση 6.1 και σχηματικά αναπαριστάται στο Σχήμα **6.1**:

$$\begin{aligned}(H \otimes H)\,|01\rangle =& \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\1 & -1 & 1 & -1\\1 & 1 & -1 & -1\\1 & -1 & -1 & 1\end{pmatrix} \times \begin{pmatrix}0\\1\\0\\0\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix}\\ =& \frac{1}{2}\left(\begin{pmatrix}1\\0\\0\\0\end{pmatrix} - \begin{pmatrix}0\\1\\0\\0\end{pmatrix} + \begin{pmatrix}0\\0\\1\\0\end{pmatrix} - \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\right)\\ =& \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle\\ =& \frac{1}{2}(|0\rangle\,|0\rangle - |0\rangle\,|1\rangle + |1\rangle\,|0\rangle - |1\rangle\,|1\rangle)\end{aligned} \tag{6.1}$$



Σχήμα 6.1 Αλγόριθμος Deutsch: αρχική κατάσταση κυκλώματος.

Έτσι, δοθείσας της συνάρτησης f με την απεικόνιση $|x\rangle\,|y\rangle \to |x\rangle\,|f(x) \oplus y\rangle$ την εφαρμόζουμε στην Σχέση 6.1. Έτσι έχουμε:

$$\frac{1}{2}(|0\rangle|f(0)\oplus 0\rangle - |0\rangle|f(0)\oplus 1\rangle + |1\rangle|f(1)\oplus 0\rangle - |1\rangle|f(1)\oplus 1\rangle) \tag{6.2}$$

Τώρα, μπορούμε να διακρίνουμε 4 περιπτώσεις για την συνάρτηση f και αφού τις αντικαταστήσουμε στην σχέση 6.2 τα αποτελέσματα είναι :

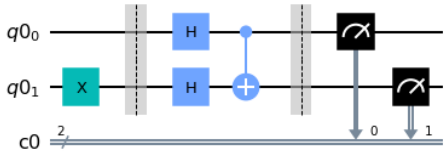
1 **Σταθερή**, $f(0)=0$ και $f(1)=0$: $\frac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle$

2 **Σταθερή**, $f(0)=1$ και $f(1)=1$: $\frac{1}{2}(|0\rangle|1\rangle - |0\rangle|0\rangle + |1\rangle|1\rangle - |1\rangle|0\rangle = \frac{1}{2}(-|0\rangle|0\rangle + |0\rangle|1\rangle - |1\rangle|0\rangle + |1\rangle|1\rangle$

3 **Ισορροπημένη**, $f(0)=0$ και $f(1)=1$: $\frac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle + |1\rangle|1\rangle - |1\rangle|0\rangle = \frac{1}{2}(|0\rangle|0\rangle - |0\rangle|1\rangle - |1\rangle|0\rangle + |1\rangle|1\rangle$

4 **Ισορροπημένη**, $f(0)=1$ και $f(1)=0$: $\frac{1}{2}(|0\rangle|1\rangle - |0\rangle|0\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle = \frac{1}{2}(-|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle$

Έτσι στην περίπτωση της *σταθερής* συνάρτησης το κύκλωμα παραμένει ως έχει, ενώ στην περίπτωση της *ισορροπημένης* μπορούμε να εφαρμόσουμε μια ελεγχόμενη not (cX) πύλη όπως φαίνεται στο Σχήμα **6.2**.



Σχήμα 6.2 Αλγόριθμος Deutsch: τελική κατάσταση κυκλώματος.

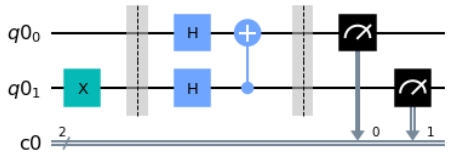
Στο τελευταίο βήμα εφαρμόζουμε πάλι την πύλη Hadamard στο πρώτο qubit (και ασφαλώς την πύλη αδράνειας - Identity στο δεύτερο). Συνεπώς, για την πρώτη περίπτωση (*σταθερή*) έχουμε:

$$H \otimes I = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \times \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\end{pmatrix}$$

Έτσι, εφαρμόζοντας τις πύλες στο προηγούμενο κύκλωμα έχουμε:

$$=\frac{1}{\sqrt{2}}\begin{pmatrix}1&0&1&0\\0&1&0&1\\1&0&-1&0\\0&1&0&-1\end{pmatrix}\times\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix}=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\\0\\0\end{pmatrix}=\frac{1}{\sqrt{2}}(\begin{pmatrix}1\\0\\0\\0\end{pmatrix}-\begin{pmatrix}0\\1\\0\\0\end{pmatrix})$$
$$=\frac{1}{\sqrt{2}}|00\rangle-\frac{1}{\sqrt{2}}|01\rangle=|0\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

Η τελική μορφή του κβαντικού κυκλώματος φαίνεται στο Σχήμα 6.3 (ισορροπημένη περίπτωση, και χωρίς την πύλη cX να είναι για τη σταθερή περίπτωση).



Σχήμα 6.3 Αλγόριθμος Deutsch: ισορροπημένη περίπτωση.

Έτσι, εφαρμόζοντας τις κατάλληλες πύλες και για τις τέσσερις περιπτώσεις μπορούμε να συνοψίσουμε:

- **Σταθερή**, $f(0)=0$ and $f(1)=0$: $|0\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$
- **Σταθερή**, $f(0)=1$ and $f(1)=1$: $|0\rangle\frac{|1\rangle-|0\rangle}{\sqrt{2}}$
- **Ισορροπημένη**, $f(0)=0$ and $f(1)=1$: $|1\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$
- **Ισορροπημένη**, $f(0)=1$ and $f(1)=0$: $|1\rangle\frac{|1\rangle-|0\rangle}{\sqrt{2}}$

Δηλαδή αρκεί να παρατηρήσουμε την κατάσταση του πρώτου qubit και να αποφανθούμε άμεσα για το είδος της συνάρτησης. Ο αντίστοιχος κώδικας σε Python είναι όπως φαίνεται στο Πρόγραμμα 6.2:

```
import random
def function(x):
    return random.randint(0,1)
def oracle(circuit, a, b):
    if a != b:
        circuit.cx(0,1)
from qiskit import *
from qiskit.tools.visualization import plot_histogram
```

```
9  qr = QuantumRegister(2)
10 cr = ClassicalRegister(1)
11 circuit = QuantumCircuit(qr,cr)
12 circuit.x(1)
13 circuit.h(0)
14 circuit.h(1)
15 x=0
16 y = function(x)
17 oracle(circuit, x, y)
18 circuit.h(0)
19 circuit.measure(0,0)
20 backend = BasicAer.get_backend('qasm_simulator')
21 shots = 1024
22 results = execute(circuit, backend=backend, shots=shots).result()
23 simulator_counts = results.get_counts()
24 circuit.draw(output='latex')
25 plot_histogram(simulator_counts, color='blue')
```

Πρόγραμμα 6.2: Αλγόριθμος Deutsch.

Σχόλια επί του Προγράμματος 6.2.

- Γραμμές 2, 3: ορισμός της συνάρτησης παραγωγής τυχαίας τιμής 0 ή 1,
- Γραμμές 4, 5 και 6: ορισμός της συνάρτησης η οποία εφαρμόζει τον αλγόριθμο του Deutsch. Εάν εντοπίσει ότι η τιμή της συνάρτησης και η παράμετρος της συνάρτησης είναι διαφορετικά εφαρμόζει πύλη ελεγχόμενου not (cX). Η ονομασία *oracle*, που θα την χρησιμοποιήσουμε πολλές φορές, αποτελεί κάτι σαν ένα μαύρο κουτί, το οποίο δέχεται σαν είσοδο το κβαντικό κύκλωμα και επιστρέφει την σωστή ενέργεια για την επίλυση του προβλήματος.
- Γραμμή 12: επειδή το επιπλέον qubit πρέπει να είναι σε κατάσταση $|1\rangle$ αλλά πάντα η αρχική κατάσταση είναι $|0\rangle$ εφαρμόζουμε μία πύλη not ώστε να αλλάξει την κατάστασή του σε $|1\rangle$,
- Γραμμή 15: έστω ότι το όρισμα της συνάρτησης είναι 0,
- Γραμμή 16: επιστροφή του αποτελέσματος της συνάρτησης,
- Γραμμή 21: επιλογή να εκτελεστεί το κύκλωμα 1024 φορές.

Συμπερασματικά, μπορούμε να δούμε ότι στην περίπτωση *σταθερή* το πρώτο qubit είναι πάντοτε 0, ενώ στην περίπτωση *ισορροπημένη* το πρώτο qubit είναι πάντοτε 1. Συνεπώς, *απαιτείται η μέτρηση του ενός μόνο qubit*! Αυτό είναι αδύνατο να επιτευχθεί σε έναν κλασικό υπολογιστή. Ενώ λοιπόν αυτό μπορεί να θεωρηθεί ένα ευχάριστο παιχνίδι και κάποιος θα μπορούσε να αναρωτηθεί «και ποιον ενδιαφέρει αυτό» , είναι μια από τις αποδείξεις ότι σε έναν κβαντικό υπολογιστή υπάρχει εκθετική επιτάχυνση σε σχέση με έναν κλασικό υπολογιστή.

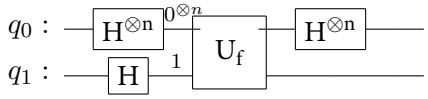
6.2.2 Ο αλγόριθμος των Deutsch-Jozsa

Ο αλγόριθμος Deutch-Jozsa είναι μια επέκταση του αλγορίθμου Deutch όπου υπάρχει μια συνάρτηση Boole $f(x_0, x_1, \ldots, x_{n-1}) \to 0$ ή 1 με την ιδιότητα να είναι είτε *σταθερή* είτε *ισορροπημένη*.

- Σταθερή: όλες οι τιμές της συνάρτησης είναι 0 ή 1,
- Ισορροπημένη: Οι μισές τιμές της συνάρτησης είναι 0 και οι άλλες μισές τιμές της είναι 1.

Η ιδιότητα αυτή της συνάρτησης (promise problem ή function), υποχρεώνει την τιμή εισόδου της συνάρτησης να ανήκει σε ένα συγκεκριμένο υποσύνολο από όλες τις πιθανές τιμές εισόδου. Έτσι σε αυτή την περίπτωση, η συνάρτηση πρέπει να είναι είτε σταθερή είτε ισορροπημένη.

Έτσι λοιπόν, εφαρμόζεται η ίδια τεχνική με αυτή που εφαρμόζεται στον αλγόριθμο Deutch (Σχήμα 6.4).



Σχήμα 6.4 Κύκλωμα αλγόριθμου Deutsch-Jozsa.

Το σχήμα με το σύμβολο U_f, (Σχήμα 6.4) ονομάζεται oracle ή μαύρο κουτί (black box). Στην κβαντική υπολογιστική, ένα oracle παριστάνει ένα μαύρο κουτί στο οποίο εισάγουμε δεδομένα και εξάγονται δεδομένα, αλλά δε γνωρίζουμε ποια ακριβώς είναι η εσωτερική του λειτουργία. Επιστρέφοντας στο πρόβλημά μας, θα δούμε ορισμένα παραδείγματα γιια να ξεκαθαρίσουμε και τους όρους που χρησιμοποιούνται. Για παράδειγμα το $H \otimes H = H^{\otimes 2}$ είναι ως εξής :

$$H \otimes H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \otimes \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{pmatrix}$$

ενώ το $H \otimes H \otimes H \otimes H \otimes H = H^{\otimes 4}$ είναι:

$$H^{\otimes 4} = \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{pmatrix} \otimes \frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{pmatrix} = \frac{1}{4}\begin{pmatrix}1 & 1 & \cdots & 1\\ 1 & -1 & \cdots & -1\\ \vdots & \vdots & \ddots & \vdots\\ 1 & -1 & \cdots & 1\end{pmatrix}$$

Συνεπώς, εφαρμόζοντας την ίδια τεχνική και εκτιμώντας μόνο την κατάσταση του πρώτου qubit, αν η συνάρτηση είναι σταθερή θα πάρουμε $|0\rangle$, διαφορετικά η συνάρτηση είναι ισορροπημένη.

Ας ξεκινήσουμε με κάποιες βοηθητικές σημειώσεις, καθώς ο υπολογισμός τέτοιων πινάκων μεγάλης τάξης είναι αρκετά δύσκολος. Όπως έχουμε δει, εφαρμογή δύο πυλών Hadamard έχει ως αποτέλεσμα :

$$H \otimes H \, |00\rangle = H^{\otimes 2} \, |0\rangle^{\otimes 2} = \frac{1}{\sqrt{2^2}} \sum_{x \in \{0,1\}^2} |x\rangle$$

όπου $x \in \{0,1\}^2$ αντιστοιχίζεται στις καταστάσεις $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$ και γενικεύοντας:

$$H^{\otimes n} \, |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \tag{6.3}$$

οπότε εάν $n = 4$ έχουμε:

$$H^{\otimes 4} \, |0\rangle^{\otimes 4} = \frac{1}{\sqrt{2^4}} \sum_{x \in \{0,1\}^4} |x\rangle = \frac{1}{4}(|0000\rangle + |0001\rangle + \cdots + |1111\rangle)$$

όπου $\langle x \cdot z \rangle$ αναπαριστά το εσωτερικό γινόμενο μεταξύ των x και z. Για παράδειγμα, εάν $x = \{110\}$ τότε:

$$H^{\otimes 3} \, |110\rangle = \frac{1}{\sqrt{2^3}} \sum_{z \in \{0,1\}^3} (-1)^{\langle x \cdot z \rangle} \, |z\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

και περισσότερο αναλυτικά,

$$\begin{aligned} H^{\otimes 3} \, |110\rangle &= \frac{1}{8}((-1)^{\langle 110 \cdot 000 \rangle} \, |000\rangle + (-1)^{\langle 110 \cdot 001 \rangle} \, |001\rangle + \cdots + (-1)^{\langle 110 \cdot 011 \rangle} \, |111\rangle) \\ &= \frac{1}{8}((-1)^0 \, |000\rangle + (-1)^0 \, |001\rangle + (-1)^1 \, |010\rangle + \cdots + (-1)^2 \, |111\rangle) \\ &= \frac{1}{8}(|000\rangle + |001\rangle - |010\rangle - |011\rangle - |100\rangle - |101\rangle + |110\rangle + |111\rangle) \end{aligned} \tag{6.4}$$

Τα σύμβολα $+$ και $-$ είναι εύκολο να υπολογιστούν. Ας θεωρήσουμε ένα παράδειγμα: ποιο θα είναι το σύμβολο της κατάστασης $|001\rangle$. Θα είναι το δυαδικό εσωτερικό γινόμενο των $|110\rangle$ και $|001\rangle = 1 \times 0 + 1 \times 0 + 0 \times 1 = 0$ και έτσι $(-1)^0 = 1$, οπότε το $|001\rangle$ θα έχει θετικό πρόσημο.

Τώρα λοιπόν, έχουμε τα απαραίτητα εργαλεία για να προχωρήσουμε στον αλγόριθμο. Στο πρώτο βήμα του αλγόριθμου για μια n bit ακολουθία, χρειάζονται $n+1$ qubits και το τελευταίο είναι το qubit ελέγχου. Η αρχική κατάσταση πρέπει να είναι $|\psi_0\rangle = |0\rangle^{\otimes n} \otimes |1\rangle$ και στη συνέχεια εφαρμόζουμε την πύλη Hadamard σε όλα τα qubits. Χρησιμοποιώντας τη σχέση 6.3, έχουμε:

$$H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

και

$$H |1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Έτσι, εφαρμόζοντας την πύλη Hadamard σε όλα τα qubits, το κύκλωμα θα μετατραπεί σε :

$$\psi_0 = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2^{n+1}}} \sum_{x\in\{0,1\}^n} |x\rangle (|0\rangle - |1\rangle)$$

Το επόμενο βήμα είναι να εφαρμόσουμε τη συνάρτηση oracle με την απεικόνιση $|x\rangle |y\rangle \to |x\rangle |f(x) \oplus y\rangle$ και παίρνουμε:

$$\begin{aligned}\psi_1 =& \frac{1}{\sqrt{2^{n+1}}} \sum_{x\in\{0,1\}^n} |x\rangle (|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) \\ =& \frac{1}{\sqrt{2^{n+1}}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle)\end{aligned}$$

Στη συνέχεια, το πραγματικά τελικό βήμα είναι να εφαρμόσουμε ξανά την πύλη Hadamard σε όλα τα qubits και εφόσον δεν μας ενδιαφέρει το qubit ελέγχου, ας δούμε τι συμβαίνει στα υπόλοιπα. Ας θυμηθούμε τη σχέση **??**:

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{\langle x\cdot z\rangle} |z\rangle$$

κι έτσι εφαρμόζοντας την πύλη Hadamard στο κύκλωμα έχουμε:

$$\begin{aligned}|\psi_2\rangle &= \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} (-1)^{\langle x\cdot z\rangle} (\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle) \\ &= \frac{1}{2^n} \sum_{z\in\{0,1\}^n} (-1)^{\langle x\cdot z\rangle} (\sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle) \\ &= \frac{1}{2^n} \sum_{z\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} ((-1)^{\langle x\cdot z\rangle} (-1)^{f(x)}) |x\rangle\end{aligned} \tag{6.5}$$

Ειδικότερα, ας επικεντρωθούμε στην περίπτωσή μας, δηλαδή $|x\rangle = |00\ldots0\rangle$ που αντιστοιχεί στον όρο $|0\rangle^{\otimes n}$. Έτσι, αν $|x\rangle = |00\ldots0\rangle$ υποδηλώνει ότι $\langle x \cdot z\rangle = 0$, κι εφόσον $(-1)^0 = 0$, χρησιμοποιώντας τη σχέση 6.5 μπορούμε να δούμε ότι:

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{2^n} \sum_{z\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} ((-1)^{\langle x\cdot z\rangle}(-1)^{f(x)})\,|x\rangle \\ &= \frac{1}{2^n} \sum_{z\in\{0,1\}^n} \sum_{x\in\{0,1\}^n} ((-1)^{f(x)})\,|x\rangle \end{aligned} \tag{6.6}$$

- Αν η συνάρτηση είναι **σταθερή**, $f(x) = 1, \forall x$ ή $f(x) = 0, \forall x$, υποδηλώνει ότι $(-1)^{f(x)} = \pm 1$ και η πιθανότητα να έχουμε το αποτέλεσμα $|00\cdots0\rangle$ είναι 1.
- Σε αντίθετη περίπτωση, αν η συνάρτηση $f(x)$ είναι **ισορροπημένη**, τα μισά εξαγόμενα αποτελέσματα είναι μηδενικά και τα άλλα μισά είναι μονάδες. Αυτό σημαίνει ότι τα αποτελέσματα αλληλοεξουδετερώνονται και το τελικό εξαγόμενο 0 υποδηλώνει ότι η πιθανότητα να έχουμε κατάσταση $|00\cdots0\rangle$ είναι επίσης 0.

Έτσι, πάλι με μια μόνο μέτρηση μπορούμε να καταλήξουμε στο συμπέρασμα αν η συνάρτηση είναι σταθερή ή ισορροπημένη. Συμπερασματικά, ο αλγόριθμος Deutsch απέδειξε την υπεροχή της Κβαντικής Υπολογιστικής, ενώ ο αλγόριθμος Deutsch-Jozsa, ο οποίος είναι η γενίκευση του αλγόριθμου Deutsch, αποδεικνύει επιπλέον και την εκθετική επιτάχυνση όταν χρησιμοποιούνται κβαντικοί υπολογιστές.

6.3 Ο αλγόριθμος Bernstein-Vazirani

Ο αλγόριθμος Bernstein-Vazirani είναι ένας κβαντικός αλγόριθμος που προτάθηκε από τους Ethan Bernstein και Umesh Vazirani το 1997. Ο σκοπός αυτού του αλγορίθμου είναι η εύρεση μιας *κρυφής* ακολουθίας bit (bit-sequence) που αποτελείται έστω από n bits. Χρησιμοποιώντας έναν κλασικό υπολογιστή χρειάζεται n αναζητήσεις, ενώ μια κβαντική υπολογιστική συσκευή θα χρειαστεί μία και μοναδική αναζήτηση. Ο αλγόριθμος Bernstein-Vazirani μπορεί να θεωρηθεί ως επέκταση του αλγορίθμου Deutsch-Jozsa.

Ορισμός του προβλήματος: Ας θεωρήσουμε μία Boolean συνάρτηση $f : \{0,1\}^n \rightarrow$ 0 ή 1 με την ιδιότητα (promise) $f(x)$ να είναι το εσωτερικό γινόμενο των x και της κρυφής bit sequence $s \in \{0,1\}^n modulo 2 \Rightarrow f(x) = x_0 s_0 \oplus x_1 s_1 \oplus \cdots \oplus x_{n-1} s_{n-1}$. Το πρόβλημα ανάγεται στην εύρεση της κρυφής bit-sequence s. Με άλλα λόγια, δεδομένης μίας κρυφής bit-sequence, για παράδειγμα $s = \{01000110\ldots\}$, να σχεδιασθεί ένας αλγόριθμος για την ανακάλυψη της s.

Για παράδειγμα, εάν $s = 010$, και έστω $x_0 = 100, x_1 = 001$, και $x_2 = 110$. Τότε,

$$x_0 \cdot s = (1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0) modulo 2 = 1$$
$$x_1 \cdot s = (1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1) modulo 2 = 0$$
$$x_2 \cdot s = (1 \cdot 1 + 1 \cdot 1 + 0 \cdot 0) modulo 2 = 0$$

που αντιπροσωπεύει ακριβώς το εσωτερικό γινόμενο των αντίστοιχων διανυσμάτων $modulo 2$. Επομένως, δεδομένης μίας έστω 3-bits συνάρτηση η οποία υπακούει στην ιδιότητα των Bernstein-Vazirani, $f(x) = s \cdot x$ και της οποίας οι τιμές δίνονται στον Πίνακα **6.1**, να προσδιοριστεί η κρυφή bit-sequence s.

Πίνακας 6.1 Bernstein-Vazirani: 3-bits συνάρτηση.

x	f(x)	x	f(x)
000	0	100	0
001	0	101	0
010	1	110	1
011	1	111	1

6.3.1 Κλασσική προσέγγιση

Για να βρεθεί η κρυφή συμβολοσειρά (bit-sequence) το μόνο που επιτρέπεται είναι να ερωτηθεί η συνάρτηση για διαφορετικές τιμές εισόδου. Επομένως, μπορούμε να αρχίσουμε με την κατάσταση '100..0' σαν πρώτο ερώτημα. Αν το πρώτο bit είναι 1, η συνάρτηση θα δώσει 1 και αν είναι 0, θα δώσει 0. Ομοίως, για να βρούμε το δεύτερο bit χρησιμοποιούμε την κατάσταση '010...0' ως ερώτημα. Για να απλοποιήσουμε το ερώτημα ζητάμε ένα bit ανά φορά. Έτσι, μπορούμε να ξεκινήσουμε με το '0' και αφού το συγκρίνουμε με την κατάλληλη έξοδο της συνάρτησης, αν αυτά είναι ίσα δεν υπάρχει καμία ενέργεια, διαφορετικά μπορούμε να αλλάζουμε το '0' σε '1'. Δηλαδή, ο αλγόριθμος αυτός έχει πολυπλοκότητα της τάξης $\Omega(n)$.

```
s = input('Secret binary number =');
n = len(s) #number of digits
s=list(s) #change the type of s list
t=[] #empty list
for i in range(n):  #initialize t -> '0000....'
    t.append('0')
for i in range(n):
    if t[i] != s[i]:
        t[i]='1'
for i in range(n):
    print(t[i], end=' ')
```

Πρόγραμμα 6.3: Bernstein-Vazirani: Κλασσική προσέγγιση.