



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

## Τίτλος Διπλωματικής

## ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΗΛΙΑΝΑΣ Θ. ΜΠΕΤΣΟΥ

Επιβλέπων: Επιβλέπων  
Τίτλος

Αθήνα, Ιούνιος 2019





Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

## Τίτλος Διπλωματικής

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΗΛΙΑΝΑΣ Θ. ΜΠΕΤΣΟΥ

Επιβλέπων: Επιβλέπων  
Τίτλος

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την Ημερομηνία Παρουσίασης.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....  
Επιβλέπων  
Τίτλος

.....  
1ος Εξεταστής  
Τίτλος

.....  
2ος Εξεταστής  
Τίτλος

Αθήνα, Ιούνιος 2019





Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Copyright ©—All rights reserved Ηλιάνα Μπέτσου, 2019.

Με την επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέχρινε.

### **Τπεύθυνη Δήλωση**

Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας, και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής Τ.Ε. του ΤΕΙ Πελοποννήσου.

(Υπογραφή)

.....  
Ηλιάνα Μπέτσου



# Περίληψη

Λέξεις Κλειδιά



# **Abstract**

## **Keywords**



*Aφιέρωση*



# Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ. .... για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων. Επίσης ευχαριστώ ιδιαίτερα τον Δρ. ..... για την καθοδήγησή του και την εξαιρετική συνεργασία που είχαμε. Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.



# Περιεχόμενα

Περίληψη	i
Abstract	iii
Ευχαριστίες	vii
Περιεχόμενα	x
<b>1 Introduction and Overview</b>	<b>1</b>
1.1 History . . . . .	1
1.2 Future Directions . . . . .	1
<b>2 Fundamentals</b>	<b>3</b>
2.1 Overview . . . . .	3
2.2 Quantum Computation . . . . .	3
2.2.1 Qubits . . . . .	3
2.2.2 Quantum Circuits . . . . .	3
2.3 The Dirac notation nad Hilbert spaces . . . . .	3
2.4 Quantum Circuits . . . . .	3
2.4.1 The quantum Circuit model . . . . .	3
2.4.2 Quantum Gates . . . . .	3
2.4.3 Universal Quantum Gates . . . . .	3
<b>3 Introduction to Computer Science</b>	<b>5</b>
3.1 Overview . . . . .	5
3.2 Turing Machines . . . . .	5
3.3 Computanional Complexity . . . . .	5
3.4 Problems in classes P and NP . . . . .	5
<b>4 Quantum Algorithms</b>	<b>7</b>
4.1 Probabilistic Versus Quantum Algorithms . . . . .	7
4.2 Quantum Parallelism . . . . .	7
4.3 Deutsch's Algorithm . . . . .	7

---

4.4	Deutsch-Jozsa Algorithm . . . . .	7
4.5	Simon's Algorithm . . . . .	7
4.6	Grover's Algorithm . . . . .	7
4.7	Quantum Search Algorithms . . . . .	7
<b>5</b>	<b>The quantum Fourier Transformation</b>	<b>9</b>
5.1	FFT . . . . .	9
5.2	Quantum Fourier Transformation . . . . .	13
<b>6</b>	<b>The Shor's Code</b>	<b>15</b>
6.1	Quantum error-correction . . . . .	15
<b>7</b>	<b>RSA Cryptosystem</b>	<b>17</b>
<b>8</b>	<b>Future Work/What's Happening Now</b>	<b>19</b>
<b>9</b>	<b>Summary</b>	<b>21</b>
<b>A'</b>	<b>Παράδειγμα Παραρτήματος</b>	<b>23</b>
A'.1	Πρώτη ενότητα . . . . .	23
A'.2	Μελλοντικές Επεκτάσεις . . . . .	23

# **Kεφάλαιο 1**

## **Introduction and Overview**

### **1.1 History**

### **1.2 Future Directions**



# **Kεφάλαιο 2**

## **Fundamentals**

**2.1 Overview**

**2.2 Quantum Computation**

**2.2.1 Qubits**

**2.2.2 Quantum Circuits**

**2.3 The Dirac notation nad Hilbert spaces**

**2.4 Quantum Circuits**

**2.4.1 The quantum Circuit model**

**2.4.2 Quantum Gates**

**2.4.3 Universal Quantum Gates**



## **Κεφάλαιο 3**

# **Introduction to Computer Science**

**3.1 Overview**

**3.2 Turing Machines**

**3.3 Computational Complexity**

**3.4 Problems in classes P and NP**



## **Kεφάλαιο 4**

# **Quantum Algorithms**

**4.1 Probabilistic Versus Quantum Algorithms**

**4.2 Quantum Parallelism**

**4.3 Deutsch's Algorithm**

**4.4 Deutsch-Jozsa Algorithm**

**4.5 Simon's Algorithm**

**4.6 Grover's Algorithm**

**4.7 Quantum Search Algorithms**



## Κεφάλαιο 5

# The quantum Fourier Transformation

### 5.1 FFT

Οι πράξεις μεταξύ πολυωνύμων μεγάλου μεγέθους χρησιμοποιούνται ευρέως. Η ολοκλήρωση τους ‘χειροκίνητα’ είναι μια χρονοβόρα και δύσκολη διαδικασία. Η επίλυση σε αυτό έρχεται με τη χρήση του Γρήγορου Μετασχηματισμού [1] *Fourier (FFT)*.

Ο *FFT*, βασίζεται σε κάποιες βασικές αρχές και ιδιότητες. Η κυριότερη είναι η ιδιότητα των πολυωνύμων να χαρακτηρίζονται με δύο διαφορετικές μορφές.

Το πολυώνυμο  $A(x)$  μπορεί να χαρακτηριστεί με τους εξής δύο τρόπους:

- Μέσω των συντελεστών του  $a_0, a_1, \dots, a_d$
- Μέσω των τιμών του  $A(x_0), A(x_1), \dots, A(x_d)$

Πολύ συνοπτικά, ο *FFT* είναι ένας αλγόριθμος διαίρει-και-βασίλευε, που βασίζεται στις ιδιότητες των μιγαδικών ριζών της μονάδας. Χρησιμοποιεί τον διακριτό μετασχηματισμό *Fourier (DFT)* και τον αντίστροφο *DFT* για να μετατρέπει τις δύο μορφές αναπαράστασης του πολυωνύμου. Από την αναπαράσταση του πολυωνύμου βαθμού  $d$  μέσω συντελεστών, προκύπτει ένα διάνυσμα με τους συντελεστές  $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_d)$ . Η απεικόνιση ενός πολυωνύμου βαθμού  $d$  σε μορφή τιμών μας δίνει ένα πακέτο από  $d$  ζευγάρια της μορφής  $(x_0, y_0), (x_1, y_1), \dots, (x_d, y_d)$  τέτοια ώστε κάθε  $x_k$  να είναι διακριτοί αριθμοί και κάθε  $y_k = A(x_k)$  για κάθε  $k = 0, 1, \dots, d$ .

Η κεντρική ιδέα πάνω στην οποία βασίστηκε ο *FFT*, είναι η δυνατότητα κάθε πολυωνύμου βαθμού  $d$ , να μπορεί να χαρακτηριστεί μοναδικά από την τιμή του σε κάθε  $d + 1$  σημείο του. Έστω λοιπόν ότι έχουμε ένα γινόμενο  $C(x)$ , δύο πολυωνύμων  $A(x)$  και  $B(x)$  βαθμού  $d$  το καθένα. Αυτό σημαίνει ότι το  $C(x)$ , θα έχει βαθμό  $2d$ . Με βάση λοιπόν την παραπάνω ιδιότητα, το  $C(x)$ , θα μπορεί να χαρακτηριστεί μοναδικά σε κάθε  $2d + 1$  σημείο του. Εάν λοιπόν το  $x$  είναι γνωστό, έστω  $z$ , ο πολλαπλασιασμός είναι μια διαδικασία γραμμικού χρόνου, καθώς το αποτέλεσμα θα προκύπτει εάν κάνουμε  $A(z)$  φορές το  $B(z)$ . Ο πολλαπλασιασμός δύο πολυωνύμων  $A(x)$  και  $B(x)$  βαθμού  $d$  το καθένα, θέλει χρόνο  $\theta(d^2)$ , καθώς κάθε συντελεστής στο διάνυσμα  $\alpha$  πρέπει να πολλαπλασιαστεί με κάθε συντελεστή στο διάνυσμα  $\beta$ . Το

διάνυσμα συντελεστών του γινομένου  $C(x)$ ,  $c = (c_0, c_1, \dots, c_{2d})$  ονομάζεται *convolution* των διανυσμάτων  $\alpha$  και  $\beta$  και γράφεται ως  $c = \alpha \otimes \beta$ .

Όμως εδώ έχουμε να αντιμετωπίσουμε το πρόβλημα, ότι και τα πολυώνυμα εισόδου αλλά και το γινόμενο θέλουμε να είναι σε μορφή συντελεστών. Επομένως πρέπει να γίνουν δύο μετατροπές. Μια μετατροπή από την μορφή των συντελεστών στην μορφή των τιμών (*evaluation*) τα επιλεγμένα σημεία στα οποία θα κάνουμε τον πολλαπλασιασμός και στη συνέχεια μετατρέπουμε ξανά το τελικό πλέον αποτέλεσμα σε μορφή συντελεστών (*interpolation*). Η διαδικασία *evaluation* στο σημείο  $x_0$ , αποτελείται από τον υπολογισμό της τιμής  $A(x_0)$ . Χρησιμοποιώντας το σχήμα *Horner* βρίσκουμε την τιμή σε χρόνο  $\theta(n)$ , ως εξής:

$$A(x_0) = \alpha_0 + x_0(\alpha_1 + x_0(\alpha_2 + \dots + x_0(\alpha_d \dots)))$$

Η αντίστροφη διαδικασία, δηλαδή η *interpolation* καθορίζει την τιμή του συντελεστή του πολυωνύμου, από την τιμή που αναπαρίσταται. Για κάθε ‘πακέτο’  $(x_0, y_0), (x_1, y_1), \dots, (x_d, y_d)$ , όπου κάθε  $x_k$  είναι διακριτός αριθμός, υπάρχει ένα μοναδικό πολυώνυμο  $A(x)$  βαθμού  $d$  για το οποίο ισχύει ότι  $y_k = A(x_k)$ , για κάθε  $k = 0, 1, \dots, d$ . Χρησιμοποιώντας τη μέθοδο *Lagrange*, μπορούμε να κάνουμε την *interpolation* σε χρόνο  $\theta(d^2)$ .

Το αμέσως επόμενο ερώτημα που προκύπτει είναι το πώς ακριβώς θα επιλεγούν τα σημεία με βαθμό  $< n - 1$  του  $A(x)$ , στα οποία θα εφαρμόσουμε τους μετασχηματισμούς. Για να μειώσουμε τις πράξεις κατά το μέγιστο δυνατό, επιλέγουμε ζευγάρια θετικών-αρνητικών αριθμών,  $\pm x_i, \dots, \pm x_{\frac{n}{2}-1}$ . Με πιο απλά λόγια, χωρίζουμε το πολυώνυμο  $A(x)$ , με αυτόν τον τρόπο:

$$A(x) = A_e(x^2) + x A_o(x^2)$$

, όπου  $A_e$  ο συντελεστής των άρτιων δυνάμεων και  $A_o$  ο συνεντελεστής των περιττών δυνάμεων. Ο υπολογισμός ενός ζευγαριού  $\pm x_i$  γίνεται ακόμα πιο έγκολος καθώς ισχύει η εξής ιδιότητα:

$$A(x_i) = A_e(x_i^2) + x_i A_o(x_i^2) \text{ kai } A(-x_i) = A_e(-x_i^2) - x_i A_o(x_i^2)$$

Κάνοντας *evaluation* του  $A(x)$  σε  $n$  θετικά αρνητικά σημεία  $\pm x_0, \dots, \pm x_{\frac{n}{2}-1}$  μειώνει την διαδικασία στα  $A_e(x)$  και  $A_o(x)$  σε  $\frac{n}{2}$  σημεία. Έτσι λοιπόν το αρχικό πρόβλημα μήκους  $n$  μειώνεται σε δύο υποπροβλήματα μήκους  $\frac{n}{2}$  το καθένα και σε κάποιες σχετικά απλές πράξεις. Ο συνολικός χρόνος εκτέλεσης είναι  $T(n) = 2T(\frac{n}{2}) + O(n)$ , το οποίο ανάγεται σε  $O(n \log n)$ , που είναι ένας πολύ ικανοποιητικός χρόνος.

Η τεχνική αυτή με την επιλογή θετικών-αρνητικών ζευγαριών εφαρμόζεται στο πρώτο επίπεδο. Για να προχωρήσουμε στο επόμενο επίπεδο θέλουμε  $\frac{n}{2}$  σημεία  $x_0^2, \dots, x_{\frac{n}{2}-1}^2$  να είναι τα ίδια θετικά-αρνητικά ζευγάρια. Δεδομένου όμως ότι τα σημεία αυτά είναι υψηλέντα στο τετράγωνο, είναι αδύνατο χωρίς την χρήση μιγαδικών αριθμών. Στο τελευταίο επίπεδο της αναδρομής θα έχουμε μόνο ένα σημείο, το  $\pm 1$ . Στο ακριβώς προηγούμενο επίπεδο της

αναδρομής θα έχουμε τις ρίζες του  $\pm 1$ , δηλαδή  $\pm i$ . Συνεχίζουμε έτσι σε κάθε επίπεδο μέχρι που καταλήγουμε στην  $n$ -οστή ρίζα του συνόλου, η οποία είναι οι μιγαδικές ρίζες της εξίσωσης  $z_n = 1$ .

Μέχρι αυτό το σημείο έχουμε δει πως γίνεται η μετατροπή σε τιμές και ο πολλαπλασιασμός αυτών. Ο *FFT*, μετατρέπει από της μορφή συντελεστών στη μορφή τιμών σε  $O(n \log n)$  όταν τα σημεία  $x_i$  είναι οι μιγαδικές  $n$ -οστές ρίζες του 1 ( $1, \omega, \omega^2, \dots, \omega^{n-1}$ ). Σχηματικά ισχύει το εξής:

$$\langle \text{values} \rangle = \text{FFT}(\langle \text{coefficients} \rangle, \omega).$$

Ωστόσο δεν μπορούμε να αγνοήσουμε τους συντελεστές, καθώς σε αυτή τη μορφή μας δίνονται όλα τα δεδομένα μας. Η τελευταία αυτή μετατροπή γίνεται με τη διαδικασία *interpolation*.

$$\langle \text{coefficients} \rangle = \frac{1}{n} \text{FFT}(\langle \text{values} \rangle, \omega^{-1}).$$

Για να αποκτήσουμε μια καλύτερη εικόνα της *interpolation*, πρέπει να δούμε λίγο πιο αναλυτικά τη σχέση ανάμεσα στις δύο διαφορετικές απεικονίσεις του  $A(x)$ . Και οι δύο μορφές αποτελούν διανύσματα  $n$  αριθμών και η κάθε απεικόνιση είναι ο γραμμικός μετασχηματισμός της άλλης.

$$\begin{bmatrix} A(x_0) \\ A(x_1) \\ \vdots \\ A(x_{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ & & \vdots & & \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^{n-1} \end{bmatrix} \cdot \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{bmatrix}$$

Ο μεσαίος πίνακας ονομάζεται  $M$  και έχει κάποιες συγκεκριμένες ιδιότητες. Εάν τα  $x_0, \dots, x_{n-1}$  είναι διακριτοί αριθμοί, τότε ο  $M$  είναι αντιστρέψιμος. Η ύπαρξη του  $M^{-1}$ , μας δίνει τη δυνατότητα να αντιστρέψουμε την εξίσωση μήτρας και να εκφράσουμε την μορφή των συντελεστών σε μορφή τιμών. Με λίγα λόγια όταν κάνουμε *evaluating* πολλαπλασιάζουμε με τον  $M$  και όταν κάνουμε *interpolation*, πολλαπλασιάζουμε με τον  $M^{-1}$ .

Ας προσπαθήσουμε να εξηγήσουμε λίγο μαθηματικά τον *FFT*. Με όρους γραμμικής άλγεβρας, ο γρήγορος μετασχηματισμός *Fourier*, έναν αυθαίρετο *vector*, διαστάσεως  $n$  (ο οποίος αποτελείται από τους συντελεστές του πολυωνύμου) με έναν πίνακα  $n \times n$  της παραχάτω μορφής:

$$M_n(\omega) = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ & & \vdots & & \\ 1 & \omega^j & \omega^{2j} & \dots & \omega^{(n-1)j} \\ & & \vdots & & \\ 1 & \omega^{(n-1)} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{bmatrix}, \text{ όπου } \eta \text{ πρώτη σειρά}$$

προκύπτει για  $\omega^0 = 1$ , η δεύτερη για  $\omega$ , η τρίτη για  $\omega^2$  μέχρι την τελευταία που προκύπτει

για  $\omega^{(n-1)}$ . Το  $\omega$  είναι η  $n$ -οστή μιγαδική ρίζα του 1 και το  $n$  είναι μια δύναμη του 2. Βλέπουμε ότι ο συγκεκριμένος πίνακας είναι πολύ απλό να περιγραφεί καθώς σε κάθε  $(j, k)$  θέση βρίσκεται το  $\omega^{jk}$ . Αυτό που μπορούμε να παρατηρήσουμε σε αυτό το στάδιο είναι ότι οι στήλες του  $M$  είναι ορθογώνιες μεταξύ τους. Αν πάρουμε το αποτέλεσμα από δύο τυχαίες στήλες του  $M$ , έστω  $j$  και  $k$  τότε προκύπτει το εξής:

$$1 + \omega^{j-k} + \omega^{2(j-k)} + \dots + \omega^{(n-1)(j-k)}$$

, το οποίο είναι γεωμετρική σειρά με πρώτο όρο το 1 και τελευταίο το  $\omega^{(n-1)(j-k)}$ . Έπομένως μετατρέπεται στο  $(1 - \omega^{n(j-k)}) / (1 - \omega^{(j-k)})$ , το οποίο είναι 0 για κάθε τιμή εκτός από  $j = k$ , όπου σε αυτή την περίπτωση όλοι οι όροι είναι 1 και το τελικό σύνολο  $n$ . Οι στήλες αυτές θα μπορούσαν να υπερηφανεύνων ως η βάση ενός εναλλακτικού συστήματος συντεταγμένων, το οποίο συχνά αποκαλείται βάση *Fourier*. Ο πολλαπλασιασμός ενός διανύσματος με τον  $M$ , οδηγεί στην περιστροφή του κλασσικού συστήματος συντεταγμένων στο σύστημα βάσης *Fourier*. Ο αντίστροφος  $M$ , προκαλεί την αντίστροφη περιστροφή. Με λίγα λόγια ισχύει ότι:  $M_n(\omega^{-1}) = \frac{1}{n} M_n(\omega^{-1})$ . Όμως το  $\omega^{-1}$  είναι και η  $n$ -οστή ρίζα της μονάδας και έτσι κάνοντας *interpolation* επί της ουσίας κάνουμε *FFT*, μόνο που αντί για  $\omega$  έχουμε  $\omega^{-1}$ . Κοιτάζοντας λοιπόν συνολικά μέχρι εδώ βλέπουμε ότι και από γεωμετρικής άποψης, ο πολλαπλασιασμός μεγάλων πολυωνύμων είναι αρκετά πιο εύκολος στην βάση *Fourier*, από ότι στην κλασσική βάση. Αρχικά περιστρέφουμε τα διανύσματα σε βάση *Fourier* (*evaluation*), στη συνέχεια κάνουμε την πράξη που θέλουμε (στην προκειμένη περίπτωση πολλαπλασιασμό) και τέλος περιστρέφουμε τα διανύσματα ξανά αντίστροφα (*interpolation*). Τα αρχικά διανύσματα είναι η απεικόνιση σε μορφή συντελεστών, όταν περιστρέφονται μετατρέπονται σε μορφή τιμών και μετά την αντίστροφη περιστροφή επανέρχονται σε μορφή συντελεστών. Η γρήγορη εναλλαγή μεταξύ των δύο αυτών καταστάσεων είναι ο γρήγορος μετασχηματισμός *Fourier*.

Αυτό είναι το συνολικό υπόβαθρο του *FFT*. Ωστόσο το πιο ενδιαφέρον κομμάτι του είναι η υπορουτίνα που κάνει αυτή την εναλλαγή που είδαμε πιο πάνω. Ο *FFT* παίρνει σαν είσοδο ένα διάνυσμα  $\alpha = (\alpha_0, \dots, \alpha_{n-1})$  και έναν μιγαδικό αριθμό  $\omega$ , του οποίου οι δυνάμεις αποτελούν τις μιγαδικές ποστές ρίζες της μονάδας. Πολλαπλασιάζει το διάνυσμα με τον πίνακα  $M_n(\omega)$  διάστασης  $n \times n$ , ο οποίος έχει σαν είσοδο σε κάθε  $j, k$  σημείο του το αντίστοιχο  $\omega^{jk}$ . Ο διαχωρισμός που αναλύσαμε παραπάνω σε ζεύγη θετικών αρνητικών είναι πολύ βοηθητικός σε αυτό ακριβώς το σημείο καθώς οι στήλες του  $M_n$  χωρίζονται σε αρνητικούς και θετικούς. Στο επόμενο βήμα απλοποιούμε τα στοιχεία στο κάτω μισό του πίνακα χρησιμοποιώντας τα  $\omega^{n/2} = -1$  και  $\omega^n = 1$ . Το πάνω αριστερά κομμάτι του πίνακα όπως και το κάτω αριστερά με διάσταση  $n/2 \times n/2$  είναι το  $M_{n/2}(\omega^2)$ . Επίσης ο πάνω δεξιά και ο κάτω δεξιά υποπίνακας είναι σχεδόν ίδιοι με τους προηγούμενους, μόνο που οι  $j$ οστές σειρές τους είναι πολλαπλασιασμένες με το  $\omega^j$  και  $-\omega^j$ , αντίστοιχα. Έτσι λοιπόν το τελικό αποτέλεσμα είναι ακινητό με την ζητούμενη διάνυσμα μας.

Συνοπτικά ο *FFT* έχει ως εξής:

---

Fast Fourier Transform

---

```

1: functionFFT( $\alpha, \omega$ )
2: Input : An array  $\alpha = (\alpha_0, \dots, \alpha_{n-1})$ , for  $n$  a power of 2
3:           A primitive  $n$ th root of unity,  $\omega$ 
4: Output :  $M_n(\omega)\alpha$ 
5:
6: if  $\omega = 1$  then return  $\alpha$ 
7:  $(s_0, s_1, \dots, s_{\frac{n}{2}-1}) = FFT((\alpha_0, \alpha_2, \dots, \alpha_{n-2}, \omega^2$ 
8: for  $j = 0 \rightarrow \frac{n}{2} - 1$  do
9:    $r_j = s_j + \omega^j s'_j$ 
10:   $r_{j+\frac{n}{2}} = s_j - \omega^j s'_j$ 
return  $(r_0, r_1, \dots, r_{n-1})$ 

```

---

## 5.2 Quantum Fourier Transformation

Όπως έχουμε αναφέρει και παραπάνω η μεγαλύτερη δύναμη των κβαντικών υπολογιστών, είναι η δυνατότητα να επιτελέσουν πράξεις και να επιλύσουν προβλήματα που δεν είναι δυνατό με τους κλασσικούς υπολογιστές. Για παράδειγμα η παραγοντοποίηση σε πρώτους αριθμούς ενός  $n - bit$  ακεραίου χρησιμοποιώντας τον καλύτερο δυνατό κλασσικό αλγόριθμο, θα χρειαζόταν  $exp(\Theta(n^{1/3} \log^{2/3} n))$ . Αυτό επί της ουσίας είναι εκθετικά το μέγεθος του ακεραίου τον οποίο θέλουμε να παραγωγίσουμε. Γι' αυτόν ακριβώς τον λόγο, το πρόβλημα της παραγοντοποίησης θεωρείται άλυτο στους κλασσικούς υπολογιστές. Αντίστοιχα, ένας κβαντικός υπολογιστής, έχει τη δυνατότητα να λύσει το ίδιο πρόβλημα σε  $O(n^2 \log n \log(\log n))$ , που σημαίνει ότι ένας κβαντικός υπολογιστής μπορεί να λύσει εκθετικά πιο γρήγορα το συγκεκριμένο πρόβλημα. Αυτό μπορεί από μόνο του να είναι εντυπωσιακό, ωστόσο σίγουρα δημιουργούνται τα ερωτήματα του πόσα άλλα προβλήματα μπορεί να λυθούν με τη χρήση κβαντικών υπολογιστών. Εδώ θα εξετάσουμε τον κβαντικό μετασχηματισμό *Fourier* (*Quantum Fourier Transformation*), ο οποίος αποτελεί τη βάση για πολλούς κβαντικούς αλγορίθμους.

Στον γρήγορο μετασχηματισμό *Fourier* παίρνουμε σαν είσοδο ένα μιγαδικό διάνυσμα  $M$ -διάστασης, α και σαν έξοδο επιστρέφει ένα μιγαδικό διάνυσμα  $M$ -διάστασης,  $\beta$ . Έχουμε δηλαδή το εξής:

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{M-1} \end{bmatrix} = \frac{1}{\sqrt{M}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{M-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(M-1)} \\ & & \vdots & & \\ 1 & \omega^j & \omega^{2j} & \dots & \omega^{(M-1)j} \\ & & \vdots & & \\ 1 & \omega^{(M-1)} & \omega^{2(M-1)} & \dots & \omega^{(M-1)(M-1)} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{M-1} \end{bmatrix}$$

,όπου το  $\omega$  είναι η Μοστή μιγαδική ρίζα της μονάδας. Οι κλασσικές μέθοδοι θα χρειαζόταν χρόνο  $O(M^2)$ , ενώ ο γρήγορος μετασχηματισμός *Fourier*(*FFT*), μπορεί να κάνει

ακριβώς το ίδιο σε χρόνο  $O(M \log M)$ . Παρά την μεγάλη αύξηση στην ταχύτητα που προκαλεί ο *FFT*, ο κβαντικός μετασχηματισμός *Fourier*, (*QFT*), καταφέρνει να μειώσει και άλλο τον χρόνο εκτέλεσης εκθετικά φτάνοντας τον σε  $O(\log^2 M)$ . Όπως είδαμε αναλυτικά παραπάνω, ο διαχριτός μετασχηματισμός *Fourier* έχει σαν είσοδο ένα διάνυσμα μιγαδικών αριθμών  $(x_0, \dots, x_N - 1)$ , όπου  $N$  είναι το μήκος τους διανύσματος. Η έξοδος μετά τη μετατροπή που εφαρμόζει ο αλγόριθμος είναι ένα διάνυσμα  $y_0, \dots, y_N - 1$ , το οποίο ορίζεται ως εξής:

Ο κβαντικός μετασχηματισμός *Fourier*,

## Κεφάλαιο 6

# The Shor's Code

### 6.1 Quantum error-correction



Κεφάλαιο 7

## RSA Cryptosystem



**Κεφάλαιο 8**

## **Future Work/What's Happening Now**



**Κεφάλαιο 9**

## **Summary**



## Παράρτημα Α'

# Παράδειγμα Παραρτήματος

A'.1 Πρώτη ενότητα

A'.2 Μελλοντικές Επεκτάσεις



# Βιβλιογραφία

[1] Fast fourier transform, χ.χ.



# Συντομογραφίες - Αρχτικόλεξα - - Ακρωνύμια

βλπ	βλέπε
κ.λπ.	και λοιπά
κ.ο.κ	και ούτω καθεξής
ΤΕΙ	Τεχνολογικό Εκπαιδευτικό Ίδρυμα
BPF	Band Pass Filter



# Απόδοση ξενόγλωσσων όρων

## Απόδοση

αδερφός  
αμεταβλητότητα  
ανάκτηση πληροφορίας  
αντιμεταθετικότητα  
απόγονος  
απορρόφηση  
βάση δεδομένων  
γνώρισμα  
διαπροσωπεία  
διαφορά  
δικτυακός κατάλογος  
δικτυωτή δομή<sup>1</sup>  
δομικές επερωτήσεις  
δομικές σχέσεις  
δομικό σχήμα  
εγκυρότητα  
ένωση

## Ξενόγλωσσος όρος

sibling  
idempotency  
information retrieval  
commutativity  
descendant  
absorption  
database  
attribute  
interface  
difference  
portal catalog  
lattice  
structural queries  
structural relationships  
schema  
validity  
union









ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Πρότυπο Σύστημα Ομότιμων  
Κόμβων Βασισμένο σε Σχήματα RDF**

Κωνσταντίνος Δ. Δημητρίου

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2014



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Πρότυπο Σύστημα Ομότιμων  
Κόμβων Βασισμένο σε Σχήματα RDF**

Κωνσταντίνος Δ. Δημητρίου

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2014

