



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

ΕΡΓΑΣΙΑ 1 ΣΤΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΑΠΟΚΕΝΤΡΩΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ – Chord DHT

<Σεϊτανίδης Νικόλαος >

A.M. <1072553>

<Οικονομόπουλος Ιωάννης >

A.M. <1072582>

Πάτρα, <2024>

Περιεχόμενα

Περιγραφή εργασίας	3
Υλοποίηση web crawler.....	3
Πιθανή απόκλιση του web crawler στα πραγματικά δεδομένα.....	5
Υλοποίηση δομής CHORD	7
Αποτελέσματα και παραδείγματα	10
Παράρτημα.....	15
Αναφορές	15

Περιγραφή εργασίας

Η παρούσα εργασία υλοποιεί την δομή δεδομένων Chord DHT χρησιμοποιώντας ως δεδομένα τα στοιχεία των Computer Scientists τα οποία ανακτήθηκαν μέσω ενός web crawler από τη σελίδα της Wikipedia: https://en.wikipedia.org/wiki/List_of_computer_scientists. Για την υλοποίηση της δομής και του crawler χρησιμοποιήθηκε η γλώσσα Python μέσω του περιβάλλοντος Visual Studio Code σε λειτουργικά Windows. Οι βασικότερες βιβλιοθηκές που χρησιμοποιήθηκαν για κάθε υλοποίηση φαίνονται παρακάτω.

Chord DHT:

- **Hashlib**: Βιβλιοθήκη η οποία προσφέρει συναρτήσεις κατακερματισμού όπως η Sha-1 που χρησιμοποιούνται για την κωδικοποίηση των key value.
- **Math**: Βιβλιοθήκη μαθηματικών που προσφέρει συναρτήσεις που εκτελούν μαθηματικές πράξεις όπως η **log2()** για την εκτέλεση λογαρίθμων με βάση το 2
- **Pandas**: Βιβλιοθήκη η οποία επιτρέπει το διάβασμα αρχείων .csv σε μορφή ενός dataframe
- **Ippaddress**: Βιβλιοθήκη που προσφέρει δυνατότητες ελέγχου ενός string για να διαπιστώσει αν η μορφή του string είναι της μορφής IPv4 ή IPv6
- **logging**: Βιβλιοθήκη απαραίτητη για την διευκόλυνση στο debugging αφού καταγράφει ορισμένα δεδομένα κατά την διάρκεια της εκτέλεσης του προγράμματος.
- **csv**: Βιβλιοθήκη απαραίτητη για την επεξεργασία csv αρχείων και στην περίπτωση μας δημιουργία και φόρτωμα τιμών σε αρχείο csv
- **requests**: Βιβλιοθήκη απαραίτητη για την δημιουργία http requests ώστε να αντλήσουμε πληροφορίες από την wikipedia
- **BeautifulSoup**: Βιβλιοθήκη απαραίτητη για την ανάκτηση δεδομένων από html και xml αρχεία.
- **urljoin**: Βιβλιοθήκη απαραίτητη για την επεξεργασία και ανάκτηση δεδομένων από κάποιο URL
- **re**: Βιβλιοθήκη απαραίτητη για ειδικές εκφράσεις και στην περίπτωσή μας για την αναζήτηση ημερομηνίας, δηλαδή τετραψήφιου αριθμού για τον υπολογισμό του αριθμού των βραβείων του κάθε επιστήμονα.

Υλοποίηση web crawler

Προκειμένου να αντλήσουμε δεδομένα από ένα αρχείο για την δημιουργία του Chord αλλά και για την υλοποίηση των επιμέρους λειτουργιών, δημιουργήσαμε έναν Web Crawler. Σκοπός είναι να δοθεί σαν είσοδος στο πρόγραμμα το URL που δίνεται στην εκφώνηση της εργασίας με την λίστα των επιστημόνων και να δημιουργηθεί ένα αρχείο csv. Αυτό θα αποτελείται από 3 στήλες (Επίθετο, Αριθμός Βραβείων και Όνομα Πανεπιστημίου) και θα δημιουργείται το Chord βάσει αυτού του αρχείου.

Όσον αφορά το πρόγραμμα, το οποίο έχει υλοποιηθεί σε γλώσσα Python στο Visual Studio Code, λειτουργεί ως εξής:

- Ορίζεται η κλάση Crawler η οποία περιέχει κάθε συνάρτηση που χρειάζεται ο Crawler για την λειτουργία του. Η `init()` ορίζει το αρχικό URL και τα υπόλοιπα URLs που έχει επισκεφτεί ο crawler.
- Η `download_url()` κάνει αίτηση για να πάρει το html του κάθε URL το οποίο θα χρειαστούμε παρακάτω για να αντλήσουμε τις πληροφορίες που χρειαζόμαστε.
- Η `parse_scientists_list()` ουσιαστικά είναι η μέθοδος που φορτώνει το επίθετο του κάθε επιστήμονα από την λίστα στο αρχείο csv. Ουσιαστικά με την `find_all('li')` βρίσκει όλα τα στοιχεία της λίστας (σε μορφή bullets) τα οποία περιέχουν στον τίτλο τους το 'wiki', δηλαδή παραπέμπουν σε βιογραφία του κάθε επιστήμονα. Χωρίζει τον τίτλο του URL που αντιστοιχεί στο full name του εκάστοτε επιστήμονα και κρατά το τελευταίο μέρος, θεωρώντας το ως επώνυμο. Τέλος, αφού απομονώσει το επίθετο το προσθέτει σε μια λίστα "scientists".
- Η `parse_scientists_page()` είναι η συνάρτηση η οποία παίρνει τα δεδομένα βάσει το κάθε URL του εκάστοτε επιστήμονα από την λίστα scientists. Καλεί την `download_url()` και κατεβάζει το περιεχόμενο του html του URL. Γίνεται σε κάθε ένα από τα URL αναζήτηση ενός header με την λέξη 'award' μέσα (συγκεκριμένα στα contents). Τα βραβεία δίνονται με διάφορους τρόπους. Ο πρώτος είναι η μέτρηση των bullets, δηλαδή η μέτρηση των στοιχείων της λίστας σε πεδία που περιέχουν την λέξη 'award', όπου και αυξάνεται ένας μετρητής για τα βραβεία κατά 1 κάθε φορά. Εάν δεν βρεθεί βραβείο με αυτόν τον τρόπο, ψάχνει στο ίδιο πεδίο 4-ψήφιους αριθμούς που αντιστοιχούν ουσιαστικά σε ημερομηνίες που απονέμονται τα βραβεία, αφού δίνεται αναλυτικό κείμενο με έτος απονομής και όχι λίστα, όπως στην 1^η περίπτωση.
- Στην συνέχεια για την εύρεση του πανεπιστημίου του κάθε επιστήμονα, έχει δημιουργηθεί μια λίστα `search_terms`. Για τους όρους της λίστας γίνεται αναζήτηση με προτεραιότητα στον πρώτο όρο. Αυτό γίνεται με αυτόν τον τρόπο αφού δεν ακολουθείται συγκεκριμένη δομή στις Wikipedia των επιστημόνων, οπότε δεν βρίσκεται αυστηρά κάπου στο κείμενο. Οπότε, ο crawler αναζητά το 1^ο URL που περιέχει κάποιον από αυτούς τους όρους, περιορίζοντας την παρουσία αριθμών ή ανεπιθύμητων χαρακτήρων ώστε να είναι ρεαλιστικά το URL με το πανεπιστήμιο του κάθε επιστήμονα. Αφού βρεθεί αυτό, επιστρέφεται ο τίτλος του URL που ουσιαστικά αντιστοιχεί στον τίτλο του Πανεπιστημίου του επιστήμονα.
- Η `run()` ουσιαστικά ανοίγει ένα αρχείο csv όπου και θα αποθηκευτούν τα δεδομένα. Καλούνται οι παραπάνω συναρτήσεις ώστε να πάρουμε τα δεδομένα για κάθε επιστήμονα. Φορτώνει τα δεδομένα βάσει τις συναρτήσεις που καλούμε στο αρχείο `computer_scientists.csv`
- Τέλος, ορίζεται η `main` στην οποία δημιουργείται ένα αντικείμενο της κλάσης Crawler με βάσει το URL με την λίστα των επιστημόνων. Καλείται η `run()` για να γίνουν τα παραπάνω.

Πιθανή απόκλιση του web crawler στα πραγματικά δεδομένα

Κατά την διάρκεια της υλοποίησης του crawler παρατηρήθηκε πως η δομή των Wikipedia δεν είναι η ίδια για κάθε επιστήμονα. Όπως έχει αναφερθεί και παραπάνω, για παράδειγμα για την αρίθμηση των βραβείων έχουν μετρηθεί τα bullets που απαριθμούν τα βραβεία ως λίστα, αλλιώς αν δεν υπάρχει αυτή η δομή ως bullets μετριοούνται οι 4-ψήφιοι αριθμοί που θεωρητικά αντιπροσωπεύουν τις ημερομηνίες απονομής των βραβείων. Αυτό δεν είναι απαραίτητα αντικειμενικό, καθώς μπορεί να μην είναι πάντα η ημερομηνία απονομής ενός βραβείου αλλά κάποιου άλλου επιτεύγματος. Επίσης, μπορεί να αναφέρεται με άλλον τρόπο ο αριθμός των βραβείων, τον οποίον δεν εντοπίσαμε. Παρατηρήθηκε πως σε ορισμένες περιπτώσεις αναφέρεται διαφορετικός αριθμός βραβείων στο κείμενο της Wikipedia και διαφορετικός στις περιληπτικές πληροφορίες του επιστήμονα κάτω από την φωτογραφία του. Σε κάθε περίπτωση ο αριθμός βραβείων είναι ακέραιος, οπότε τα δεδομένα είναι ρεαλιστικά.

Όσον αφορά το όνομα, στις περισσότερες περιπτώσεις πρόκειται για επιστήμονες των δυτικών χωρών όπου το επίθετο τους είναι το τελευταίο όνομα από το full name τους. Όμως σε ορισμένες περιπτώσεις ενδέχεται να μην ισχύει το ίδιο εάν ανήκει σε συγκεκριμένο πολιτισμικό σύνολο. Παρόλα αυτά, επειδή δεν μας είναι ξεκάθαρο σε ποια ακριβώς ονόματα το τελευταίο όνομα δεν είναι το επίθετο που ζητείται, εμείς φορτώνουμε στο csv αρχείο το τελευταίο όνομα σε κάθε περίπτωση.

Τέλος, όσον αφορά και το Πανεπιστήμιο του κάθε επιστήμονα έχουμε θεωρήσει το πανεπιστήμιο που σπούδασε. Συγκεκριμένα, θεωρούμε τον τίτλο του 1^{ου} URL που περιέχει τα "univers", "institut", "school", το οποίο τις περισσότερες φορές είναι το πανεπιστήμιο που φοίτησε. Όμως, πολλές φορές είναι το πανεπιστήμιο στο οποίο δίδασκε ή στο πανεπιστήμιο που πήρε το phd του. Θεωρήσαμε αυτόν τον πιο βέλτιστο τρόπο ώστε να αντλήσουμε όσο το δυνατόν πιο ρεαλιστικά δεδομένα. Σε μερικές περιπτώσεις το πανεπιστήμιο μπορεί να είναι Null, αφού δεν βρέθηκε κάποιο, ή και να περιέχει ειδικούς χαρακτήρες, αφού το csv δεν μπορούσε να μεταφράσει για παράδειγμα τα τονούμενα γράμματα στα γαλλικά, χωρίς αυτό να εμποδίζει την υλοποίηση του Chord.

Παραθέτουμε ενδεικτικό screenshot στα δεδομένα του αρχείου csv:

40	Bernus	0	Griffith University			
41	Bezos	0	Princeton University			
42	Bhushan	0	Indian Institute of Technology Kanpur			
43	Blaauw	0	Delft University of Technology			
44	Blei	2	Brown University			
45	Blum	0	Null			
46	Blum	0	Carnegie Mellon University			
47	Blum	0	University of California, Berkeley			
48	Boehm	8	University of Southern California			
49	Bollacker	0	University of Texas at Austin			
50	Bonwick	0	Null			
51	Booch	9	University of California, Santa Barbara			
52	Boole	3	University College Cork			
53	Booth	0	University of London			
54	Booth	0	Birkbeck College, University of London			
55	Borg	5	New York University			
56	Bos	0	University of Groningen			
57	Botvinnik	0	Oxford University Press			
58	Bowen	0	University College			
59	Bourne	0	University of Cambridge Computer Laboratory			
60	Bouwman	0	VU University Amsterdam			
61	Boyer	0	The University of Texas at Austin			
62	Brandenb	28	University of Erlangen-Nuremberg			
63	Brassard	11	Université de Montréal			
64	Breed	0	Stanford University			
65	Bresenhan	0	Winthrop University			
66	Brin	0	University of Maryland, College Park			
67	Hansen	6	Technical University of Denmark			
68	Brinkkemper	0	Utrecht University			
69	Brooks	26	Duke University			
70	Brooks	0	Stanford University			
71	Burnett	0	Oregon State University			
72	Gil	3	University of La Laguna			
73	Camp	2	Michigan State University			
74	Candlin	0	University of Cambridge			
75	Canion	0	University of Houston			
76	Cantrill	0	Brown University			
77	Cardelli	3	University of Pisa			
78	Carmack	0	University of Missouri—Kansas City			
79	Casperse	0	Aarhus University			
80	Catmull	12	University of Utah			
81	Cerf	38	Stanford University			

Υλοποίηση δομής CHORD

Αρχικά για την υλοποίηση της δομής χρησιμοποιήθηκαν τρεις κλάσεις οι οποίες ονομαστικά είναι η Node, η Key και η Chord. Η κλάση Node αφορά τους κόμβους του ring και κάθε αντικείμενο της έχει τιμές `Ip`, `port`, `id` που προκύπτει από το κατακερματισμό του στοιχείου `ip:port` και τη διαίρεση με το μέγεθος του ring, μια λίστα `data` που περιέχει τα κλειδιά που αντιστοιχούν στον κόμβο, έναν κόμβο `prev` που δείχνει τον προηγούμενο κόμβο εντός του ring, ένα finger table μεγέθους K όπου 2^K είναι το μέγεθος του ring και συναρτήσεις για εμφάνιση κόμβων και ενημέρωση του finger table. Η κλάση Key αφορά τα κλειδιά που βρίσκονται στο ring και για κάθε κλειδί υπάρχουν οι τιμές `key`, `awards`, `surname` όπου αφορούν τα πεδία `education`, `awards`, `surname` από το αρχείο `.csv`, υπάρχει τιμή `id` η οποία παράγεται από τον κατακερματισμό του στοιχείου `key` και τη διαίρεση με το μέγεθος του ring και υπάρχει και ένας κόμβος `succ` που δείχνει τον κόμβο στον οποίο αντιστοιχεί το εκάστοτε κλειδί. Τέλος η κλάση Chord περιέχει τη δομή και έχει τιμή `size` για το μέγεθος του ring το οποίο ορίζει πόσοι κόμβοι μπορούν να βρίσκονται στο ring και το πεδίο τιμών για τα `id` τους (π.χ. αν $K = 7$ το μέγεθος του ring $2^K = 128$, άρα 128 κόμβοι με `id` στο $[0,127]$), τιμή `num_of_nodes` για τον αριθμό των κόμβων εντός του ring, έναν αρχικό κόμβο και υλοποιεί και όλες τις συναρτήσεις της δομής οι οποίες αναλύονται παρακάτω αναλυτικά.

- **Update_finger_table:** Εκτελεί την ενημέρωση του finger table ενός κόμβου n όπου εκτελεί K επαναλήψεις ξεκινώντας με $i = 0$ και υπολογίζει το `id` κάθε επανάληψης ως `table_id = n + 2^i` και βρίσκει τον successor για την θέση i του finger table μέσω της `find_successor` για ορίσματα το `table_id` και το `id` του αρχικού κόμβου (π.χ. αν n έχει `id` 10, $K = 7$ το `table_id` για την θέση 3 του finger_table είναι $10 + 2^2 = 14$ και βρίσκει τον successor βάσει του 14 και του αρχικού για να βρει τον κόμβο που αντιστοιχεί στη θέση `finger_table[2]`)
- **Distance:** Υπολογίζει την απόσταση δύο κόμβων στο ring $n1, n2$. Αν το `id` του $n1$ ίσο με του $n2$ τότε επιστρέφει 0, αν το `id` του $n2$ μεγαλύτερο επιστρέφει το αποτέλεσμα της αφαίρεσης των `id` $n2 - n1$ και αν το `id` του $n1$ μεγαλύτερο επιστρέφει την απόσταση μέσω του κύκλου του ring χρησιμοποιώντας το `size` και επιστρέφοντας το `size - n1 + n2` (π.χ. αν το `size` = 128, $n1 = 12$, $n2 = 5$ τότε το αποτέλεσμα είναι $128 - 12 + 5 = 121$)
- **Update_all_fingers:** Ξεκινώντας από τον αρχικό κόμβο ενημερώνει το finger table του κόμβου και έπειτα χρησιμοποιεί αναδρομικά την πρώτη εισαγωγή του finger table κάθε κόμβου για να ενημερώσει όλα τα finger tables.
- **Hash_id:** Συνάρτηση που δοσμένου `ip, port` υπολογίζει την κατακερματισμένη τιμή του `ip:port` μέσω Sha-1 συνάρτησης και έπειτα υπολογίζει το `id` του κόμβου με τη διαίρεση με το μέγεθος του ring (2^K) ώστε το `id` να βρίσκεται στο πεδίο $[0,127]$.
- **Join_node:** Η συνάρτηση αυτή αφορά την εισαγωγή ενός κόμβου στο ring. Αρχικά ελέγχει αν στο ring υπάρχει αρχικός κόμβος και σε περίπτωση που δεν υπάρχει αρχικοποιεί το ring με αρχικό κόμβο τον δοσμένο προς εισαγωγή. Η εισαγωγή του κόμβου ξεκινά με την παραγωγή του `id` βάση των τιμών `Ip, port` και `size` του ring. Στην συνέχεια αν υπήρχαν κόμβοι στο ring βρίσκει τον successor του κόμβου μέσω της συνάρτησης `find_successor`. Βάσει του successor αρχικοποιεί τις τιμές `prev` των δύο κόμβων και ενημερώνει τα finger table των κόμβων. Τέλος ελέγχει αν κάποιο από τα

κλειδιά του successor πρέπει να μεταφερθεί στον νέο κόμβο χρησιμοποιώντας την συνάρτηση distance.

- **Find_successor:** Συνάρτηση που δέχεται ως ορίσματα δύο κόμβους n1, n2 και βρίσκει τον successor του n2 ξεκινώντας από τον n1. Συνήθως ο n1 είναι ο αρχικός κόμβος του ring. Έπειτά υπάρχουν τρεις περιπτώσεις αναζήτησης. Η πρώτη είναι τα id των κόμβων να είναι ίδια οπότε ο successor είναι ο ίδιος ο κόμβος. Η δεύτερη αφορά την σύγκριση της απόστασης του n1,n2 και της απόστασης του successor του n1(successor του n1 ο κόμβος στη θέση finger_table[0]), n2. Αν η απόσταση n1,n2 μικρότερη της απόστασης finger_table[0], n2 τότε ο successor του n2 είναι ο successor του n1. Διαφορετικά αν δεν ισχύουν οι πρώτες δύο περιπτώσεις επαναλαμβάνουμε την σύγκριση για κάθε 2 κόμβους στο finger_table του n1 ως εξής finger_table[i], n2 και finger_table[i+1],n2 και αν η σύγκριση ισχύει τότε ορίζουμε ως next_node τον κόμβο στη θέση finger_table[i] ή αν δεν ισχύει η σύγκριση για κανένα ζευγάρι τιμών ο next_node είναι ο κόμβος στη τελευταία θέση. Τέλος επαναλαμβάνουμε τη διαδικασία με τον next_node να είναι ο n1.
- **Leave_node:** Διαγράφει έναν κόμβο από το ring εφόσον αυτός υπάρχει. Αν ο κόμβος είναι μοναδικός, αρχικός κόμβος τότε ορίζει ως κενό τον αρχικό κόμβο του ring. Αν δεν είναι μοναδικός στο ring τότε μεταφέρει όλα τα κλειδιά του κόμβου στον successor του και ενημερώνει τα finger_table όλων των κόμβων. Τέλος αν δεν ήταν μοναδικός αλλά ήταν αρχικός ορίζει ως νέο αρχικό κόμβο τον successor.
- **Search_node:** Εφόσον υπάρχουν κόμβοι στο ring αναζητά έναν δοσμένο κόμβο βάσει id. Αρχικά ελέγχει αν ο αρχικός κόμβος έχει το id που αναζητείται αλλιώς ελέγχει αναδρομικά κάθε κόμβο χρησιμοποιώντας τους successor κάθε κόμβου. Αν ο κόμβος βρεθεί τότε εμφανίζει μήνυμα επιτυχίας αλλιώς εμφανίζει μήνυμα αποτυχίας.
- **Insert_key:** Συνάρτηση που εισάγει ένα κλειδί στο ring εφόσον το ring έχει δημιουργηθεί και υπάρχει έστω ένας κόμβος εντός. Εφόσον μπορεί να εισαχθεί το υπολογίζεται το id του κλειδιού με παρόμοιο τρόπο με τον υπολογισμό του id ενός κόμβου με τη διαφορά πως δεν χρησιμοποιείται το ip:port αλλά το education πεδίο. Έπειτα βάσει του id βρίσκεται ο successor κόμβος του κλειδιού. Αν ο κόμβος δεν περιέχει κλειδιά τότε το κλειδί προστίθεται αλλιώς αν περιέχει ελέγχεται πρώτα για το αν υπάρχει πανομοιότυπο κλειδί και στην περίπτωση που υπάρχει τότε η εισαγωγή του νέου απορρίπτεται.
- **Lookup_key:** Παρόμοια με την αναζήτηση κόμβου με τη διαφορά πως βρίσκει όλα τα κλειδιά με δοσμένο key value που παράγουν ίδιο id αλλά μπορεί να έχουν διαφορετικές τιμές awards, surname (π.χ. όλα τα κλειδιά με value CEID έχουν ίδιο ID αλλά διαφορετικές τιμές).
- **Update_key:** Δοσμένων όλων των τιμών ενός κλειδιού key, awards, surname γίνεται αναζήτηση ενός κλειδιού και ενημέρωση των τιμών awards, surname ή ενός από τις δύο.
- **Delete_key:** Βρίσκει ένα συγκεκριμένο κλειδί όπως η update_key όμως αντί για ενημέρωση τιμών διαγράφει το κλειδί από το ring.
- **Range_search:** Υλοποιεί αναζήτηση εύρους ανάμεσα στα κλειδιά με το εύρος να χωρίζεται σε δύο στάδια. Το πρώτο αφορά των αριθμό των βραβείων και έχει ένα ελάχιστο και ένα μέγιστο όριο. Το δεύτερο στάδιο αφορά το πρώτο γράμμα του

επωνύμου και έχει επίσης ένα ελάχιστο και ένα μέγιστο όριο (π.χ. Ένα ερώτημα εύρους είναι τα κλειδιά με συγκεκριμένο id και βραβεία από 0 έως 4 και πρώτο γράμμα επωνύμου από G έως Z). Για να πραγματοποιηθεί η αναζήτηση αυτή αρχικά αναζητούνται τα κλειδιά με συγκεκριμένο id βάσει δοσμένου key (δηλαδή δοσμένου education π.χ. CEID). Στη συνέχεια για κάθε κλειδί που ταιριάζει στο id ελέγχονται οι τιμές awards, surname για το αν ταιριάζουν με τα όρια του εύρους και εφόσον ταιριάζουν προστίθεται το κλειδί στη λίστα αποτελεσμάτων.

Οι παραπάνω συναρτήσεις αφορούν τις βασικές συναρτήσεις για την υλοποίηση της δομής Chord Ring DHT. Ωστόσο υπάρχουν και ορισμένες βοηθητικές συναρτήσεις όπως συναρτήσεις print για εμφάνιση στοιχείων του ring. Αυτές οι συναρτήσεις είναι οι εξής:

- **Print_finger:** Συνάρτηση που εμφανίζει το finger_table ενός κόμβου.
- **Print_prev:** Συνάρτηση που για κάθε κόμβο εμφανίζει τον προηγούμενο στο ring.
- **Print_nodes:** Συνάρτηση που εμφανίζει τους κόμβους που υπάρχουν στο ring.
- **Print_keys:** Συνάρτηση που εμφανίζει κάθε κόμβο και τα κλειδιά που αντιστοιχούν στο κόμβο εντός του ring.
- **Check_for_duplicate_node:** Συνάρτηση που ελέγχει αν υπάρχει ήδη κόμβος με συγκεκριμένο id και επιστρέφει true αν υπάρχει αλλιώς επιστρέφει false.
- **Validate_ip:** Ελέγχει ένα αλφαριθμητικό για το αν βρίσκεται στη μορφή της IPv4 διεύθυνσης (δηλαδή x.x.x.x με x στο πεδίο [0,255]).
- **Validate_port:** Ελέγχει αν ένα αλφαριθμητικό είναι έγκυρος αριθμός Port (δηλαδή αριθμός από το 0 έως το 65535).
- **Validate_alnum_spaces:** Ελέγχει αν η είσοδος του χρήστη είναι αλφαριθμητικό με κενά.
- **Validate_alpha_with_spaces:** Ελέγχει αν η είσοδος του χρήστη περιέχει μόνο γράμματα και κενά.
- **Create_ring:** Αρχικοποιεί το ring με ένα προκαθορισμένο αριθμό κόμβων και εισάγει στο ring τα κλειδιά που προκύπτουν από το .csv αρχείο.

Αποτελέσματα και παραδείγματα

Για την χρήση της δομής δημιουργήθηκε ένα μενού 12 επιλογών (0-11). Παρακάτω αναλύεται κάθε επιλογή και δίνονται στιγμιότυπα για τα αποτελέσματα της καθεμιάς.

- **Επιλογή 0, Δημιουργία του ring και εισαγωγή προκαθορισμένων δεδομένων για κόμβους και κλειδιά:** Με την επιλογή αυτή δημιουργούνται 5 κόμβοι που αποτελούν το ring και δημιουργούνται τα κλειδιά βάσει του αρχείου `computer_scientists.csv`.

```
Here are the nodes that are part of the chord ring:  
Start node: 102  
Node id: 113  
Node id: 28  
Node id: 59  
Node id: 74
```

Εικόνα 1: Τα id των προκαθορισμένων κόμβων που εισάγονται στο ring.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
Here are the keys that are part of each node:  
Start node: 102  
--Key id: 100 Key value: b'University of Toronto' Key awards: 0 Key surname: Aho  
--Key id: 99 Key value: b'South Dakota State University' Key awards: 7 Key surname: Amdahl  
--Key id: 76 Key value: b'Wesleyan University' Key awards: 0 Key surname: Anderson  
--Key id: 79 Key value: b'University of Florida' Key awards: 6 Key surname: Anthony  
--Key id: 90 Key value: b'Purdue University' Key awards: 0 Key surname: Arden  
--Key id: 79 Key value: b'University of Florida' Key awards: 0 Key surname: Atanasoff  
--Key id: 85 Key value: b'Carnegie Mellon University' Key awards: 0 Key surname: Blum  
--Key id: 79 Key value: b'University of Groningen' Key awards: 0 Key surname: Bos  
--Key id: 87 Key value: b'Oxford University Press' Key awards: 0 Key surname: Botvinnik  
--Key id: 88 Key value: b'University of Cambridge Computer Laboratory' Key awards: 0 Key surname: Bourne  
--Key id: 99 Key value: b'Stanford University' Key awards: 0 Key surname: Breed  
--Key id: 93 Key value: b'Utrecht University' Key awards: 0 Key surname: Brinkkemper  
--Key id: 99 Key value: b'Stanford University' Key awards: 0 Key surname: Brooks  
--Key id: 86 Key value: b'Oregon State University' Key awards: 0 Key surname: Burnett  
--Key id: 91 Key value: b'Michigan State University' Key awards: 2 Key surname: Camp  
--Key id: 97 Key value: b'University of Cambridge' Key awards: 0 Key surname: Candlin  
--Key id: 99 Key value: b'Stanford University' Key awards: 38 Key surname: Cerf  
--Key id: 99 Key value: b'Stanford University' Key awards: 0 Key surname: Chou  
--Key id: 90 Key value: b'Purdue University' Key awards: 8 Key surname: Comer  
--Key id: 86 Key value: b'University of New South Wales' Key awards: 0 Key surname: Compton  
--Key id: 96 Key value: b'University College London' Key awards: 0 Key surname: Cox  
--Key id: 97 Key value: b'University of Cambridge' Key awards: 8 Key surname: Crowcroft  
--Key id: 99 Key value: b'Stanford University' Key awards: 0 Key surname: Daswani  
--Key id: 97 Key value: b'University of Cambridge' Key awards: 0 Key surname: Date
```

Εικόνα 2: Παράδειγμα μερικών κλειδιών που βρίσκονται στο κόμβο 102.

- **Επιλογή 1, Εισαγωγή νέου κόμβου:** Ο χρήστης δίνει την Ip διεύθυνση και την θύρα(port) του νέου κόμβου προς εισαγωγή στο ring. Εφόσον είναι έγκυρη η διεύθυνση, η θύρα και δεν υπάρχει κόμβος με ίδια στοιχεία στο ring εισάγεται ο κόμβος στο δίκτυο μέσω της `join_node`.

```

Menu:
0. Create a ring with pre determined values of keys based on a list of computer scientists
1. Add a new node
2. Delete a node
3. Print nodes
4. Search node and print its finger table
5. Insert a key
6. Print keys
7. Lookup key
8. Update key
9. Delete key
10. Range query
11. Exit
Enter your choice (0/1/2/3/4/5/6/7/8/9/10/11):
1

Please enter the ip address of the new node(x.x.x.x format): 2.2.2.2

Please enter the port of the new node(dddd format): 2000

Node joined the ring succesfully.

```

Εικόνα 3: Παράδειγμα εισαγωγής νέου κόμβου μέσω της επιλογής 1

```

Here are the nodes that are part of the chord ring:
Start node: 102
Node id: 113
Node id: 21
Node id: 28
Node id: 59
Node id: 74

```

Εικόνα 4: Λίστα κόμβων μετά την εισαγωγή του νέου κόμβου με id = 21

- **Επιλογή 2, Διαγραφή κόμβου:** Ο χρήστης δίνει το id ενός κόμβου προς διαγραφή και εφόσον το id είναι έγκυρο και υπάρχει στο δίκτυο ο κόμβος αφαιρείται.

```

Here are the nodes that are part of the chord ring:
Start node: 102
Node id: 113
Node id: 21
Node id: 28
Node id: 59
Node id: 74

Please enter the id of a node to delete(Positive integer): 21

Node found.

Node deleted succesfully.

```

Εικόνα 5: Παράδειγμα διαγραφής του νέου κόμβου με id = 21

- **Επιλογή 3, Εμφάνιση κόμβων:** Εμφανίζει τους κόμβους που βρίσκονται εντός του ring. Παραδείγματα εκτέλεσης φαίνονται στις εικόνες 1,4.

- **Επιλογή 4, Αναζήτηση κόμβου και εμφάνιση του finger table του:** Γίνεται αναζήτηση ενός κόμβου βάσει του id του και εφόσον ο κόμβος υπάρχει εντός του ring εμφανίζεται το finger table.

```
Enter a node id to search for(Positive integer): 59

Node found.
Finger table of node: 59
--Position: 0 Node id: 74
--Position: 1 Node id: 74
--Position: 2 Node id: 74
--Position: 3 Node id: 74
--Position: 4 Node id: 102
--Position: 5 Node id: 102
--Position: 6 Node id: 28
```

Εικόνα 6: Αποτέλεσμα εκτέλεσης της επιλογής 4 για κόμβο με id = 59

- **Επιλογή 5, Εισαγωγή νέου κλειδιού:** Γίνεται δημιουργία και εισαγωγή ενός νέου κλειδιού βάσει των τιμών που δίνει ο χρήστης για τα πεδία value, awards, surname. Για να φανούν τα αποτελέσματα η εισαγωγή του κλειδιού στο στιγμιότυπο γίνεται σε ring που δεν περιέχει κλειδιά.

```
Please enter the value of the new key: CEID

Please enter the number of awards of the new key(Positive integer): 0

Please enter the surname of the new key(Only letters): Oikonomopoulos

Key inserted succesfully!
```

Εικόνα 7: Παράδειγμα εισαγωγής νέου κλειδιού

```
Here are the keys that are part of each node:
Start node: 21
--Key id: 122 Key value: b'CEID' Key awards: 0 Key surname: Oikonomopoulos
Node id: 53
Node id: 102
```

Εικόνα 8: Αποτέλεσμα εισαγωγής νέου κλειδιού

- **Επιλογή 6, Εμφάνιση κλειδιών:** Εμφανίζει όλα τα κλειδιά που βρίσκονται στο ring και τον κόμβο στον οποίο αντιστοιχούν. Στιγμιότυπα εκτέλεσης της επιλογής φαίνονται στις εικόνες 2, 7.

- **Επιλογή 7, Αναζήτηση κλειδιού:** Εμφανίζει όλα τα κλειδιά με συγκεκριμένο value, δηλαδή τα κλειδιά με ίδιο education πεδίο και συνεπώς με ίδιο id.

```
Please enter key value to search for: Harvard University
--Key id: 61 Key value: b'Harvard University' Key awards: 10 Key surname: Cook
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Curry
--Key id: 61 Key value: b'Harvard University' Key awards: 6 Key surname: Etzioni
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Even
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Gates
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Goguen
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Greibach
--Key id: 61 Key value: b'Harvard University' Key awards: 4 Key surname: Hillis
--Key id: 61 Key value: b'Harvard University' Key awards: 5 Key surname: Ingalls
--Key id: 61 Key value: b'Harvard University' Key awards: 3 Key surname: Karger
--Key id: 61 Key value: b'Harvard University' Key awards: 1 Key surname: Karp
--Key id: 61 Key value: b'Harvard University' Key awards: 15 Key surname: Lampson
--Key id: 61 Key value: b'Harvard University' Key awards: 5 Key surname: Minsky
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Morris
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Pham
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Press
--Key id: 61 Key value: b'Harvard University' Key awards: 7 Key surname: Ritchie
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Sadowsky
--Key id: 61 Key value: b'Harvard University' Key awards: 3 Key surname: Salton
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Shortcliffe
--Key id: 61 Key value: b'Harvard University' Key awards: 26 Key surname: Stallman
--Key id: 61 Key value: b'Harvard University' Key awards: 16 Key surname: Sutherland
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Sweeney
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Zuckerberg
```

Εικόνα 9: Αποτέλεσμα εκτέλεσης της επιλογής 7 για κλειδιά με value = Harvard University

- **Επιλογή 8, Ενημέρωση κλειδιού:** Δίνοντας και τις τρεις τιμές ενός κλειδιού μπορεί να γίνει ενημέρωση των τιμών awards, surname του κλειδιού. Στο στιγμιότυπο φαίνεται η ενημέρωση του κλειδιού με id = 61, value = Harvard University, awards = 10, surname = Cook σε awards = 12 ενώ τα υπόλοιπα στοιχεία παραμένουν ίδια.

```
Please enter key value to update: Harvard University
Please enter old awards value to update(Positive integer): 10
Please enter old surname value to update(Only letters): Cook
Please enter new awards value to update to(Positive integer or blank to not change): 12
Please enter new surname to update to(Only letter or blank to not change):
Key was updated.
```

Εικόνα 10: Παράδειγμα ενημέρωσης κλειδιού

```
Please enter key value to search for: Harvard University
--Key id: 61 Key value: b'Harvard University' Key awards: 12 Key surname: Cook
```

Εικόνα 11: Αποτέλεσμα ενημέρωσης κλειδιού

- **Επιλογή 9, Διαγραφή κλειδιού:** Παρομοίως με την επιλογή 8 δίνοντας κάθε τιμή ενός κλειδιού εφόσον οι τιμές είναι έγκυρες το κλειδί αφαιρείται από την δομή.

```
Please enter key value to delete: Harvard University

Please enter awards value(Positive integer): 10

Please enter surname value(Only letters): Cook

Key was removed.
```

Εικόνα 12: Παράδειγμα διαγραφής κλειδιού

```
Please enter key value to search for: Harvard University
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Curry
--Key id: 61 Key value: b'Harvard University' Key awards: 6 Key surname: Etzioni
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Even
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Gates
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Goguen
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Greibach
--Key id: 61 Key value: b'Harvard University' Key awards: 4 Key surname: Hillis
--Key id: 61 Key value: b'Harvard University' Key awards: 5 Key surname: Ingalls
--Key id: 61 Key value: b'Harvard University' Key awards: 3 Key surname: Karger
--Key id: 61 Key value: b'Harvard University' Key awards: 1 Key surname: Karp
--Key id: 61 Key value: b'Harvard University' Key awards: 15 Key surname: Lampson
--Key id: 61 Key value: b'Harvard University' Key awards: 5 Key surname: Minsky
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Morris
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Pham
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Press
--Key id: 61 Key value: b'Harvard University' Key awards: 7 Key surname: Ritchie
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Sadowsky
--Key id: 61 Key value: b'Harvard University' Key awards: 3 Key surname: Salton
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Shortliffe
--Key id: 61 Key value: b'Harvard University' Key awards: 26 Key surname: Stallman
--Key id: 61 Key value: b'Harvard University' Key awards: 16 Key surname: Sutherland
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Sweeney
--Key id: 61 Key value: b'Harvard University' Key awards: 0 Key surname: Zuckerberg
```

Εικόνα 13: Αποτέλεσμα της διαγραφής κλειδιού

- **Επιλογή 10, Αναζήτηση εύρους:** Εκτελεί αναζήτηση εύρους στα κλειδιά βάσει των τιμών που έδωσε ο χρήστης χρησιμοποιώντας τη range search συνάρτηση.

```
Please enter key value for the range search: Harvard University

Please enter minimum awards value(Positive integer with blank for no limit): 10

Please enter maximum awards value(Positive integer with blank for no limit):

Please enter minimum letter value(From A to Z with blank for no limit): B

Please enter maximum letter value(From A to Z with blank for no limit): Z
--Key id: 61 Key value: b'Harvard University' Key awards: 10 Key surname: Cook
--Key id: 61 Key value: b'Harvard University' Key awards: 15 Key surname: Lampson
--Key id: 61 Key value: b'Harvard University' Key awards: 26 Key surname: Stallman
--Key id: 61 Key value: b'Harvard University' Key awards: 16 Key surname: Sutherland
```

Εικόνα 14: Αποτέλεσμα εκτέλεσης αναζήτησης εύρους

- **Επιλογή 11, Έξοδος από το πρόγραμμα**

Παράρτημα

Αναφορές

1. <https://www.scrapingbee.com/blog/crawling-python/>
2. [https://en.wikipedia.org/wiki/Chord_\(peer-to-peer\)](https://en.wikipedia.org/wiki/Chord_(peer-to-peer))
3. Διαφάνειες μαθήματος