ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Bot Ιστού με ενισχυμένες δυνατότητες»

Του φοιτητή
Κωνσταντίνου Στουγιάννου
Αρ. Μητρώου: 144346

Επίκουρος Καθηγητής
Αντώνης Σιδηρόπουλος

Ημερομηνία 04/09/2022

Bot Ιστού με ενισχυμένες δυνατότητες
#21371
Στουγιάννου Κωνσταντίνος
Αντώνης Σιδηρόπουλος
15/10/2021
Ημερομηνία περάτωσης Δ.Ε. 04/09/2022

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Πρόλογος

Το web scrapping είναι μια τεχνική η οποία χρησιμοποιείται για την εξαγωγή δεδομένων από ιστοσελίδες. Ο τρόπος με τον οποίο το πετυχαίνει είναι ότι αφού κατεβάσει μια σελίδα, στη συνέχεια επεξεργάζεται τις html ετικέτες της με σκοπό να εξάγει τα δεδομένα στην μορφή που μας ενδιαφέρει. Συνήθως είναι χρήσιμο να μεταφερθούν απευθείας σε μια βάση δεδομένων ή σε csv αρχεία. Αυτή η τεχνολογία μπορεί να βρει εφαρμογή σε πολλές περιπτώσεις όπως η εξόρυξη δεδομένων, η παρακολούθηση αλλαγών τιμών αλλά και η έρευνα [1]. Ο λόγος που επιλέχθηκε το συγκεκριμένο θέμα είναι ανάγκη μελέτης των συγκεκριμένων τεχνολογιών διότι προσφέρουν επιπλέον δυνατότητες στις σύγχρονες εφαρμογές, τις κάνουν πιο ανεξάρτητες και αυτοματοποιούν τη διαδικασία τροφοδοσίας αλλά και ενημέρωσης των δεδομένων προς αξιοποίηση. Το όφελος που μου πρόσφερε είναι η άμεση εξοικείωση με τη συγκεκριμένη τεχνολογία που είναι ιδιαίτερα χρήσιμη σε όλων των ειδών των εφαρμογών, η καλή γνώση με τη γλώσσα προγραμματισμού python όπως και επιλεγμένων βιβλιοθηκών της όπως το Selenium και τέλος η εφαρμογή και βελτιστοποίηση αλγορίθμων δομών δεδομένων που βοηθούν στην αναζήτηση της πληροφορίας που μας ενδιαφέρει σε μορφή γράφου.

Περίληψη

Σκοπός της πτυχιακής είναι η δημιουργία ενός webbot το οποίο, μπορεί να υπολογίσει τη σημαντικότητα του κάθε συνδέσμου, βασιζόμενοι στα παρακάτω χαρακτηριστικά: μέγεθος εικόνας, μέγεθος γραμματοσειράς, χρώμα, θέση μέσα στην ιστοσελίδα. Για κάθε ιστοσελίδα αποθηκεύονται και οι βασικές πληροφορίες όπως το URL, ο τίτλος, το μέγεθος αλλά και η ώρα πρόσβασης και τροποποίησης. Όλα αυτά γίνονται με τη χρήση της python και της δημοφιλής βιβλιοθήκης της το Selenium. Γίνεται αναφορά στην τεχνολογία του web scrapping, στις διαθέσιμες τεχνολογίες που υπάρχουν, η σύγκριση μεταξύ τους αλλά και πως γίνεται ο αυτοματισμός μια εργασίας στον browser. Μετέπειτα γίνεται ανάλυση δύο αλγορίθμων δομών δεδομένων που μας βοηθούν στην οργάνωση και αναζήτηση της πληροφορίας σε δεντρική δομή. Τέλος παρουσιάζονται οι μελλοντικές επεκτάσεις της εφαρμογής αλλά και τα αποτελέσματα.

«Web Bot with Enhanced Capabilities»

Konstantinos Stougiannou

Abstract

The purpose of the dissertation is to create a webbot which can calculate the importance of each link, based on the following characteristics: image size, font size, color, position within the website. For each purchase, the basic information such as the URL, the title, the size but also the access time and the changes are stored. All this is done using python and its popular library Selenium. Reference is made to the technology of web scrapping, to the available technologies that exist, the comparison between them but also how to automate a task in the browser. Then two algorithm data structures are analyzed that help us organize and search for information in a tree structure. Finally, its future extensions and results are presented.

Περιεχόμενα

Κατάλογος Σχημάτων

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Κεφάλαιο 1ο: Web

1.1 Εισαγωγή

Ο Παγκόσμιος Ιστός (WWW), κοινώς γνωστός ως Ιστός, είναι ένα σύστημα πληροφοριών που επιτρέπει την πρόσβαση σε έγγραφα και άλλους πόρους του Ιστού μέσω του Διαδικτύου. Τα έγγραφα και τα μέσα με δυνατότητα λήψης διατίθενται στο δίκτυο μέσω διακομιστών Ιστού και είναι προσβάσιμα από προγράμματα όπως προγράμματα περιήγησης Ιστού. Οι διακομιστές και οι πόροι στον Παγκόσμιο Ιστό αναγνωρίζονται και εντοπίζονται μέσω συμβολοσειρών χαρακτήρων που ονομάζονται ενιαίοι εντοπιστές πόρων (URL). Ο αρχικός και ακόμα πολύ συνηθισμένος τύπος εγγράφου είναι μια ιστοσελίδα μορφοποιημένη σε γλώσσα σήμανσης υπερκειμένου (HTML). Αυτή η γλώσσα σήμανσης υποστηρίζει απλό κείμενο, εικόνες, ενσωματωμένα περιεχόμενα βίντεο και ήχου και σενάρια (σύντομα προγράμματα) που υλοποιούν πολύπλοκη αλληλεπίδραση με τον χρήστη. Η γλώσσα HTML υποστηρίζει επίσης υπερσυνδέσμους (ενσωματωμένες διευθύνσεις URL) που παρέχουν άμεση πρόσβαση σε άλλους πόρους Ιστού. Η παρακολούθηση τέτοιων υπερσυνδέσμων σε πολλούς ιστότοπους είναι γνωστή ως πλοήγηση στο διαδίκτυο, μερικές φορές γνωστή ως διαδικτυακή περιήγηση. Οι ιστοσελίδες που χρησιμεύουν ως λογισμικό εφαρμογής είναι γνωστές ως εφαρμογές Ιστού. Το πρωτόκολλο μεταφοράς υπερκειμένου χρησιμοποιείται για τη μεταφορά δεδομένων από τον Ιστό μέσω του Διαδικτύου (HTTP). Ένας ιστότοπος αποτελείται από πολλές ιστοσελίδες που μοιράζονται ένα κοινό θέμα και χρησιμοποιούν συχνά το ίδιο όνομα τομέα. Ενώ ορισμένοι ιστότοποι, ειδικά οι πιο διάσημοι, μπορεί να φιλοξενούνται από πολλούς διακομιστές, ένας μόνο διακομιστής ιστού μπορεί να φιλοξενεί πολλούς ιστότοπους. Ένας τεράστιος όγκος εκπαιδευτικών, ψυχαγωγικών, εμπορικών και κυβερνητικών πληροφοριών προσφέρεται από μια πληθώρα επιχειρήσεων, οργανισμών, κυβερνήσεων και μεμονωμένων χρηστών ως περιεχόμενο ιστότοπου. Η κυρίαρχη πλατφόρμα λογισμικού στον κόσμο σήμερα είναι ο Παγκόσμιος Ιστός. [2]

1.2 Web 1

Η βασική δομή του πρώιμου Ιστού ήταν ότι ένας μικρός αριθμός ατόμων που παρήγαγε ιστοσελίδες και περιεχόμενο για ένα μεγάλο κοινό αναγνωστών, επιτρέποντάς τους να αποκτήσουν γεγονότα, πληροφορίες και υλικό από διάφορες πηγές. Θα μπορούσε να αναφερθεί ότι το Web 1.0 δημιουργήθηκε για να διευκολύνει τους χρήστες να βρίσκουν πληροφορίες. Οι χρήστες που αναζητούσαν δεδομένα ήταν το επίκεντρο αυτής της έκδοσης του ιστού. Επειδή δεν διέθετε τις απαραίτητες φόρμες, εικόνες, στοιχεία ελέγχου αλλά και την αλληλεπίδραση που θεωρούμε δεδομένα στο σημερινό Διαδίκτυο, αυτή η έκδοση Ιστού αναφέρεται συχνά ως "ο Ιστός μόνο για ανάγνωση". Οι άνθρωποι αναφέρονται στην πρώιμη έκδοση του Διαδικτύου ως "Web 1.0". Οι χρήστες έγιναν μάρτυρες της πρώτης παρουσίας ενός παγκόσμιου δικτύου που υπαινίσσεται δυνατότητες μελλοντικής ψηφιακής επικοινωνίας και ανταλλαγής πληροφοριών. Τα χαρακτηριστικά του Web 1.0 περιλάμβαναν στατικές σελίδες που συνδέονται με ένα σύστημα χρησιμοποιώντας υπερσυνδέσμους. Διαθέτει στοιχεία της HTML 3.2, συμπεριλαμβανομένων πινάκων και πλαισίων. Το email χρησιμοποιείται για την αποστολή φορμών HTML. Αντί για ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων, το περιεχόμενο προέρχεται από το σύστημα αρχείων του διακομιστή. Επίσης διέθετε γραφικά και κουμπιά GIF. Τελικά το web 1

ήταν στην ουσία ένα ψηφιοποιημένο λεξικό το οποίο έγινε παγκόσμια διαθέσιμο σε πολλούς χρήστες στο διαδίκτυο. [3]

1.3 Web 2

Εάν το Web 1.0 αποτελούνταν από μια μικρή ομάδα ατόμων που παράγουν υλικό για μεγαλύτερο κοινό, το Web 2.0 αποτελείται από πολλά άτομα που παράγουν ακόμη περισσότερο περιεχόμενο για ένα διευρυνόμενο κοινό. Το Web 2.0 δίνει μεγαλύτερη έμφαση στη συμμετοχή και τη συνεισφορά από ότι το Web 1.0 στην ανάγνωση. Το περιεχόμενο που δημιουργείται από τον χρήστη, η χρηστικότητα, η αλληλεπίδραση και η βελτιωμένη συνδεσιμότητα με άλλα συστήματα και συσκευές είναι τα κύρια σημεία εστίασης αυτής της φόρμας Διαδικτύου. Στο Web 2.0, η εμπειρία του χρήστη ήταν το παν. Ως αποτέλεσμα, αυτή η φόρμα Ιστού ήταν υπεύθυνη για τη δημιουργία μέσων κοινωνικής δικτύωσης, συνεργασιών και κοινοτήτων. Ως εκ τούτου, το Web 2.0 θεωρείται ως η κυρίαρχη μέθοδος αλληλεπίδρασης Ιστού για την πλειοψηφία των χρηστών στον σημερινό κόσμο. Το Web 2.0 αναφέρεται ως "ο συμμετοχικός κοινωνικός ιστός", ενώ ο Web 1.0 ήταν γνωστός ως "ο Ιστός μόνο για ανάγνωση". Το Web 2.0 ενσωματώνει τεχνολογία προγράμματος περιήγησης ιστού όπως τα πλαίσια JavaScript για να δημιουργήσει μια καλύτερη, πιο βελτιωμένη έκδοση του προκατόχου του. Τα βασικά χαρακτηριστικά του Web 2.0 είναι να έχει δυναμικό περιεχόμενο που αλλάζει ανάλογα με την ανθρώπινη συμβολή και τις ενέργειες που κάνει ο χρήστης στην ιστοσελίδα, να χρησιμοποιούνται διεπαφές προγραμματισμού εφαρμογών (API) το οποίο προωθεί την αυτοχρήση και επιτρέπει αλληλεπιδράσεις όπως: Podcasting ,blogging με ετικέτες και ψηφοφορία σε περιεχόμενο ιστού. Χρησιμοποιείται από το κοινωνικό σύνολο και όχι μόνο από συγκεκριμένες κοινότητες. Η ανάπτυξη των κοινωνικών δικτύων και η συνδεσιμότητα στο Διαδίκτυο μέσω κινητού έχουν επιταχύνει σημαντικά την ανάπτυξη του Web 2.0. Η τεράστια δημοτικότητα των κινητών είναι ένας άλλος παράγοντας που πυροδοτεί αυτήν την έκρηξη. Επιπλέον, η ανάπτυξη του Web 2.0 επέτρεψε την επέκταση και την κυριαρχία δημοφιλών εφαρμογών αναπαραγωγής βίντεο. [3]

1.4 Web 3

Φτάσαμε στην πιο πρόσφατη έκδοση του Ιστού την Web 3. Πρέπει να κοιτάξουμε το μέλλον για να κατανοήσουμε την πραγματική έννοια του web 3.0. Αν και το Web 3.0 είναι ήδη σε ορισμένες μορφές, υπάρχει ακόμη δουλειά που πρέπει να γίνει προτού υλοποιηθεί πλήρως. Το θεμέλιο του Web 3.0, συχνά γνωστό ως Web3, αποτελείται από τις θεμελιώδεις έννοιες της αποκέντρωσης, της διαφάνειας και της βελτιωμένης χρησιμότητας των χρηστών. Το Web 3.0 είναι το "read, write, execute Web", ενώ το Web 1.0 είναι το "read-only Web", το Web 2.0 είναι ο "συμμετοχικός κοινωνικός ιστός" και ούτω καθεξής. Οι χρήστες μεταβαίνουν από δημοφιλής κεντρικές πλατφόρμες σε αποκεντρωμένες, πρακτικά ανώνυμες πλατφόρμες κατά τη διάρκεια αυτού του σταδίου αλληλεπίδρασης και χρήσης Ιστού. Ο Tim Berners-Lee, ο οποίος δημιούργησε τον Παγκόσμιο Ιστό, αρχικά αναφέρθηκε στο Web 3.0 ως Σημασιολογικός Ιστός και είδε ένα έξυπνο, αυτάρκης και ανοιχτό Διαδίκτυο που χρησιμοποιούσε την τεχνητή νοημοσύνη και τη μηχανική μάθηση για να λειτουργήσει ως «παγκόσμιος εγκέφαλος» και να ερμηνεύσει περιεχόμενο εννοιολογικά και συμφραζόμενα. Λόγω των τεχνολογικών περιορισμών, όπως το πόσο ακριβό και δύσκολο είναι να μεταφραστεί η ανθρώπινη γλώσσα σε μια μορφή που μπορούν να κατανοήσουν οι υπολογιστές, αυτή η εξιδανικευμένη έκδοση δεν λειτούργησε αρκετά. Τα βασικά χαρακτηριστικά του Web 3.0 είναι:

- Ένας σημασιολογικός ιστός είναι αυτός στον οποίο η δημιουργία, η κοινή χρήση και οι συνδέσεις περιεχομένου βάσει αναζήτησης και ανάλυσης καθίστανται δυνατές μέσω της

τεχνολογίας του Ιστού. Αντί να χρησιμοποιεί αριθμούς και λέξεις-κλειδιά, επικεντρώνεται στην κατανόηση λέξεων.

- Χρησιμοποιεί μηχανική μάθηση και τεχνητή νοημοσύνη. Το αποτέλεσμα είναι ένας υπολογιστής που χρησιμοποιεί το Web 3.0 για να γίνει πιο έξυπνος και πιο δεκτικός στις απαιτήσεις των χρηστών. Εάν αυτές οι ιδέες συνδυάζονται με την Επεξεργασία Φυσικής Γλώσσας (NLP), το αποτέλεσμα είναι ένας υπολογιστής που χρησιμοποιεί NLP.
- Επεξηγεί πώς το Διαδίκτυο των Πραγμάτων συνδέει διάφορες συσκευές και εφαρμογές (IoT). Αυτή η διαδικασία καθίσταται δυνατή μέσω σημασιολογικών μεταδεδομένων, επιτρέποντας την αποτελεσματική εκμετάλλευση όλων των διαθέσιμων δεδομένων. Επιπλέον, οποιοσδήποτε μπορεί να έχει πρόσβαση στο Διαδίκτυο από οπουδήποτε και ανά πάσα στιγμή χωρίς υπολογιστή ή άλλη έξυπνη συσκευή.
- Δίνεται στους χρήστες η επιλογή να αλληλεπιδρούν δημόσια ή ιδιωτικά χωρίς κάποιος μεσάζων να τους εκθέτει σε κινδύνους, παρέχοντας «απίστευτα» δεδομένα.
- Χρησιμοποιεί τρισδιάστατα γραφικά. Στην πραγματικότητα, αυτό είναι ήδη εμφανές στο ηλεκτρονικό εμπόριο, τις εικονικές περιηγήσεις και τα παιχνίδια υπολογιστών. Διευκολύνει τη συμμετοχή χωρίς να απαιτείται συναίνεση από μια κυβερνώσα οντότητα. Είναι χωρίς εξουσιοδότηση. Είναι κατάλληλο για:
- Metaverses: Ένα άπειρο, τρισδιάστατο εικονικό σύμπαν
- Βιντεοπαιχνίδια Blockchain Συμμορφώνονται με τα ιδανικά των NFT δίνοντας τη δυνατότητα στους χρήστες να κατέχουν πραγματική ιδιοκτησία πόρων εντός του παιχνιδιού.
- Ψηφιακή υποδομή και απόρρητο. Με αυτόν τον τρόπο χρησιμοποιούνται αποδείξεις μηδενικής γνώσης και πιο ασφαλείς προσωπικές πληροφορίες. Αυτή η χρήση συνοδεύεται από πληρωμή. Αποκεντρωμένοι αυτόνομοι οργανισμοί, ψηφιακές χρηματοοικονομικές συναλλαγές peer-to-peer, έξυπνα συμβόλαια και κρυπτονομίσματα Οι διαδικτυακές κοινότητες ανήκουν στην κοινότητα.

Στο τέλος, το Web 3.0 επιτρέπει την αλληλεπίδραση των χρηστών, την ανταλλαγή πληροφοριών και τις ασφαλείς οικονομικές συναλλαγές χωρίς την ανάγκη συντονιστή ή κεντρικής αρχής. Ως αποτέλεσμα, κάθε χρήστης δεν χρησιμοποιεί πλέον απλώς κάτι, αλλά και το κατέχει. Λάβετε υπόψη ότι το Web 3.0 δεν είναι ακόμη πλήρως λειτουργικό. Ωστόσο, στοιχεία του Web 3.0 όπως τα NFT, το Blockchain, τα κατανεμημένα λογιστικά βιβλία και το σύννεφο AR έχουν ήδη βρει το δρόμο τους στον τρόπο χρήσης του Διαδικτύου. Επιπλέον, η τεχνολογία Web 3.0 περιλαμβάνει το Internet of Things και το Siri. Από την άλλη πλευρά, εάν και όταν πραγματοποιηθεί η πλήρης υλοποίηση, θα είναι περισσότερο σύμφωνη με το σχέδιο του αρχικού Lee της Berners για το Web 3.0. Θα είναι μια τοποθεσία όπου «δεν απαιτείται έγκριση από μια κεντρική αρχή για να αναρτηθεί οτιδήποτε», ισχυρίζεται. Δυστυχώς, υπάρχει ακόμη περισσότερη δουλειά που πρέπει να γίνει, ιδιαίτερα στον τομέα της αναγνώρισης ομιλίας, επειδή η ανθρώπινη ομιλία έχει μια εκπληκτική σειρά από λεπτές εκφράσεις και ορολογία που η τεχνολογία δεν μπορεί να κατανοήσει πλήρως. Αν και υπήρξαν βελτιώσεις, η διαδικασία δεν έχει ακόμη κατακτηθεί. [3]

1.5 Ομοιότητες ανάμεσα σε web 1, web 2, web 3

Εάν εξεταστεί προσεκτικά καθεμία από τις τρεις εκδόσεις ιστού, παρατηρείτε ότι μοιράζονται μόνο έναν μικρό αριθμό βασικών χαρακτηριστικών. Όλα εξετάζουν τον τρόπο με τον οποίο αλληλεπιδρούν

οι χρήστες και οι πληροφορίες και όλα παρέχουν στους χρήστες πρόσβαση στη λειτουργία "ανάγνωσης". Για να ολοκληρώσουν τις ευθύνες τους πιο γρήγορα, βασίζονται όλοι στο Διαδίκτυο. [3]

1.6 Δυνατότητες του web 1, web 2, web 3

Παρακάτω αναφέρονται οι κύριες δυνατότητες της κάθε έκδοσης του διαδικτύου. Αρχικά το Web 1.0 δεν πρόσφερε καμία αλληλεπίδραση μεταξύ χρήστη και διακομιστή. Οι ιστοσελίδες ήταν αδρανείς χωρίς αλληλεπίδραση και πρόσφερε μόνο προβολή περιεχομένου. Επίσης πρόσφερε την δυνατότητα σελιδοδεικτών και δημιουργία συνδέσμων σε σελίδες. Στη συνέχεια το Web 2.0 είχε βελτιωμένη αλληλεπίδραση με τον χρήστη σε σύγκριση με τις εφαρμογές Web 1.0. Χρησιμοποιεί λειτουργίες όπως ροή βίντεο και ηλεκτρονικά έγγραφα, για παράδειγμα. Τα πάντα είναι online και οι διακομιστές είναι εκεί όπου διατηρούνται οι εφαρμογές και οι πληροφορίες. Υπάρχουν υπηρεσίες πληρωμής με το <<κλικ>> όπως και διαδραστικές διαφημίσεις υπολογιστικού νέφους αλλά και συγκεντρωτικά δεδομένα που υποστηρίζουν ανάγνωση και γραφή στο διαδίκτυο. Τέλος το Web 3.0 υλοποιεί δυνατότητες όπως έξυπνες λειτουργίες και εφαρμογές που βασίζονται στο Web με διασκορπισμένες διαδικασίες. Αναπαράσταση γνώσης σε συνδυασμό με προσωποποιημένο μάρκετινγκ προς το χρήστη. Υποστηρίζει ζωντανό βίντεο όπως και τεχνολογίες του διαδικτύου των πραγμάτων (IoT). [3]

1.7 Web 1.0 vs. Web 2.0: Μεταξύ τους Σύγκριση

Το «παλιό» Διαδίκτυο (Web 1.0) είναι αναμφίβολα γνωστό στους έμπειρους χρήστες του Διαδικτύου και δεδομένου ότι το Web 2.0 είναι πλέον το συνήθη πρότυπο, όλοι είχαν κάποια αλληλεπίδραση μαζί του. Αν και το Web 3.0 υπάρχει σε κομμάτια, δεν έχει ακόμη εφαρμοστεί πλήρως. Οι πληροφορίες στο Web 1.0 δεν μπορούν να αλλάξουν, ενώ οι πληροφορίες στο Web 2.0 μπορούν. Το Web 2.0 είναι ένας δυναμικός Ιστός που περιέχει μη γραμμικές πληροφορίες, ενώ το Web 1.0 είναι ένας στατικός Ιστός με γραμμικές πληροφορίες. Το κείμενο που πρέπει να διαβαστεί με κλασικό ευθύγραμμο τρόπο από την αρχή μέχρι το τέλος αναφέρεται ως γραμμικές πληροφορίες. Αντίθετα, το μη γραμμικό περιεχόμενο δεν έχει τέτοιους περιορισμούς και μπορεί να διαβαστεί με όποια σειρά επιλέξει ο αναγνώστης. Το Web 2.0 είναι μια δυναμική οντότητα, σε αντίθεση με το στατικό Web 1.0. [3]

1.8 Επίλογος

Κλείνοντας το web 4 είναι ήδη στα σκαριά. Υπάρχουν πολλές εικασίες, και ορισμένοι ισχυρίζονται ότι θα είναι πιο διανοητικές και θα αντιμετωπίσει τα προβλήματα με την αποκέντρωση που δημιουργεί το Web 3.0. Εάν η αποκέντρωση εφαρμοστεί ευρέως, θα χρειαστεί να τροποποιηθεί εκτενώς επειδή δεν είναι ιδανική. Ορισμένοι ειδικοί προβλέπουν ακόμη ότι το Web 4.0, στο οποίο οι άνθρωποι έχουν πρόσβαση στον Ιστό μέσω φυσικών εμφυτευμάτων, θα είναι η κορυφή της εξέλιξης του Ιστού. Ανάλογα με την άποψη του καθενός, αυτή είναι είτε μια φανταστική ιδέα είτε ένας δυστοπικός εφιάλτης που γίνεται πραγματικότητα. Ήδη έχουμε φορητή τεχνολογία, όπως έξυπνα ρολόγια ή οθόνες καρδιάς που τροφοδοτούν δεδομένα στον πάροχο πρωτοβάθμιας φροντίδας του ασθενούς, για όποιον πιστεύει ότι αυτή η ιδέα είναι πολύ μακριά στον κόσμο της επιστημονικής φαντασίας. Η μετάβαση σε μια εμφυτεύσιμη συσκευή που επιτρέπει την πρόσβαση στον Ιστό κατά παραγγελία και καταργεί την απαίτηση για φορητή συσκευή χειρός δεν είναι ιδιαίτερα σημαντική. Παρόλα αυτά, το Web 4.0 απέχει ακόμα δεκαετίες, ανεξάρτητα από την εμφάνισή του. Ο κλάδος της πληροφορικής επικεντρώνεται επί του παρόντος στην πλήρη εφαρμογή του Web 3.0. [3]

Κεφάλαιο 2ο : Web Scrapping

2.1 Εισαγωγή

Το web scraping είναι η διαδικασία χρησιμοποίησης κατάλληλων bot – αλγορίθμων για την εξαγωγή περιεχομένου και δεδομένων από έναν ιστότοπο. Το web scraping εξάγει τον κώδικα HTML και μαζί του τα δεδομένα που είναι αποθηκευμένα σε μια βάση δεδομένων.

2.2 Τι είναι βιβλιοθήκη

Η βιβλιοθήκη είναι μια συλλογή από καλά ορισμένες κλάσεις και μεθόδους και προορίζονται να εκτελούν συγκεκριμένες λειτουργίες. Στην ουσία είναι επαναχρησιμοποίηση κώδικα που έχει γραφτεί από άλλους προγραμματιστές. Συνήθως ορίζουν συγκεκριμένες λειτουργίες σε συγκεκριμένο τομέα (πχ βιβλιοθήκες μαθηματικών επιτρέπουν στον προγραμματιστή απλά να καλέσει μια συνάρτηση χωρίς να επαναλάβει τη λειτουργία του αλγορίθμου). Μια βιβλιοθήκη εστιάζει συνήθως σε ένα κομμάτι λειτουργικότητας στο οποίο έχεις πρόσβαση μέσω κάποιου API. [4]

2.3 Τι είναι Framework

Ένα framework είναι πιο περίπλοκο συνήθως από μια βιβλιοθήκη και μπορεί να ενσωματώνει πολλές βιβλιοθήκες. Συνήθως αποτελεί στον σκελετό για το πρόγραμμα μας πάνω στο οποίο υπάρχουν προκαθορισμένα σημεία στα οποία μπορούμε να ενσωματώσουμε τον κώδικα μας έτσι ο προγραμματιστής δεν ασχολείται τοσο με τη σχεδίαση . Διευκολύνει πολύ την ανάπτυξη λογισμικού καθώς έχει ήδη υλοποιημένες πολλές λειτουργίες , κατάλληλη ένδειξη λαθών όπως και επιτρέπει την εύκολη συνεργατική ανάπτυξη κώδικα. [4]

2.4 Τι είναι Web Scrapping

Το web scraping είναι μια αυτοματοποιημένη τεχνική για τη συλλογή άφθονων όγκων δεδομένων από ιστότοπους. Τα περισσότερα από αυτά τα δεδομένα δεν είναι δομημένα σε μορφή HTML και μετατρέπονται σε δομημένα δεδομένα σε μια βάση δεδομένων ή υπολογιστικό φύλλο, ώστε να μπορούν να χρησιμοποιηθούν σε πολλαπλές εφαρμογές. Η συλλογή δεδομένων από ιστότοπους μπορεί να γίνει με ποικίλες μεθόδους. Αυτά περιλαμβάνουν τη χρησιμοποίηση συγκεκριμένων API, διαδικτυακές υπηρεσίες ή ακόμα και τη σύνταξη δικού μας κώδικα από την αρχή για να επιτευχθεί αυτή η διαδικασία. Μπορεί να αποκτηθεί πρόσβαση στα δομημένα δεδομένα σε πολλούς δημοφιλής ιστότοπους, χρησιμοποιώντας τα API τους. Αυτή είναι η καλύτερη επιλογή, ωστόσο υπάρχουν εναλλακτικοί ιστότοποι που είτε δεν διαθέτουν την τεχνολογική πολυπλοκότητα είτε δεν επιτρέπουν στους χρήστες να έχουν πρόσβαση σε σημαντικό όγκο δομημένων δεδομένων. Σε αυτήν την περίπτωση, συνιστάται η χρήση web scraping για τη συλλογή δεδομένων από τον ιστότοπο. Ο scraper είναι το εργαλείο που χρειάζεται για την συλλογή των δεδομένων ιστού. Ο scraper είναι ένα μοναδικό εργαλείο που έχει σχεδιαστεί για την εξαγωγή δεδομένων από έναν ιστότοπο. Η αρχιτεκτονική του scraper μπορεί να ποικίλλει σημαντικά ανάλογα με τη δυσκολία και το μέγεθος του έργου προκειμένου να εξαχθούν αποτελεσματικά και με ακρίβεια τα δεδομένα. [5]

2.5 Πως λειτουργεί ένας scraper

Οι web scrapers μπορούν να συλλέξουν όλες τις πληροφορίες από συγκεκριμένους ιστότοπους ή τις συγκεκριμένες πληροφορίες που ζητά ένας χρήστης. Είναι ιδανικό εάν περιγράφετε τα δεδομένα που

χρειάζεστε, έτσι ώστε ο web scraper να ανακτά μόνος γρήγορα αυτές τις πληροφορίες. Για παράδειγμα, μπορεί να θέλουμε να συλλέξουμε δεδομένα έναν ιστότοπο τύπου eshop για να μάθουμε ποια είδη αποχυμωτών προσφέρονται, αλλά μπορεί να χρειαζόμαστε μόνο πληροφορίες για τα μοντέλα των διαφόρων αποχυμωτών και όχι τα σχόλια από τους πελάτες. Επομένως, χρειάζονται οι διευθύνσεις URL για αρχή όταν ένας web scraper χρειάζεται να συλλέξει δεδομένα από έναν ιστότοπο. Στη συνέχεια, φορτώνεται όλος ο κώδικας HTML των ιστοτόπων. Ένα πιο εξελιγμένο scraper μπορεί επίσης να εξαγάγει όλα τα τμήματα CSS και Javascript και μετέπειτα να συλλέξει τα δεδομένα που χρειάζεται. Στη συνέχεια, το scraper εξάγει τα απαραίτητα δεδομένα από αυτόν τον κώδικα HTML και τα εξάγει με τον τρόπο που έχει επιλέξει ο χρήστης. Τα δεδομένα αποθηκεύονται συνήθως ως υπολογιστικό φύλλο Excel ή αρχείο CSV, αλλά είναι επίσης δυνατή η αποθήκευση τους σε άλλες μορφές, όπως ένα αρχείο JSON ή και απευθείας σε μια σχεσιακή βάση δεδομένων. [5]

2.6 Html Parsing vs Render

Υπάρχουν βιβλιοθήκες για web scrapping οι οποίες άλλες κάνουν απλό html parsing και άλλες render. Η διαφορά του parsing από το render είναι ότι το ένα απλά κατεβάζει το πηγαίο κώδικα της σελίδας όπως είναι ενώ το render εκτελεί και όλα τα scripts της σελίδας πριν πάρει το περιεχόμενο. Το parsing είναι πιο γρήγορο αλλά σε δυναμικές ιστοσελίδες που το περιεχόμενο δημιουργείτε με JavaScript δεν θα μας φέρει δεδομένα ενώ το render μπορεί να είναι λίγο πιο αργό αλλά σε αυτή τη περίπτωση μας εξυπηρετεί καλύτερα.

Η python λόγω της εύκολης και γρήγορης εκμάθησης καθώς και της ταχύτητάς της μπορεί να θεωρηθεί κατάλληλή για web scrapping.

Οι πιο δημοφιλείς βιβλιοθήκες και framework για web scrapping σε python είναι:

- Scrapy – framework
- Selenium – library
- Beautiful Soup – library

2.7 Διαφορετικοί τύποι Scrapers

Οι web scrapers μπορούν να ταξινομηθούν με βάση διάφορους παράγοντες, συμπεριλαμβανομένου του εάν είναι αυτο-κατασκευασμένοι ή προκατασκευασμένοι, εάν είναι επεκτάσεις λογισμικού ή προγράμματος περιήγησης και εάν είναι τοπικά ή στο cloud. Οι αυτο-κατασκευασμένοι web scrappers είναι δυνατοί, αλλά απαιτούν υψηλό επίπεδο τεχνογνωσίας προγραμματισμού. Επιπλέον, χρειάζεται ακόμη περισσότερη κατανόηση εάν θέλετε ο web scraper να έχει περισσότερες δυνατότητες. Οι έτοιμοι web scrappers, από την άλλη πλευρά, είναι έτοιμοι αλγόριθμοι που έχουν ήδη κατασκευαστεί και είναι απλοί στη χρήση τους. Μπορείτε να τα παραμετροποιηθούν εύκολα αλλά και να προστεθούν και πιο εξελιγμένες επιλογές. Ενναλακτικά να μπουν πρόσθετα στον περιηγητή ιστού σαν επεκτάσεις που ονομάζονται web scrapers. Οι web scrapers που εκτελούνται σε επεκτάσεις προγράμματος περιήγησης δεν μπορούν να χρησιμοποιήσουν εξελιγμένες λειτουργίες που υπερβαίνουν τις δυνατότητες του προγράμματος περιήγησής μας. Οι scrapers που τρέχουν στο cloud προσφέρονται συνήθως σε επιχειρήσεις. Το cloud είναι ένας διακομιστής εκτός τοποθεσίας. Δεδομένου ότι δεν απαιτούν συλλογή δεδομένων από ιστότοπους, ο υπολογιστής μας μπορεί να επικεντρωθεί σε άλλες δραστηριότητες. Από την άλλη πλευρά, οι scrapers που τρέχουν τοπικά χρησιμοποιούν τους πόρους του υπολογιστή μας για

να λειτουργήσουν. Επομένως, εάν οι web scrapers χρειάζονται περισσότερη CPU ή RAM, ο υπολογιστής σας θα επιβραδυνθεί και θα γίνει ανίκανος να χειριστεί άλλες εργασίες. [5]

2.8 Γιατί η python είναι δημοφιλή για web scrapping?

Στις μέρες μας, η γλώσσα python φαίνεται να είναι αρκετά δημοφιλή. Είναι η πιο ευρέως χρησιμοποιούμενη γλώσσα για συλλογή δεδομένων ιστού, καθώς μπορεί εύκολα να χειριστεί τις περισσότερες διαδικασίες. Επιπλέον, προσφέρει μια σειρά από βιβλιοθήκες που έχουν σχεδιαστεί ρητά για web scrapping. Το Scrapy που βασίζεται σε Python είναι ένα αρκετά γνωστό framework ανοιχτού κώδικα. Είναι τέλειο τόσο για εξαγωγή δεδομένων βάσει API όσο και για συλλογή δεδομένων ιστού. Μια άλλη λειτουργική μονάδα της Python που είναι εξαιρετική για συλλογή δεδομένων ιστού ονομάζεται beautiful soup. Για την εξαγωγή δεδομένων από HTML σε έναν ιστότοπο, παράγει ένα δέντρο ανάλυσης. Αυτά τα δέντρα ανάλυσης μπορούν να πλοηγηθούν, να αναζητηθούν και να τροποποιηθούν χρησιμοποιώντας μια ποικιλία δυνατοτήτων στο Beautiful Soup. [5]

2.9 Σύγκριση Διαφορετικών Τεχνολογιών

2.9.1 Scrapy

Το scrapy είναι ένα πολύ δημοφιλή framework το οποίο διακρίνετε για την ταχύτητα του διότι καταναλώνει λίγη επεξεργαστική ισχύ και μνήμη. Εξ αρχής δεν κάνει render αλλά απλό html parsing τις σελίδες για να εξάγει γρήγορα τα δεδομένα, βέβαια με κατάλληλες παραμετροποιήσεις μπορεί να γίνει. Μπορεί να κάνει ασύγχρονα requests σε σελίδες όπως και parsing διότι έχει ήδη έτοιμες μεθόδους. Μας έχει από την αρχή ήδη έτοιμο το σκελετό της εφαρμογής μας και πολλές μεθόδους ήδη υλοποιημένες (πχ. Δυνατότητα παράλληλης επεξεργασίας του προγράμματος , αποθήκευση σε βάσεις δεδομένων όπως και επιλογή έτοιμων αλγορίθμων για την αναζήτηση στις ιστοσελίδες).Επίσης είναι εύκολα επεκτάσιμο. [6]

2.9.2 Beautiful Soup

Το beautiful soup είναι και αυτό μια βιβλιοθήκη είναι επίσης γρήγορη και πολύ εύκολη στην εκμάθηση και χρήση, μας παρέχει τα λιγότερα από τα άλλα δύο, μόνο το πολύ απαραίτητα για την εξαγωγή δεδομένων από τις σελίδες. Δεν διαθέτει έτοιμες μεθόδους για requests σε σελίδες , ούτε για html parsing ή rendering. Συνεπώς χρειάζεται εξωτερικές βιβλιοθήκες για την κανονική λειτουργεία του. Το πλεονέκτημα είναι ότι το χτίζεις εσύ όπως θες και με τις λειτουργίες που θες και έτσι έχεις τον πλήρη έλεγχο. Μπορεί να κάνει είτε render τις σελίδες είτε απλό html parsing με τις κατάλληλες βιβλιοθήκες. [6]

2.10 Εφαρμογές

Υπάρχουν πολλές χρήσεις για την συλλογή δεδομένων ιστού σε διαφορετικές βιομηχανίες. Μια από αυτές είναι η παρακολούθηση τιμολόγησης. Οι επιχειρήσεις μπορούν να χρησιμοποιήσουν το web scraping για να συλλέξουν πληροφορίες προϊόντων τόσο για τα δικά τους όσο και για παρόμοια προϊόντα, προκειμένου να αξιολογήσουν πώς επηρεάζει τη στρατηγική τιμολόγησης τους. Οι εταιρείες μπορούν να χρησιμοποιήσουν αυτές τις πληροφορίες για να καθορίσουν την καλύτερη τιμή για τα αντικείμενά τους, προκειμένου να έχουν το μεγαλύτερο εισόδημα. Επίσης το web scraping βρίσκει εφαρμογή στην ανάλυση της αγοράς. Οι εταιρείες μπορούν να χρησιμοποιήσουν το web scraping για

έρευνα αγοράς. Μεγάλοι όγκοι δεδομένων υψηλής ποιότητας που έχουν αφαιρεθεί από τον ιστό μπορεί να είναι αρκετά επωφελής για τις επιχειρήσεις όσον αφορά την αξιολόγηση των προτύπων πελατών και τον προσδιορισμό της κατεύθυνσης που πρέπει να ακολουθήσει η εταιρεία στο μέλλον. Μία άλλη εφαρμογή είναι η παρατήρηση ειδήσεων. Μια εταιρεία μπορεί να λαμβάνει λεπτομερείς αναφορές για τις πιο πρόσφατες ειδήσεις από ειδησεογραφικούς ιστότοπους. Για τις επιχειρήσεις που κάνουν συχνά τις ειδήσεις ή των οποίων οι καθημερινές δραστηριότητες εξαρτώνται από τις ειδήσεις, αυτό είναι ακόμη πιο σημαντικό. Ακόμα μπορεί να βρει εφαρμογή στην συναισθηματική αξιολόγηση. Η ανάλυση συναισθήματος είναι απαραίτητη εάν οι επιχειρήσεις επιθυμούν να κατανοήσουν πώς νιώθουν οι πελάτες για τα προϊόντα τους γενικά. Οι επιχειρήσεις μπορούν να χρησιμοποιήσουν το web scraping για να συγκεντρώσουν πληροφορίες σχετικά με τη γενική γνώμη που έχουν οι άνθρωποι για τα προϊόντα τους από πλατφόρμες κοινωνικών μέσων. Αυτό θα τους επιτρέψει να αναπτύξουν προϊόντα που θέλουν οι καταναλωτές και να ξεπεράσουν τους αντιπάλους τους. Τέλος σημαντικό να αναφερθεί η χρήση του στο τομέα του email marketing το οποίο χρησιμοποιείται καθημερινά. Το web scraping είναι ένα εργαλείο που μπορούν να χρησιμοποιήσουν οι επιχειρήσεις για το email marketing. Χρησιμοποιώντας το web scraping, μπορούν να συλλέξουν αναγνωριστικά email από πολλούς ιστότοπους και να στείλουν μαζικά διαφημιστικά μηνύματα και μηνύματα μάρκετινγκ σε όλους όσους έχουν ένα από αυτά τα αναγνωριστικά email. [5]

2.11 Selenium

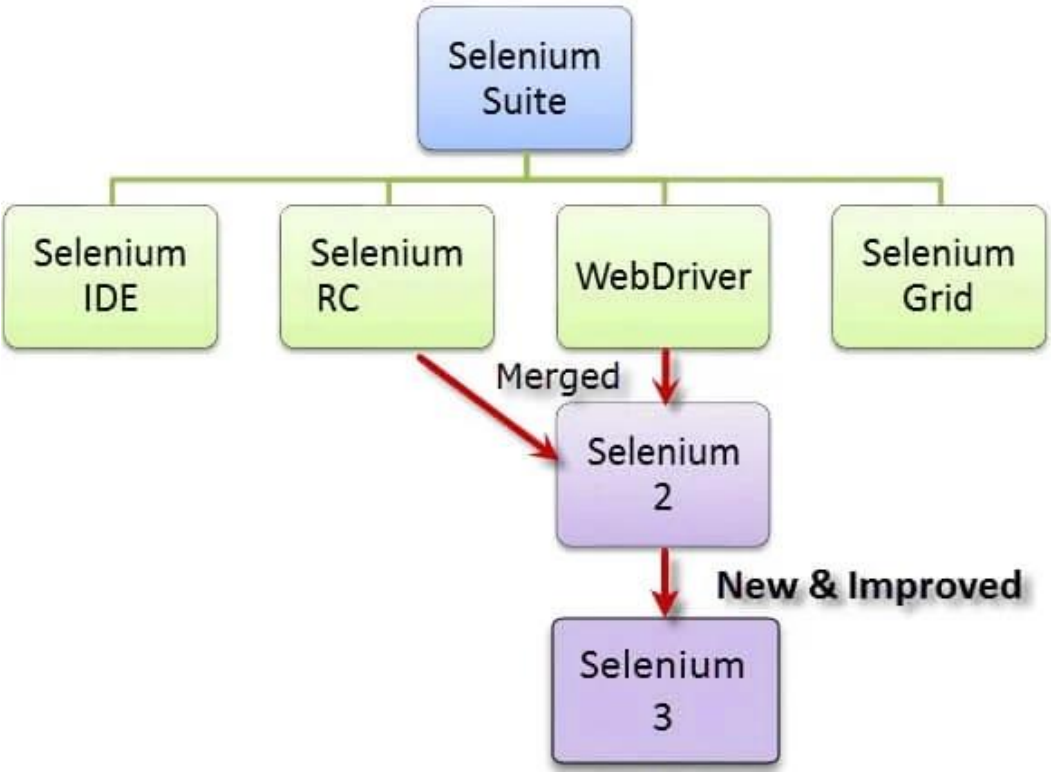
2.11.1 Εισαγωγή

Το selenium είναι μια διάσημη βιβλιοθήκη η οποία χρησιμοποιείται για web scrapping αλλά ειδικεύεται στους αυτοματισμούς του browser. Μπορεί να κάνει requests σε σελίδες όπως και parsing – rendering διότι έχει ήδη έτοιμες μεθόδους για αυτή τη δουλειά. Είναι σε θέση να κάνει render τις σελίδες ανοίγοντας κανονικά τον browser (χρειάζεται ειδικός driver για κάθε browser) και να κάνει αυτοματισμούς , κινήσεις που θα έκανε και ένας άνθρωπος από μόνος τους (πχ. υποβολή φορμών , κλικ σε κουμπιά κλπ.). Συνεπώς είναι πιο αργό από το scrapy και δεν μας παρέχει κάποιο σκελετό και πρέπει ο προγραμματιστής να ξεκινήσει από την αρχή. Επίσης λόγω της πολυπλοκότητας του render καταναλώνει περισσότερη επεξεργαστική ισχύ και μνήμη. [6]

2.11.2 Περισσότερα για το Selenium

Το selenium μπορεί να χρησιμοποιηθεί με διάφορες γλώσσες προγραμματισμού όπως Java, C#, Python κ.λπ. Επίσης το selenium είναι ανοιχτού κώδικα που σημαίνει ότι είναι ελεύθερο προς χρήση. Στη προσπάθεια το selenium να είναι ένα ενιαίο εργαλείο web scrapping και αυτοματισμού δημιουργήθηκε το selenium IDE που είναι ένα ανεξάρτητο εργαλείο που μπορεί να χρησιμοποιηθεί για όλες τις δουλειές αυτού του είδους μέσα από μια ενιαία πλατφόρμα. Εκτός από το ότι είναι ένα ενιαίο εργαλείο, το Selenium Software είναι μια συλλογή προγραμμάτων που το καθένα ανταποκρίνεται στις ανάγκες ενός συγκεκριμένου οργανισμού για δοκιμές ποιότητας. Τα εργαλεία παρατίθενται παρακάτω. [7]

- Integrated Development Environment for Selenium (IDE)
- Remote control for selenium (RC)
- Remote control for selenium (RC)



Εικόνα 1 - Selenium IDE

2.11.3 Σύντομη ανάλυση του Selenium IDE

Το απλούστερο και πιο φιλικό προς τον χρήστη περιβάλλον είναι το Selenium Integrated Development Environment (IDE). Όπως και άλλα πρόσθετα, είναι ένα πρόσθετο Firefox που είναι εύκολο στη ρύθμιση. Το Selenium IDE θα πρέπει να χρησιμοποιείται μόνο ως εργαλείο δημιουργίας πρωτοτύπων, λόγω της απλότητάς του. Πρέπει να χρησιμοποιηθεί είτε το Selenium RC είτε το WebDriver εάν θέλετε να αναπτύξετε πιο περίπλοκα σενάρια δοκιμών. [7]

Θετικά	Αρνητικά
Πολύ εύκολο στην εγκατάσταση	Είναι συμβατό μόνο με Firefox
Δεν χρειάζεται εξοικείωση με τον προγραμματισμό και γνώση της html και του Dom	Σχεδιάστηκε για την ανάπτυξη πρωτότυπων τεστ
Μπορεί να εξάγει τα τεστ σε μορφή κατάλληλη για χρήση με το Selenium RC και το WebDriver	Δεν υποστηρίζει λογικούς ελέγχους και επαναλήψεις
Ενσωματωμένη λειτουργεία βοηθείας και σύστημα αναφοράς των τεστ	Η εκτέλεση των test είναι αργή σε σχέση με το Selenium RC και το WebDriver

Προσφέρει υποστήριξη για επεκτάσεις	

Πίνακας 1 - Selenium IDE Πλεονεκτήματα και Μειονεκτήματα

2.11.4 Σύντομη ανάλυση του Selenium RC

Για ένα διάστημα, το Selenium RC χρησίμευσε ως το κύριο πλαίσιο δοκιμών του έργου Selenium. Αυτό ήταν το πρώτο αυτοματοποιημένο εργαλείο δοκιμών ιστού που επέτρεπε στους χρήστες να επιλέξουν την προτιμώμενη γλώσσα προγραμματισμού. Το RC μπορεί επί του παρόντος να χειριστεί τις ακόλουθες γλώσσες προγραμματισμού: [7]

- Java
- C#
- PHP
- Python
- Perl
- Ruby

Θετικά	Αρνητικά
Συμβατό με όλους τους browser και λειτουργικά συστήματα	Η εγκατάσταση είναι πιο περίπλοκη από τον IDE
Υποστηρίζει λογικούς ελέγχους και επαναλήψεις	Πρέπει να υπάρχουν γνώσεις προγραμματισμού
Πιο γρήγορη εκτέλεση από το Selenium IDE	Πιο αργή εκτέλεση από το Selenium Web Driver
Έχει ολοκληρωμένη διεπαφή (API)	Λιγότερο ρεαλιστική η αλληλεπίδραση με τον browser

Πίνακας 2 - Selenium Rc Θετικά και αρνητικά

2.11.5 Σύντομη ανάλυση του Selenium WebDriver

Με πολλούς τρόπους, το Selenium WebDriver δείχνει γιατί είναι ανώτερο από το Selenium IDE και το Selenium RC. Χρησιμοποιεί μια πιο ενημερωμένη και αξιόπιστη μέθοδο αυτοματοποίησης των ενεργειών του προγράμματος περιήγησης. Το WebDriver δεν βασίζεται σε JavaScript για δοκιμές αυτοματισμού όπως το Selenium RC. Μιλώντας απευθείας με το πρόγραμμα περιήγησης, μπορεί να το δώσει εντολές. [7]

Θετικά	Αρνητικά
Απλούστερη εγκατάσταση από το Selenium Rc	Πιο δύσκολη εγκατάσταση από το Selenium IDE

Επικοινωνεί κατευθείαν με τον browser	Χρειάζεται γνώσεις προγραμματισμού
Αλληλεπίδραση με τον browser πιο ρεαλιστική	Δεν υποστηρίζει τους καινούργιους browser
Δεν χεριάζετε έξτρα εγκατάσταση προγράμματος όπως ο RC Server	Κατά την εκτέλεση δεν παράγει μηνύματα λάθους

Πίνακας 3 - Selenium WebDriver Θετικά και Αρνητικά

2.11.6 Selenium Grid

Το selenium grid χρησιμοποιείται σε συνδυασμό με το selenium rc για να εκτελούν παράλληλα τεστ πάνω σε διαφορετικούς υπολογιστές και περιηγητές ιστού την ίδια χρονική στιγμή. Παράλληλη εκτέλεση σημαίνει πολλά τεστ με την μία. Οι δυνατότητες που προσφέρει είναι: [7]

- Εκτέλεση παράλληλων τεστ σε πολλαπλούς περιηγητές και περιβάλλοντα
- Μεγάλη μείωση χρόνου
- Χρησιμοποιεί την έννοια κεντρικό σημείο με κόμβους. Το κεντρικό σημείο ενεργεί ως πηγή στις εντολές του selenium σε κάθε κόμβο που συνδέετε σε αυτό

2.12 Επίλογος

Ολόκληρη η σουίτα δοκιμών του selenium αποτελείται από τέσσερα συστατικά:

- Το selenium ide, που είναι το πρόσθετο του Firefox το οποίο μπορεί μόνο να δημιουργεί πολύ απλά τεστ δοκιμών
- Το selenium remote control, γνωστό και ως selenium 1 που ήταν το πρώτο εργαλείο για προγραμματιστές που επέτρεπε τη δημιουργία πολύπλοκων τεστ
- Τον web driver η πιο πρόσφατη τεχνολογία που επιτρέπει τα script δοκιμών να επικοινωνούν κατευθείαν με τον περιηγητή ιστού από το επίπεδο του λειτουργικού συστήματος
- Το selenium grid είναι επίσης ένα εργαλείο το οποίο χρησιμοποιείται μαζί με το selenium rc για να εκτελεί παράλληλα τεστ σε πολλαπλά λειτουργικά συστήματα και περιηγητές ιστού

Το selenium rc και ο web driver ενώθηκαν και έγιναν το selenium 2. [7]

Κεφάλαιο 3ο: Αλγόριθμοι δομών δεδομένων

3.1 Εισαγωγή

Μια ειδικά σχεδιασμένη μορφή για την τακτοποίηση, την επεξεργασία, την πρόσβαση και την αποθήκευση δεδομένων ονομάζεται δομή δεδομένων. Οι δομές δεδομένων διατίθενται τόσο σε απλές όσο και σε σύνθετες μορφές, όλες κατασκευασμένες για να οργανώνουν δεδομένα για μια συγκεκριμένη χρήση. Οι χρήστες βρίσκουν απλό την πρόσβαση στα δεδομένα που χρειάζονται και τα χρησιμοποιούν κατάλληλα χάρη στις δομές δεδομένων. Η οργάνωση των πληροφοριών πλαισιώνεται από δομές δεδομένων με τρόπο που τόσο οι μηχανές όσο και οι άνθρωποι μπορούν να κατανοήσουν καλύτερα. Μπορεί να επιλεγεί ή να δημιουργηθεί μια δομή δεδομένων για την αποθήκευση δεδομένων στην επιστήμη των υπολογιστών και στον προγραμματισμό υπολογιστών με σκοπό τη χρησιμοποίησή τους με διαφορετικές μεθόδους. Μερικές φορές, οι θεμελιώδεις λειτουργίες του αλγορίθμου σχετίζονται στενά με το σχεδιασμό της δομής δεδομένων. Κάθε δομή δεδομένων έχει λεπτομέρειες σχετικά με τις τιμές δεδομένων, τις συνδέσεις μεταξύ των δεδομένων και, σε ορισμένες περιπτώσεις, τις λειτουργίες που μπορούν να χρησιμοποιηθούν στα δεδομένα. Για παράδειγμα, η δομή δεδομένων και οι συνοδευτικές μέθοδοι συνδέονται μεταξύ τους ως μέρος του ορισμού κλάσης σε μια αντικειμενοστραφή γλώσσα προγραμματισμού. Αν και μπορεί να έχουν σχεδιαστεί για να λειτουργούν με τη δομή δεδομένων σε μη αντικειμενοστραφείς γλώσσες, αυτές οι συναρτήσεις δεν θεωρούνται μέρος της δομής δεδομένων. [8]

3.2 Γιατί οι δομές δεδομένων είναι τόσο σημαντικές

Οι τυπικοί βασικοί τύποι δεδομένων των περισσότερων γλωσσών προγραμματισμού υπολογιστών, είναι οι ακέραιοι ή αριθμοί κινητής υποδιαστολής, είναι συνήθως ανεπαρκείς για να μεταφέρουν τη λογική πρόθεση για την επεξεργασία των δεδομένων. Ωστόσο, προκειμένου να διευκολυνθεί η επεξεργασία, οι εφαρμογές που τα χρησιμοποιούν, που χειρίζονται και εξάγουν πληροφορίες πρέπει να γνωρίζουν πώς πρέπει να τακτοποιούνται τα δεδομένα. Οι δομές δεδομένων επιτρέπουν την αποτελεσματική χρήση, και την κοινή χρήση δεδομένων συνδυάζοντας λογικά τα μέρη δεδομένων. Προσφέρουν ένα επίσημο μοντέλο που περιγράφει τη διάταξη των στοιχείων δεδομένων. Οι δομές δεδομένων χρησιμεύουν ως βάση για πιο σύνθετες εφαρμογές. Προκειμένου να δημιουργηθούν, τα στοιχεία δεδομένων συνδυάζονται σε λογικές μονάδες που αντικατοπτρίζουν αφηρημένους τύπους δεδομένων που σχετίζονται με τον αλγόριθμο ή την εφαρμογή. Ένα "όνομα πελάτη", το οποίο αποτελείται από τις συμβολοσειρές χαρακτήρων για "όνομα", "μεσαίο όνομα" και "επώνυμο", είναι μια απεικόνιση ενός αφηρημένου τύπου δεδομένων. Εκτός από τη χρήση δομών δεδομένων, είναι σημαντικό να επιλέξετε την καλύτερη δομή δεδομένων για κάθε δραστηριότητα. Μια κακή επιλογή δομής δεδομένων θα μπορούσε να οδηγήσει σε αργούς χρόνους εκτέλεσης ή σε κώδικα που δεν ανταποκρίνεται. Όταν επιλέγουμε μια δομή δεδομένων, πρέπει να λαμβάνουμε υπόψη τα ακόλουθα πέντε πράγματα: [8]

- Τι είδους δεδομένα θα διατηρηθούν στο αρχείο
- Τι σκοπό θα εξυπηρετούν αυτά τα δεδομένα
- Πού πρέπει να διατηρούνται ή να αποθηκεύονται τα δεδομένα που δημιουργήθηκαν πρόσφατα
- Ποια είναι η πιο αποτελεσματική προσέγγιση για την τακτοποίηση των δεδομένων
- Ποιοι παράγοντες πρέπει να λαμβάνονται υπόψη κατά τη διαχείριση της δέσμευσης μνήμης και αποθήκευσης

3.3 Πως χρησιμοποιούνται οι δομές δεδομένων

Οι φυσικές αναπαραστάσεις των αφηρημένων τύπων δεδομένων συνήθως υλοποιούνται μέσω δομών δεδομένων. Ο σχεδιασμός αποτελεσματικού λογισμικού απαιτεί προσεκτική εξέταση των δομών δεδομένων. Οι προγραμματιστές θα μπορούσαν να δημιουργήσουν τις δικές τους δομές δεδομένων χρησιμοποιώντας πρώιμες γλώσσες προγραμματισμού όπως Fortran, C και C++. Σήμερα, μια μεγάλη ποικιλία δομών δεδομένων ενσωματώνεται σε πολλές γλώσσες προγραμματισμού για να βοηθήσει στην οργάνωση κώδικα και δεδομένων. Για την αποθήκευση και την ανάκτηση δεδομένων, οι κοινές δομές κωδικοποίησης περιλαμβάνουν λίστες και λεξικά Python, πίνακες JavaScript και αντικείμενα. Οι αλγόριθμοι που χρησιμοποιούνται από τους μηχανικούς λογισμικού σχετίζονται στενά με τις δομές δεδομένων με τις οποίες εργάζονται, όπως λίστες, ουρές και αντιστοιχίσεις από το ένα σύνολο τιμών στο άλλο. Αυτή η μέθοδος μπορεί να συνδυαστεί σε πολλές εφαρμογές, όπως η διαχείριση συλλογών εγγραφών σε μια σχεσιακή βάση δεδομένων και η δημιουργία ενός ευρετηρίου αυτών των εγγραφών χρησιμοποιώντας τη δομή δεδομένων δυαδικού δέντρου. Ακολουθούν ορισμένες περιπτώσεις για το πώς χρησιμοποιούνται οι δομές δεδομένων:

- Αποθήκευση δεδομένων. Κατά την παροχή του συνόλου των χαρακτηριστικών και των δομών αντιστοίχισης που θα χρησιμοποιηθούν για την αποθήκευση εγγραφών σε ένα σύστημα διαχείρισης βάσης δεδομένων, οι δομές δεδομένων χρησιμοποιούνται για την αποτελεσματική διατήρηση των δεδομένων.
- Διαχείριση πόρων και υπηρεσιών. Δομές δεδομένων που περιλαμβάνουν συνδεδεμένες λίστες για εκχώρηση μνήμης, διαχείριση καταλόγου αρχείων και δέντρα δομής αρχείων, καθώς και ουρές προγραμματισμού διεργασιών, χρησιμοποιούνται για να επιτρέψουν τους πόρους και τις λειτουργίες του βασικού λειτουργικού συστήματος (OS).
- Μεταφορά δεδομένων. Οι πληροφορίες που μοιράζονται μεταξύ των εφαρμογών, όπως πακέτα TCP/IP, οργανώνονται χρησιμοποιώντας δομές δεδομένων.
- Ταξινόμηση και διάταξη. Τα δυαδικά δέντρα αναζήτησης, που μερικές φορές αναφέρονται ως ταξινομημένο ή ταξινομημένο δυαδικό δέντρο, είναι δομές δεδομένων που προσφέρουν αποτελεσματικούς τρόπους ταξινόμησης πραγμάτων, όπως συμβολοσειρές χαρακτήρων που χρησιμοποιούνται ως ετικέτες. Οι προγραμματιστές μπορούν να ελέγχουν αντικείμενα τακτοποιημένα σε μια δεδομένη προτεραιότητα χρησιμοποιώντας δομές δεδομένων όπως ουρές προτεραιότητας.
- Ευρετήρια. Για την ευρετηρίαση στοιχείων, όπως αυτά που διατηρούνται σε μια βάση δεδομένων, χρησιμοποιούνται ακόμη πιο πολύπλοκες δομές δεδομένων όπως τα B-δέντρα.
- Αναζήτηση. Τα δέντρα B, οι πίνακες κατακερματισμού και τα δυαδικά δέντρα αναζήτησης είναι κοινές τεχνικές που χρησιμοποιούνται για τη δημιουργία ευρετηρίων που επιταχύνουν τη διαδικασία εύρεσης ενός συγκεκριμένου στοιχείου.
- Επεκτασιμότητα. Οι δομές δεδομένων χρησιμοποιούνται από εφαρμογές μεγάλων δεδομένων για την κατανομή και τη διαχείριση της αποθήκευσης δεδομένων σε κατανεμημένους χώρους αποθήκευσης, διασφαλίζοντας απόδοση και επεκτασιμότητα. Για να διευκολυνθεί η αναζήτηση, πολλά περιβάλλοντα προγραμματισμού μεγάλων δεδομένων, όπως το Apache Spark, προσφέρουν δομές δεδομένων που αναπαράγουν τη θεμελιώδη δομή των καταχωρήσεων της βάσης δεδομένων. [8]

3.4 Χαρακτηριστικά των δομών δεδομένων

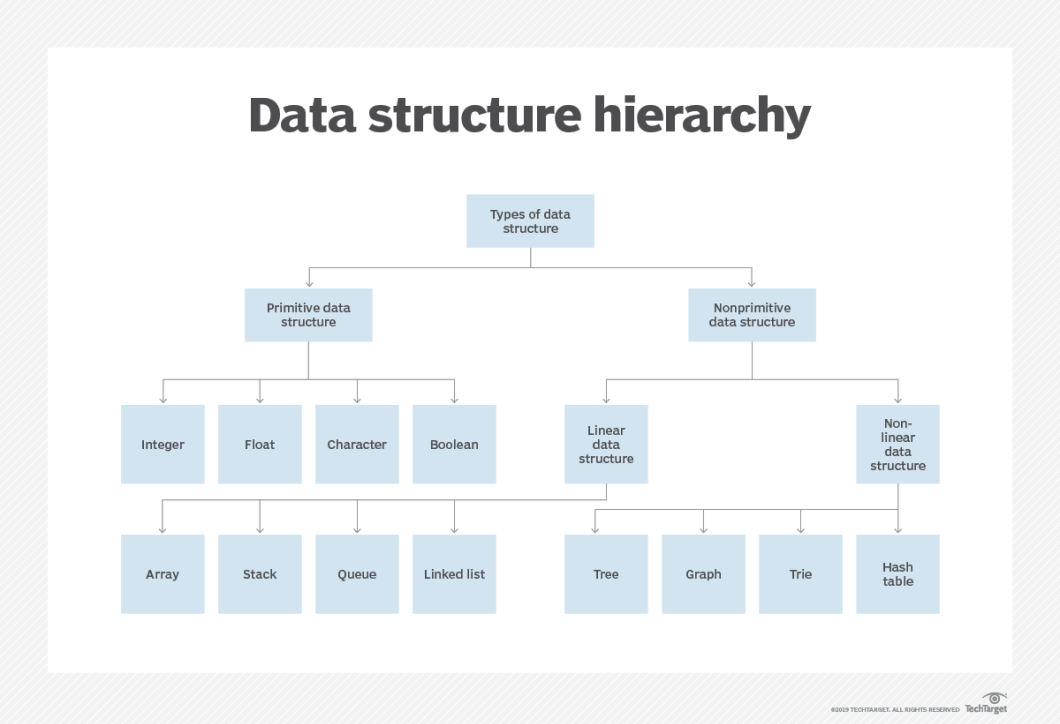
Οι δομές δεδομένων συχνά κατηγοριοποιούνται ανάλογα με τις ιδιότητές τους. Παραδείγματα των τριών πρώτων ιδιοτήτων είναι:

- Γραμμικό και μη γραμμικό. Αυτό το χαρακτηριστικό καθορίζει εάν τα τεμάχια δεδομένων είναι οργανωμένα με μη ταξινομημένη ή διαδοχική σειρά, όπως χρησιμοποιώντας ένα γράφημα ή έναν πίνακα.
- Ομογενής ή ετερογενής. Αυτό το χαρακτηριστικό υποδεικνύει εάν κάθε στοιχείο δεδομένων σε ένα συγκεκριμένο χώρο αποθήκευσης ανήκει στον ίδιο τύπο. Μια σειρά στοιχείων ή μια συλλογή διαφορετικών τύπων, όπως ένας αφηρημένος τύπος δεδομένων που περιγράφεται ως δομή στη C ή μια προδιαγραφή κλάσης στη Java, είναι δύο παραδείγματα.
- Στατικές και Δυναμικές. Αυτό το χαρακτηριστικό εξηγεί τη διαδικασία μεταγλώττισης για τις δομές δεδομένων. Κατά τη στιγμή της μεταγλώττισης, οι στατικές δομές δεδομένων έχουν σταθερά μεγέθη, δομές και θέσεις μνήμης. Ανάλογα με την εφαρμογή τους, τα μεγέθη, οι δομές και οι θέσεις μνήμης των δομών δυναμικών δεδομένων ενδέχεται να αλλάξουν. [8]

3.5 Τύποι δεδομένων

Οι αρχικοί ή βασικοί τύποι δεδομένων είναι τα δομικά στοιχεία των δομών δεδομένων, όπως οι αλγόριθμοι και τα προγράμματα υπολογιστών είναι χτισμένα πάνω σε δομές δεδομένων. Ακολουθούν παραδείγματα κοινών τύπων δεδομένων:

- Boolean είναι ένας λογικός τύπος δεδομένων που μπορεί να αποθηκεύσει μόνο αληθείς ή ψευδείς τιμές.
- Ακέραιος, ο οποίος διατηρεί ένα σύνολο αριθμών μέτρησης ή μαθηματικών ακεραίων.
- Ένας ακέραιος αριθμός 8 bit (integer) μπορεί να αποθηκεύσει τιμές μεταξύ -128 και 127, ενώ ένας μακρύς ακέραιος αριθμός 32 bit (long) μπορεί να αποθηκεύσει τιμές μεταξύ 0 και 4.294.967.295 σε διάφορα μεγέθη.
- Μια τυπική αναπαράσταση πραγματικών αριθμών αποθηκεύεται σε αριθμούς κινητής υποδιαστολής (float, double).
- Χαρακτήρας, ο οποίος χρησιμοποιεί σύμβολα από μια λίστα συμβόλων που αντιστοιχούν σε ακέραιες τιμές.
- Οι τιμές αναφοράς που παραπέμπουν σε άλλες τιμές ονομάζονται δείκτες.
- Χρησιμοποιώντας ένα πεδίο μήκους που περιέχει μια ακέραια τιμή, γίνεται διαχείριση μιας συμβολοσειράς. Μια συμβολοσειρά είναι ένας πίνακας χαρακτήρων που ακολουθείται από έναν κωδικό τερματισμού, ο οποίος είναι συχνά μια τιμή "0". [8]

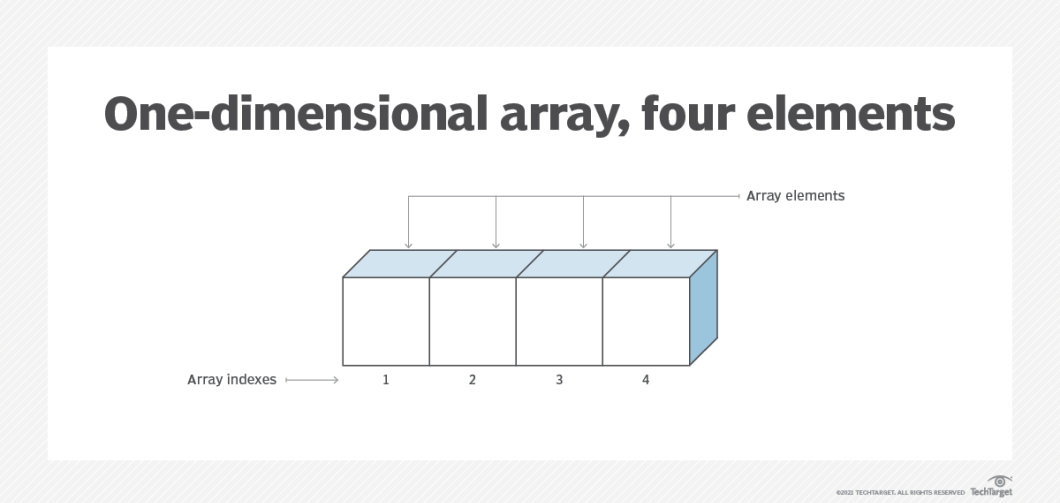


Εικόνα 2 - Ιεραρχία Δομών Δεδομένων

3.6 Τύποι δομών δεδομένων

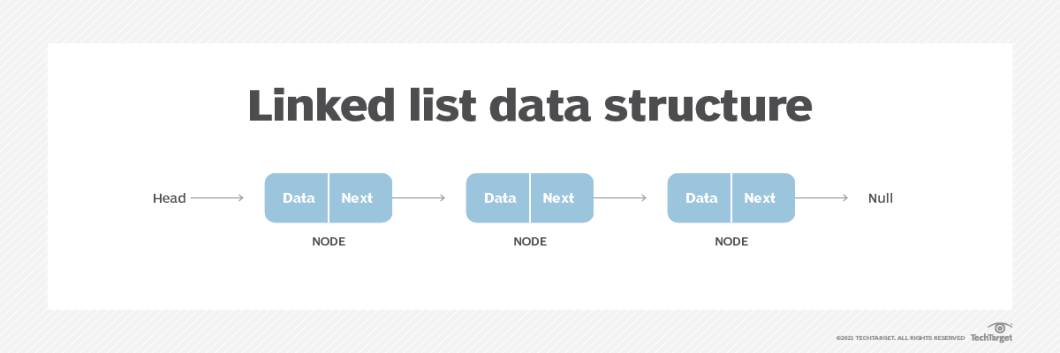
Το είδος των λειτουργιών που θα είναι απαραίτητες ή τα είδη των αλγορίθμων που θα χρησιμοποιηθούν σε μια δεδομένη κατάσταση θα καθορίσουν τον τύπο δομής δεδομένων που θα χρησιμοποιηθεί. Ανάμεσα στα διάφορα είδη δομών δεδομένων είναι τα ακόλουθα:

- Πίνακας. Μια ομάδα πραγμάτων διατηρείται σε μια συστοιχία σε κοντινές θέσεις μνήμης. Ο ίδιος τύπος στοιχείων αποθηκεύονται μαζί, έτσι ώστε ένα ευρετήριο να μπορεί απλώς να προσδιορίζει ή να ανακτά τη θέση κάθε στοιχείου. Οι πίνακες μπορεί να έχουν μεταβλητά ή καθορισμένα μήκη.



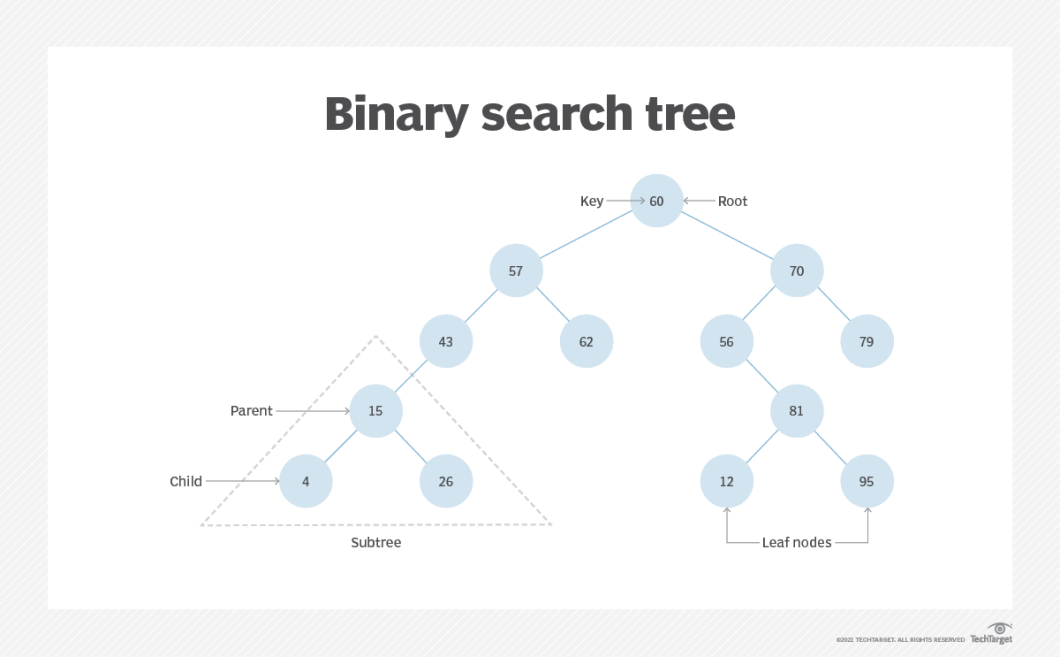
Εικόνα 3- Μονοδιάστατος Πίνακας

- Σωρός. Μια στοίβα οργανώνει μια ομάδα πραγμάτων στη γραμμική ακολουθία με την οποία εκτελούνται οι λειτουργίες. First in, first out (FIFO) ή last in, first out (LIFO) θα μπορούσε να ισχύει για αυτήν τη ρύθμιση (FIFO).
- Ουρά. Παρόμοια με μια στοίβα, μια ουρά διατηρεί μια συλλογή αντικειμένων, αλλά η σειρά των πράξεων είναι πάντα πρώτη μέσα, πρώτη έξω.
- Λίστα σύνδεσης. Μια συνδεδεμένη λίστα διατηρεί μια ομάδα πραγμάτων σε μια λογική σειρά. Τα στοιχεία ή οι κόμβοι μιας συνδεδεμένης λίστας περιέχουν στοιχεία δεδομένων και αναφορές στα επόμενα στοιχεία της λίστας.



Εικόνα 4 - Συνδεδεμένη Λίστα

- Δέντρο. Ένα δέντρο οργανώνει μια ομάδα πραγμάτων σε μια αφηρημένη ιεραρχία. Κάθε κόμβος έχει μια τιμή κλειδιού που του έχει εκχωρηθεί και οι γονικοί κόμβοι συνδέονται με θυγατρικούς κόμβους, γνωστούς και ως υποκόμβους. Κάθε κόμβος στο δέντρο έχει έναν κόμβο ρίζας που χρησιμεύει ως πρόγονός του.

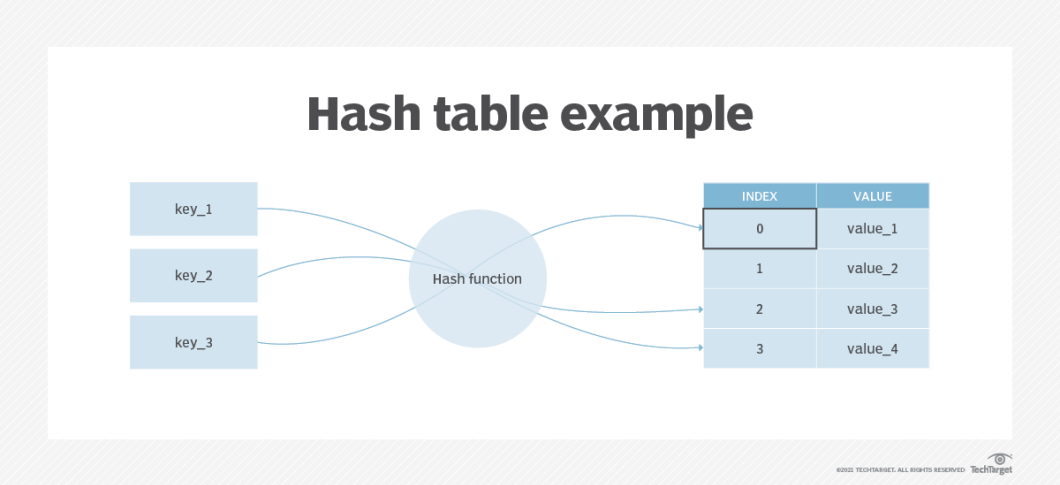


Εικόνα 5 - Δυαδικό Δέντρο

- Σωρός. Σε μία σωρό, η τιμή κλειδιού κάθε γονικού κόμβου είναι μεγαλύτερη ή ίση με κάθε τιμή κλειδιού των παιδιών του. Ένας σωρός είναι μια δομή που βασίζεται σε δέντρα.
- Γραφική παράσταση. Μια συλλογή στοιχείων αποθηκεύεται μη γραμμικά σε ένα γράφημα. Ένας πεπερασμένος αριθμός κόμβων, γνωστοί και ως κορυφές, και οι γραμμές σύνδεσης μεταξύ

τους, γνωστές και ως ακμές, συνθέτουν ένα γράφημα. Αυτά είναι χρήσιμα για την προσομοίωση πραγματικών συστημάτων, όπως τα δίκτυα υπολογιστών.

- Πίνακας κατακερματισμού. Ένας πίνακας κατακερματισμού, που συχνά αναφέρεται ως χάρτης κατακερματισμού, είναι μια δομή δεδομένων που χρησιμοποιείται για την αποθήκευση μιας ποικιλίας πραγμάτων σε έναν συσχετιστικό πίνακα που αντιστοιχίζει κλειδιά σε τιμές. Ένας πίνακας κατακερματισμού μετατρέπει ένα ευρετήριο σε μια σειρά από κουβάδες που περιέχουν το ζητούμενο στοιχείο δεδομένων χρησιμοποιώντας μια συνάρτηση κατακερματισμού.



Εικόνα 6 - Πίνακας Κατακερματισμού

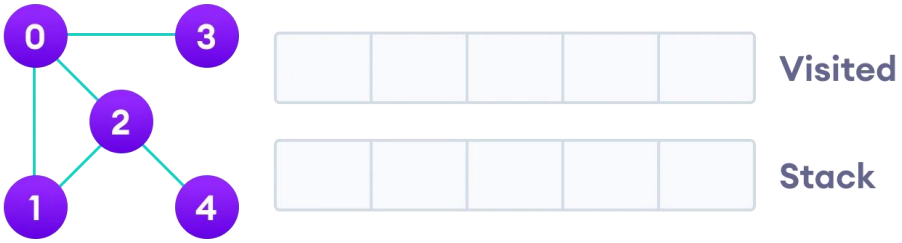
Αυτά μπορεί να περιέχουν σημαντικό όγκο σχετικών δεδομένων, γι' αυτό και θεωρούνται εξελιγμένες δομές δεδομένων. [8]

3.7 Depth First Algorithm

Μια αναδρομική προσέγγιση που ονομάζεται πρώτη αναζήτηση βάθους ή πρώτη διέλευση βάθους χρησιμοποιείται για την αναζήτηση κάθε κορυφής σε ένα γράφημα ή μια δομή δεδομένων δέντρου. Οι κόμβοι ενός γραφήματος επισκέπτονται κατά τη διέλευση του αλγορίθμου. Κάθε κορυφή του γραφήματος εκχωρείται σε μία από τις δύο κατηγορίες σε μια τυπική υλοποίηση Depth First Algorithm:

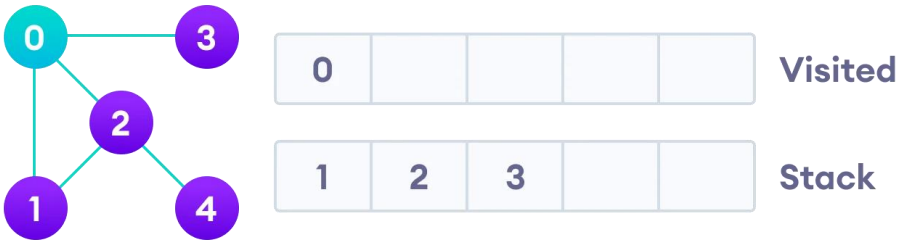
- Επισκέφθηκε
- Δεν επισκέφθηκε

Ο στόχος του αλγόριθμου είναι να αποφεύγει τους κύκλους ενώ επισημαίνει κάθε κορυφή ως επισκέψιμη. Ξεκινάει στοιβάζοντας οποιαδήποτε κορυφή του γραφήματος πάνω από μια άλλη. Προσθέτει το στοιχείο στην κορυφή της στοίβας στη λίστα επισκέψεων. Δημιουργεί μια λίστα με τους κόμβους που βρίσκονται κοντά σε αυτήν την κορυφή. Τοποθετεί τα στοιχεία που δεν έχετε επισκεφτεί στην κορυφή της στοίβας. Επαναλαμβάνει τις ενέργειες 2 και 3 μέχρι να αδειάσει τελείως η στοίβα. Παράδειγμα του αλγορίθμου χρησιμοποιώντας ένα μη κατευθυνόμενο γράφημα με 5 κορυφές.



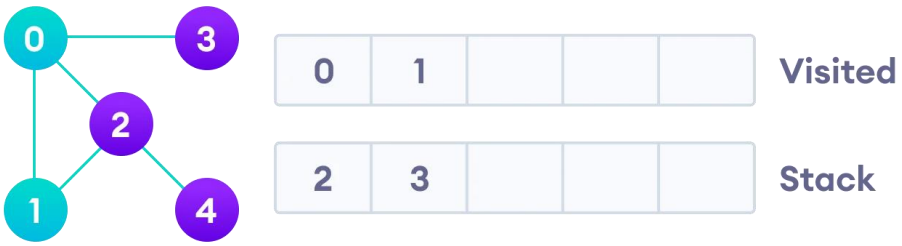
Εικόνα 7 - Παράδειγμα DFS (0)

Ξεκινάμε από την κορυφή 0, ο αλγόριθμος DFS ξεκινά βάζοντάς το στη λίστα Visited και βάζοντας όλες τις γειτονικές κορυφές του στη στοίβα.



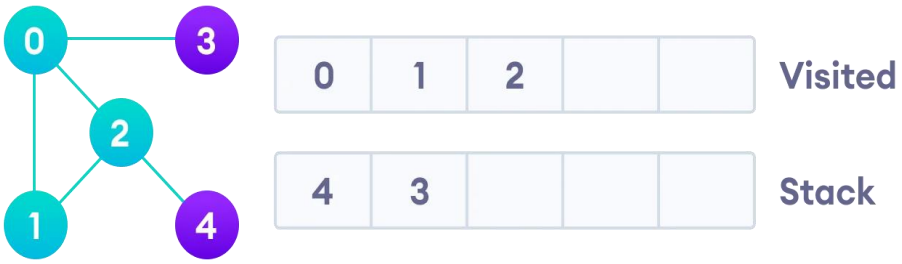
Εικόνα 8 - Παράδειγμα DFS (1)

Στη συνέχεια, επισκεπτόμαστε το στοιχείο στην κορυφή της στοίβας, δηλαδή 1 και πηγαίνουμε στους παρακείμενους κόμβους του. Εφόσον το 0 έχει ήδη επισκεφθεί, επισκεπτόμαστε το 2.

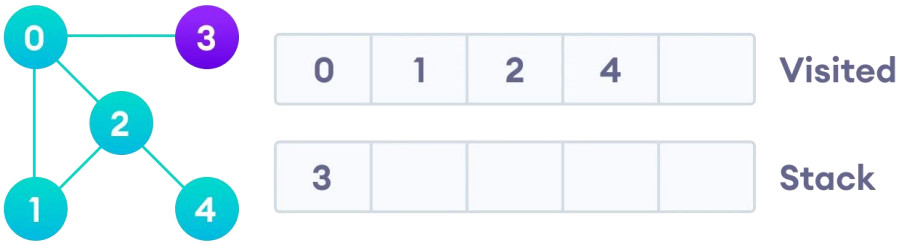


Εικόνα 9 - Παράδειγμα DFS (2)

Ο κόμβος 2 έχει μια μη επισκέψιμη γειτονική κορυφή στο 4, οπότε την προσθέτουμε στην κορυφή της στοίβας και την επισκεπτόμαστε.



Εικόνα 10 - Παράδειγμα DFS (3)



Εικόνα 11 - Παράδειγμα DFS (4)

Αφού επισκεφτούμε το τελευταίο στοιχείο 3, δεν έχει κανέναν μη επισκέψιμο παρακείμενο κόμβο, επομένως ολοκληρώσαμε την πρώτη διέλευση βάθους του γραφήματος. [9]

3.7.1 Ψευδοκώδικας του Depth First Algorithm

Ο ψευδοκώδικας για το DFS φαίνεται παρακάτω. Στη συνάρτηση init(), παρατηρούμε ότι εκτελούμε τη συνάρτηση DFS σε κάθε κόμβο. Αυτό συμβαίνει επειδή το γράφημα μπορεί να έχει δύο διαφορετικά αποσυνδεδεμένα μέρη, οπότε για να βεβαιωθούμε ότι καλύπτει κάθε κορυφή, μπορεί επίσης να εκτελέσουμε τον αλγόριθμο DFS σε κάθε κόμβο. [9]

```
DFS(G, u)
    u.visited = true
    for each v ∈ G.Adj[u]
        if v.visited == false
            DFS(G,v)

init() {
    For each u ∈ G
        u.visited = false
     For each u ∈ G
       DFS(G, u)
}
```

Κώδικας 1 - DFS (1)

3.7.2 Κώδικας Depth First Algorithm σε python

Ο κώδικας για τον αλγόριθμο πρώτης αναζήτησης βάθους με ένα παράδειγμα φαίνεται παρακάτω. Ο κώδικας έχει απλοποιηθεί, ώστε να μπορούμε να εστιάσουμε στον αλγόριθμο και όχι σε άλλες λεπτομέρειες. [9]

```
# DFS algorithm
def dfs(graph, start, visited=None):
    if visited is None:
        visited = set()
    visited.add(start)

    print(start)

    for next in graph[start] - visited:
        dfs(graph, next, visited)
    return visited



graph = {'0': set(['1', '2']),
         '1': set(['0', '3', '4']),
         '2': set(['0']),
         '3': set(['1']),
         '4': set(['2', '3'])}

dfs(graph, '0')
```

Κώδικας 2 - DFS Python

3.7.3 Πολυπλοκότητα

Η χρονική πολυπλοκότητα του αλγορίθμου DFS αναπαρίσταται με τη μορφή O(V + E), όπου V είναι ο αριθμός των κόμβων και E είναι ο αριθμός των ακμών. Η χωρική πολυπλοκότητα του αλγορίθμου είναι O(V). [9]

3.7.4 Εφαρμογές του Depth First Algorithm

Εφαρμογές αλγορίθμου DFS:

- Για την εύρεση του μονοπατιού
- Για να ελέγξουμε εάν το γράφημα είναι διμερές
- Για την εύρεση των ισχυρά συνδεδεμένων στοιχείων ενός γραφήματος

- Για την ανίχνευση κύκλων σε ένα γράφημα [9]

3.8 Breadth First Algorithm

Ένας αλγόριθμος διέλευσης γραφήματος που ονομάζεται αναζήτηση πλάτους-πρώτα ερευνά κάθε κόμβο στο γράφημα ξεκινώντας από τον ριζικό κόμβο. Στη συνέχεια επιλέγει τον πλησιέστερο κόμβο και ερευνά κάθε κόμβο που δεν έχει εξερευνηθεί. Οποιοσδήποτε κόμβος στο γράφημα μπορεί να χρησιμεύσει ως ο ριζικός κόμβος όταν χρησιμοποιείται BFS για διέλευση. Υπάρχουν και άλλες τεχνικές για την πλοήγηση στο γράφημα, ωστόσο το BFS είναι η μέθοδος που εφαρμόζεται συχνότερα. Η διαδικασία αναζήτησης κάθε κορυφής σε μια δομή δεδομένων δέντρου ή γραφήματος είναι αναδρομική. Κάθε κορυφή στο γράφημα χωρίζεται σε δύο ομάδες στον BFS: επισκέφθηκε και δεν επισκέφθηκε. Επιλέγει έναν μεμονωμένο κόμβο σε ένα γράφημα και, μετά από αυτό, επισκέπτεται όλους τους κόμβους που βρίσκονται δίπλα στον επιλεγμένο κόμβο. [10]

3.8.1 Χρήσεις του Breadth First Algorithm

Ακολουθούν μερικά παραδείγματα εφαρμογών αλγορίθμων πρώτου εύρους:

- Από μια δεδομένη τοποθεσία πηγής, το BFS μπορεί να χρησιμοποιηθεί για τον εντοπισμό κοντινών τοποθεσιών.
- Ο αλγόριθμος BFS μπορεί να χρησιμοποιηθεί ως μέθοδος διέλευσης σε ένα δίκτυο peer-to-peer για τον εντοπισμό κάθε γειτονικού κόμβου.
- Το BFS μπορεί να χρησιμοποιηθεί σε προγράμματα ανίχνευσης ιστού για τη δημιουργία ευρετηρίων για ιστοσελίδες. Είναι ένας από τους πιο σημαντικούς αλγόριθμους για την ευρετηρίαση ιστοσελίδων. Ξεκινά το ταξίδι του στην αρχική σελίδα και στη συνέχεια πλοηγείται στους συνδέσμους της σελίδας. Κάθε ιστοσελίδα εμφανίζεται σε αυτήν την περίπτωση ως κόμβος στο γράφημα.
- Η συντομότερη διαδρομή και το λιγότερο εκτεινόμενο δέντρο βρίσκονται χρησιμοποιώντας το BFS.
- Μπορεί να χρησιμοποιηθεί για τον υπολογισμό της μέγιστης ροής σε ένα δίκτυο ροής χρησιμοποιώντας την προσέγγιση Ford-Fulkerson. [10]

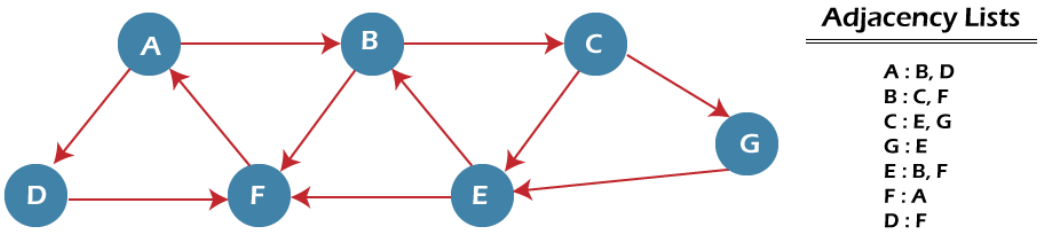
3.8.2 Ψευδοκώδικας

Ο ψευδοκώδικας για το BFS στην python έχει ως εξής:

- Δημιουργήστε μια ουρά Q
- Επισημάνετε το v ως επισκέψιμο και βάλτε το v στο Q
- Για όσο το Q δεν είναι κενό
- Αφαιρέστε το αρχικό στοιχείο u του Q
- Σημειώστε και βάλτε στην ουρά όλους τους (μη επισκέψιμους) γείτονες του u

3.8.3 Παράδειγμα Αλγορίθμου

Στο παράδειγμα που δίνεται παρακάτω, υπάρχει ένα κατευθυνόμενο γράφημα με 7 κορυφές.



Εικόνα 12 - Breadth First Algorithm

Στο παραπάνω γράφημα, η ελάχιστη διαδρομή 'P' μπορεί να βρεθεί χρησιμοποιώντας το BFS που θα ξεκινά από τον Κόμβο A και θα τελειώνει στον Κόμβο E. Ο αλγόριθμος χρησιμοποιεί δύο ουρές, δηλαδή QUEUE1 και QUEUE2. Η QUEUE1 κρατά όλους τους κόμβους που πρόκειται να υποβληθούν σε επεξεργασία, ενώ η QUEUE2 κρατά όλους τους κόμβους που υποβάλλονται σε επεξεργασία και διαγράφονται από την QUEUE1. Τώρα, ας αρχίσουμε να εξετάζουμε το γράφημα ξεκινώντας από τον Κόμβο A.

- Βήμα 1 - Πρώτα, προσθέστε το A στην ουρά 1 και το NULL στην ουρά 2.

```
1. QUEUE1 = {A}
2. QUEUE2 = {NULL}
```

Κώδικας 3 - Παράδειγμα BFS(0)

- Βήμα 2 - Τώρα, διαγράψτε τον κόμβο A από την ουρά 1 και προσθέστε τον στην ουρά 2. Εισαγάγετε όλους τους γείτονες του κόμβου A στην ουρά 1.

```
1. QUEUE1 = {B, D}
2. QUEUE2 = {A}
```

Κώδικας 4 - Παράδειγμα BFS (1)

- Βήμα 3 - Τώρα, διαγράψτε τον κόμβο B από την ουρά 1 και προσθέστε τον στην ουρά 2. Εισαγάγετε όλους τους γείτονες του κόμβου B στην ουρά 1.

```
1. QUEUE1 = {D, C, F}
2. QUEUE2 = {A, B}
```

Κώδικας 5 - Παράδειγμα BFS (2)

- Βήμα 4 - Τώρα, διαγράψτε τον κόμβο D από την ουρά 1 και προσθέστε τον στην ουρά 2. Εισαγάγετε όλους τους γείτονες του κόμβου D στην ουρά 1. Ο μόνος γείτονας του κόμβου D είναι ο F αφού έχει ήδη εισαχθεί, επομένως δεν θα εισαχθεί ξανά.

```
1. QUEUE1 = {C, F}
2. QUEUE2 = {A, B, D}
```

Κώδικας 6 - Παράδειγμα BFS (3)

- Βήμα 5 - Διαγράψτε τον κόμβο C από την ουρά 1 και προσθέστε τον στην ουρά 2. Εισαγάγετε όλους τους γείτονες του κόμβου C στην ουρά 1.

```
1. QUEUE1 = {F, E, G}
2. QUEUE2 = {A, B, D, C}
```

Κώδικας 7 - Παράδειγμα BFS (4)

- Βήμα 6 - Διαγράψτε τον κόμβο F από την ουρά 1 και προσθέστε τον στην ουρά 2. Εισαγάγετε όλους τους γείτονες του κόμβου F στην ουρά 1. Επειδή όλοι οι γείτονες του κόμβου F είναι ήδη παρόντες, δεν θα τους εισαγάγουμε ξανά.

```
1. QUEUE1 = {E, G}
2. QUEUE2 = {A, B, D, C, F}
```

Κώδικας 8 - Παράδειγμα BFS (5)

- Βήμα 7 - Διαγράψτε τον κόμβο E από την ουρά 1. Δεδομένου ότι όλοι οι γείτονές του έχουν ήδη προστεθεί, δεν θα τους προσθέσουμε ξανά. Τώρα, όλοι οι κόμβοι επισκέπτονται και ο κόμβος στόχος E συναντάται στην ουρά 2. [10]

```
1. QUEUE1 = {G}
2. QUEUE2 = {A, B, D, C, F, E}
```

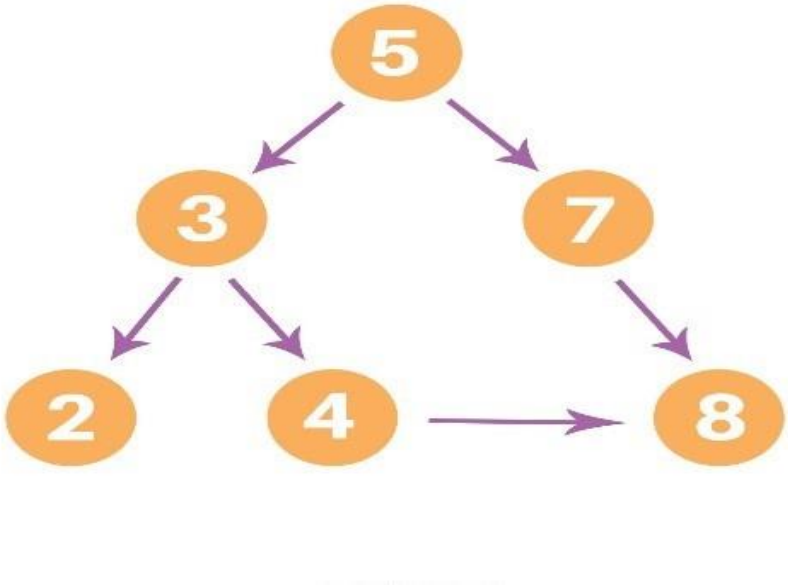
Κώδικας 9 - Παράδειγμα BFS (6)

3.8.4 Πολυπλοκότητα Αλγορίθμου

Η χρονική πολυπλοκότητα του BFS εξαρτάται από τη δομή δεδομένων της αναπαράστασης γραφήματος. Δεδομένου ότι ο αλγόριθμος BFS αναζητά κάθε κόμβο και άκρη στο χειρότερο σενάριο, η χρονική του πολυπλοκότητα είναι O(V+E). Ο αριθμός των ακμών σε ένα γράφημα είναι O, ενώ ο αριθμός των κορυφών είναι O(V) (E). Ο αριθμός των κορυφών, V, καθορίζει την πολυπλοκότητα του χώρου του BFS, η οποία συμβολίζεται με το σύμβολο O(V). [10]

3.8.5 Κώδικας Breadth First Algorithm σε python

Παρακάτω φαίνεται παράδειγμα του αλγορίθμου breadth first algorithm στη γλώσσα python.



Εικόνα 13 - Breadth First Algorithm

```
graph = {
  '5' : ['3','7'],
  '3' : ['2', '4'],
  '7' : ['8'],
  '2' : [],
  '4' : ['8'],
  '8' : []
}

visited = []
queue = []

def bfs(visited, graph, node):
  visited.append(node)
  queue.append(node)

  while queue:
    m = queue.pop(0)
    print (m, end = " ")

    for neighbour in graph[m]:
      if neighbour not in visited:
        visited.append(neighbour)
        queue.append(neighbour)

print("Following is the Breadth-First Search")
bfs(visited, graph, '5')
```

Κώδικας 10 - Breadth First Algorithm Python

Στον παραπάνω κώδικα, δημιουργούμε πρώτα το γράφημα για το οποίο θα χρησιμοποιηθεί η αναζήτηση κατά πλάτος. Μετά τη δημιουργία, θα φτιάξουμε δύο λίστες: μία για να παρακολουθούμε τους κόμβους που επισκέφθηκαν το γράφημα και την άλλη για να παρακολουθούμε τους κόμβους στην ουρά. Ακολουθώντας την προαναφερθείσα διαδικασία, θα δηλώσουμε μια συνάρτηση με τις ακόλουθες παραμέτρους: κόμβους επίσκεψης, το πραγματικό γράφημα και τον κόμβο. Επιπλέον, θα συνεχίσουμε να προσθέτουμε τις λίστες επισκέψεων και ουρών σε μια συνάρτηση. Στη συνέχεια, θα αφαιρέσουμε τον ίδιο κόμβο και θα τον εκτυπώσουμε όπως επισκέπτεται πριν εκτελέσουμε τον βρόχο while για την ουρά για την επίσκεψη στους κόμβους. Για να προσαρτήσουμε τους κόμβους που δεν έχουν επισκεφτεί από τη λίστα επισκέψεων και ουρών, θα εκτελέσουμε τέλος τον βρόχο for για να τους ελέγξουμε. Θα ζητήσουμε από τον χρήστη να ορίσει τη συνάρτηση bfs με τον πρώτο κόμβο που θέλουμε να επισκεφτούμε ως μέρος του κώδικα προγράμματος οδήγησης. [11]

3.9 Σύγκριση Αλγορίθμων

Και οι δύο μέθοδοι αναζήτησης το κάνουν με την υπέρθεση ενός δέντρου το δέντρο αναζήτησης πάνω από το γράφημα. Το DFS και το BFS ορίζουν τον κόμβο έναρξης ως ρίζα τους και τον επεκτείνουν προσθέτοντας τα φύλλα που θα διαδεχθούν τα τρέχοντα φύλλα του δέντρου. Το DFS και το BFS διασχίζουν έτσι ολόκληρο το γράφημα μέχρι να φτάσουν στον κόμβο προορισμού ή να φτάσουν στο όριό τους. Η σειρά με την οποία επισκέπτονται τους κόμβους, ή τους προσθέτουν στο δέντρο αναζήτησης, είναι όπου αποκλίνουν. Το DFS προσθέτει ένα θυγατρικό από τον πιο πρόσφατα συμπεριλαμβανόμενο κόμβο που έχει τουλάχιστον ένα μη συμπεριλαμβανόμενο παιδί στο δέντρο σε κάθε βήμα. Το DFS εισάγει πρώτα τον κόμβο έναρξης, ακολουθούμενο από το παιδί του, το εγγόνι του και ούτω καθεξής. Εξαιτίας αυτού, το DFS καταβάλλει κάθε δυνατή προσπάθεια για να εμβαθύνει το δέντρο αναζήτησης σε κάθε φάση. Στη συνέχεια, η διαδικασία επιστρέφει στον γονέα ενός κόμβου εάν δεν υπάρχουν άλλα παιδιά ενός κόμβου για προσθήκη. Το BFS, από την άλλη πλευρά, χτίζει το δέντρο στρώμα-στρώμα. Πρώτον, το επίπεδο 1 ολοκληρώνεται προσθέτοντας κάθε παιδί του κόμβου έναρξης. Στη συνέχεια, προσθέτει έναν προς έναν κάθε απογόνους από κάθε φύλλο στο πρώτο επίπεδο. Στη συνέχεια, προσθέτει καθένα από τα εγγόνια των απογόνων του αρχικού κόμβου και ούτω καθεξής. Το BFS διευρύνει το δέντρο σε κάθε επίπεδο εάν ο συντελεστής διακλάδωσης είναι σταθερός σε όλα τα επίπεδα. Το BFS το κάνει αυτό προσθέτοντας ένα θυγατρικό από τον λιγότερο πρόσφατα προστιθέμενο κόμβο που έχει τουλάχιστον ένα μη συμπεριλαμβανόμενο παιδί στο δέντρο, το οποίο είναι ακριβώς το αντίθετο από αυτό που κάνει ο DFS. [12]

3.10 Διαφορές Πολυπλοκότητας

- Ως προς τον χρόνο.
  Στη χειρότερη περίπτωση, ο DFS δημιουργεί ένα δέντρο αναζήτησης με m επίπεδα, καθιστώντας τη χρονική του πολυπλοκότητα O(bm). Το BFS είναι το βέλτιστο, επομένως το χειρότερο σενάριο είναι ότι βρίσκει τον κόμβο στόχο αφού προσθέσει b κόμβους στο επίπεδο 1, b2 στο επίπεδο 2 και ούτω καθεξής μέχρι το επίπεδο d, όπου προσθέτει κόμβους bd. Το δέντρο BFS στη χειρότερη περίπτωση έχει συνολικά 1+b+b2+ldots+bd κόμβους, κάτι που δείχνει ότι ο χρόνος που απαιτείται για την ανάπτυξή του είναι της τάξης O(bd). [12]

- Ως προς τον χώρο.
  Δεδομένου ότι το BFS συνεχίζει να προσθέτει περισσότερους κόμβους στα σύνορα, αλλά εμφανίζει τον παλαιότερο από αυτούς, πρέπει να τους αποθηκεύσει όλους. Έτσι, η πολυπλοκότητα του χώρου του στο χειρότερο σενάριο είναι O(bd) επειδή αυτό είναι το χειρότερο μέγεθος των συνόρων. Ο DFS είναι διαφορετικός. Αφού εξερευνήσει ολόκληρο το υποδέντρο του οποίου η ρίζα είναι ο κόμβος u, ο DFS δοκιμάζει τα αδέρφια του u. Όμως, ενώ επεξεργάζεται το εσένα και τους απογόνους του, ο DFS δεν χρειάζεται να διατηρεί τους απογόνους των αδερφών του στην κύρια μνήμη. Μπορεί να εστιάσει μόνο στην τρέχουσα ενεργή διαδρομή και το μόνο που χρειάζεται για οπισθοδρόμηση είναι τα αδέρφια των κόμβων σε αυτό. Έτσι, η χειρότερη πολυπλοκότητα χώρου του DFS O(bd). Μπορούμε να μειώσουμε την πολυπλοκότητα του DFS εάν αποδώσουμε τα παιδιά του βαθύτερου κόμβου στην τρέχουσα ενεργή διαδρομή ένα προς ένα αντί να τα επιστρέψουμε όλα ταυτόχρονα. Σε αυτή την περίπτωση, αποθηκεύουμε όχι περισσότερους από d κόμβους σε οποιοδήποτε σημείο κατά την εκτέλεση. Και μπορούμε να πάμε ακόμα παραπέρα. Εάν μπορούμε να μετατρέψουμε τους κόμβους ο ένας στον άλλο, μπορούμε να χρησιμοποιήσουμε έναν μόνο κόμβο και να εφαρμόσουμε μετασχηματισμούς όταν τον αντικαθιστούμε με το παιδί ή τα αδέρφια του. Ως αποτέλεσμα, έχουμε μια πολυπλοκότητα χώρου O(1), αλλά οι μετασχηματισμοί πρέπει να είναι αναστρέψιμοι λόγω της οπισθοδρόμησης. [12]

3.11 Επίλογος

Τέλος ενώ ο BFS έχει ορισμένα θεωρητικά πλεονεκτήματα σε σχέση με το DFS, δεν είναι πρακτικό λόγω της υψηλής πολυπλοκότητας του χώρου του. Αυτός είναι ο λόγος που χρησιμοποιούμε DFS πιο συχνά.

Κεφάλαιο 4ο: Enhanced Web Bot

4.1 Εισαγωγή

Το enhanced web bot είναι ένα script που είναι γραμμένο στη γλώσσα python. Χρησιμοποιεί τεχνολογίες αυτοματισμού στον περιηγητή ιστού ώστε να εκτελεί αυτοματοποιημένες λειτουργίες. Σκοπός του είναι να αναλύει και να εξάγει από μια ιστοσελίδα όλους τους συνδέσμους της μαζί με τα χαρακτηριστικά τους και να περιηγείται σε αυτούς με βάση μια δεντρική δομή με στόχο την ανάλυση του ιστού.

4.2 Τι είναι web bot

Ένα απλό web bot είναι ένα πρόγραμμα υπολογιστή που λειτουργεί ως πράκτορας για έναν χρήστη, ένα άλλο πρόγραμμα ή για την προσομοίωση ανθρώπινης δράσης είναι γνωστό ως bot, το οποίο είναι συντομογραφία του ρομπότ και είναι επίσης γνωστό ως bot Διαδικτύου. Τα bots χρησιμοποιούνται συνήθως για την αυτοματοποίηση συγκεκριμένων εργασιών, ώστε να μπορούν να χρησιμοποιηθούν χωρίς συγκεκριμένες ανθρώπινες οδηγίες. Ένας οργανισμός ή ένα άτομο μπορεί να χρησιμοποιήσει ένα bot για να πάρει τη θέση ενός ανθρώπου που διαφορετικά θα έπρεπε να κάνει μια επαναλαμβανόμενη εργασία. Επιπλέον, τα bots είναι πολύ πιο γρήγορα από τα άτομα σε αυτές τις εργασίες. Τα ρομπότ μπορούν να κάνουν ωφέλιμες εργασίες, αλλά μπορεί επίσης να είναι κακόβουλα και μεταμφιεσμένα ως κακόβουλο λογισμικό. [13]

4.2.1 Τύποι bot

Τα bots διατίθενται σε διάφορες μορφές, το καθένα με τις δικές του λειτουργίες και στόχους. Μεταξύ των δημοφιλών bots είναι τα ακόλουθα:

- Chatbots. Αυτά τα εργαλεία μπορούν να μιμηθούν την επικοινωνία από άνθρωπο σε άνθρωπο. Οι εικονικοί βοηθοί όπως το Alexa της Amazon, το Siri της Apple και το Google Assistant είναι πιο πρόσφατα παραδείγματα chatbot.
- Κοινωνικά ρομπότ Αυτά τα ρομπότ, τα οποία αναφέρονται συχνά ως ρομπότ γνώμης, έχουν αντίκτυπο στις συνομιλίες των χρηστών στους ιστότοπους κοινωνικής δικτύωσης.
- Shopbots. Πολλές από αυτές τις εφαρμογές αναζητούν στο Διαδίκτυο τη μεγαλύτερη προσφορά για ένα προϊόν που ενδιαφέρεται να αγοράσει ένας χρήστης. Το Shopify chatbot και άλλα shopbots επιτρέπουν στους ιδιοκτήτες καταστημάτων να αυτοματοποιούν το μάρκετινγκ και την εξυπηρέτηση πελατών.
- Knowbots. Αυτά τα εργαλεία βοηθούν τους χρήστες να μαθαίνουν κάνοντας αυτόματη περιήγηση σε ιστότοπους και ανακτώντας δεδομένα που ικανοποιούν προκαθορισμένα κριτήρια. Αρχικά, τα knowbots χρησιμοποιήθηκαν ως πλεονάζοντες βοηθοί μέσω υπολογιστή.
- Crawlers ή Spiders. Είναι είδη web bot που συχνά αναφέρονται ως ανιχνευτές ιστού, περιηγούνται σε ιστότοπους και συλλέγουν περιεχόμενο για ευρετήρια σε μηχανές αναζήτησης όπως το Google και το Bing.
- Web scraping crawlers. Παρόμοια με τα προγράμματα ανίχνευσης, αλλά με σκοπό τη συλλογή δεδομένων και την ανάκτηση σχετικών πληροφοριών από ιστότοπους.
- Transactions bots. Αυτά τα ρομπότ είναι φτιαγμένα για να κάνουν πιο απλές ενέργειες που συνήθως θα χειριζόταν ένας άνθρωπος μέσω τηλεφώνου, όπως ο αποκλεισμός μιας πιστωτικής

κάρτας για την οποία έχει αναφερθεί κλοπή ή η επαλήθευση των ωρών λειτουργίας μιας τράπεζας. [13]

4.3 Τεχνολογίες που χρησιμοποιεί

Οι τεχνολογίες που χρησιμοποιεί τον enhanced web bot είναι οι εξής:

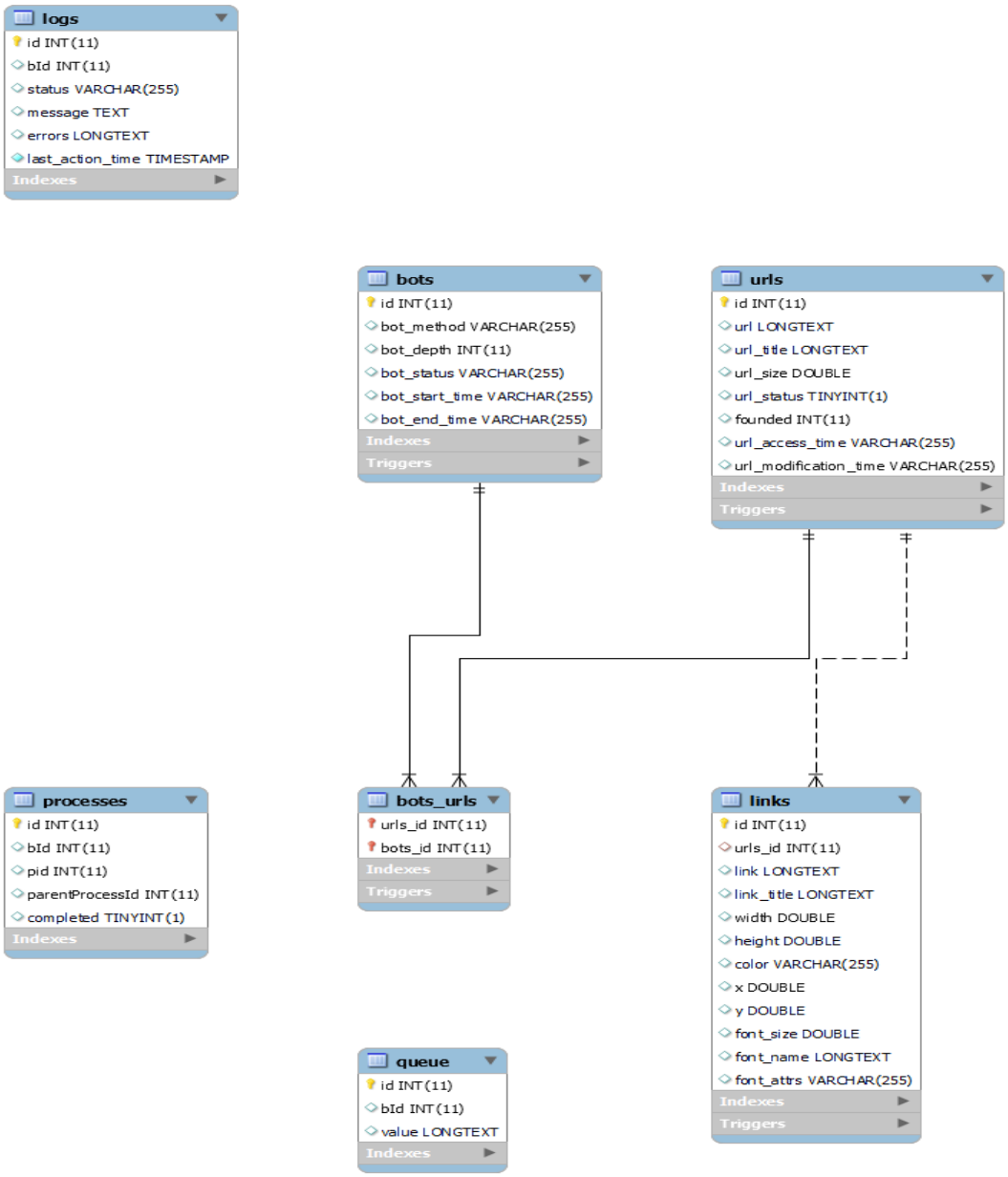
- Selenium web driver για να εκτελεί αυτοματοποιημένες λειτουργίες του περιηγητή ιστού αλλά και για την εξαγωγή τον δεδομένων από το html elements.
- MySQL για σχεσιακή βάση δεδομένων και χρησιμοποιείται για την αποθήκευση των δεδομένων που εξάγονται.

4.4 Αρχή Λειτουργίας

Αρχικά το enhanced web bot ανήκει στην κατηγορία των bot τα web scraping crawlers που χρησιμοποιούνται κυρίως στον περιηγητή ιστού. Κατά την εκκίνηση του web bot από τη κονσόλα του υπολογιστή ζητείται από τον χρήστη να δοθούν κάποια ορίσματα διότι είναι ένα python script. Τα απαραίτητα ορίσματα για να ξεκινήσει είναι η μέθοδος δηλαδή ο αλγόριθμος που θα χρησιμοποιηθεί. Αν θα είναι ο depth first algorithm ή ο breadth first algorithm για την τρόπο περιήγησης στους συνδέσμους του ιστού. Στη συνέχεια ζητείται το μέγιστο βάθος κόμβων που επιθυμούμε να εξερευνήσει το bot όπως και τη αρχική ή σύνολο αρχικών ιστοσελίδων που επιθυμούμε να δώσουμε σαν είσοδο ώστε να ξεκινήσει η εξερεύνηση. Αφού δοθούν τα απαραίτητα ξεκινάει το web bot την εξερεύνηση από την αρχική ιστοσελίδα και χρησιμοποιώντας τον αλγόριθμο που επιλέχθηκε για την περιήγηση πηγαίνει σε κάθε σύνδεσμο της ιστοσελίδας τον κρατάει εξάγοντας πληροφορίες όπως το όνομά του, τον ίδιο τον σύνδεσμο, το μέγεθός του, το που βρίσκεται μέσα στην σελίδα τις συντεταγμένες του, το μέγεθος της γραμματοσειράς του αλλά και όλες τις ιδιότητες γραμματοσειράς και το χρώμα του. Αφού γίνουν αυτά αποθηκεύονται όλες οι πληροφορίες στη βάση δεδομένων και συνεχίζει στον επόμενο σύνδεσμο με την ίδια ακριβώς λειτουργία. Μόλις εξερευνήσει όλους τους συνδέσμους της αρχικής σελίδας, συνεχίζει πιο βαθιά παίρνοντας σαν συνέχεια τους συνδέσμους που βρήκε ήδη και βρίσκει για τον καθένα τους δικούς του σύνδεσμους και συνεχίζει έως ότου φτάσει το επιθυμητό βάθος που δόθηκε στην αρχή. Επίσης αποθηκεύει και βασικές πληροφορίες για τις σελίδες που επισκέπτεται όπως το όνομά της, το μέγεθος που έχει, το url της όπως και την ώρα πρόσβασης και τροποποίησης. Με το τέλος εκτέλεσης του web bot υπάρχουν στη βάση δεδομένων όλες οι πληροφορίες για τις ιστοσελίδες που επισκέφθηκε και τους συνδέσμους που βρήκε με όλες τις ιδιότητές τους αλλά τη σειρά περιήγησης και πως βρέθηκε από κάποιον σε κάποιον άλλο και γενικά το τελικό δέντρο που δημιουργήθηκε με το πέρας της εκτέλεσης του. Πλέον με την παρακολούθηση των τελικών δεδομένων μπορεί να υπολογιστεί η σημασιολογία των συνδέσμων σε μια ιστοσελίδα με βάση τις πληροφορίες που αποκτήθηκαν.

4.5 Βάση Δεδομένων

Η σχεσιακή βάση δεδομένων που χρησιμοποιήθηκε είναι η MySQL. Το σχήμα και οι πίνακες φαίνονται παρακάτω:



Εικόνα 14 - Διάγραμμα Er Βάσης

```
CREATE TABLE bots(
     id integer auto_increment,
    bot_method varchar(255),
    bot_depth integer,
    bot_status varchar(255),
    bot_start_time varchar(255),
    bot_end_time varchar(255),
    primary key(id)
);

CREATE TABLE urls(
     id integer auto_increment,
     url longtext unique,
    url_title longtext,
    url_size real,
    url_status bool,
    founded integer,
    url_access_time varchar(255),
    url_modification_time varchar(255),
    primary key(id)
);

CREATE TABLE bots_urls(
     urls_id integer ,
    bots_id integer,
    primary key(urls_id,bots_id),
    foreign key(bots_id) references bots(id)  on delete cascade on
update cascade,
     foreign key(urls_id) references urls(id)  on delete cascade on
update cascade
);
```

Κώδικας 11 - DDL Βάσης

```
CREATE TABLE links(
     id integer auto_increment,
    urls_id integer,
    link longtext,
    link_title longtext,
    width real,
    height real,
    color varchar(255),
    x real,
    y real,
    font_size real,
    font_name longtext,
    font_attrs varchar(255),
    primary key(id),
    foreign key(urls_id) references urls(id) on delete cascade on
update cascade
);


CREATE TABLE logs(
     id integer auto_increment,
     bId integer,
     status varchar(255),
    message text,
    errors longtext default null,
    last_action_time timestamp default current_timestamp,
    primary key(id)
);
```

Κώδικας 12 - DDL Βάσης

4.6 Render

Το enhanced web bot είναι σε θέση όταν κατεβάζει μια σελίδα να την κάνει render και μετά να ξεκινάει την περιπλάνηση στο html dom της δέντρο. Με αυτό τον τρόπο έχουμε στα χέρια μας την τελική σελίδα σε περίπτωση που είναι δυναμική εκτελούνται πρώτα όλα τα javascript αρχεία και μετά έχουμε την τελική εικόνα με όλα τα δεδομένα. Σε αυτό βοηθάει το selenium web driver που ανοίγει κρυφά έναν

περιηγητή ιστού και με αυτόν τον τρόπο ανοίγει τις σελίδες. Στην ουσία αντιγράφει την συμπεριφορά ενός χρήστη καθώς ανοίγει μια ιστοσελίδα μέσω ενός περιηγητή ιστού.

4.7 Timeout – Retry Queue

Το enhanced web bot καθώς κατεβάζει μια ιστοσελίδα είναι σε θέση σε περίπτωση αστοχίας μιας ιστοσελίδας να ξανά προσπαθήσει να την κατεβάσει ώστε να συνεχιστεί ομαλά η ροή του προγράμματος. Σε περίπτωση που δεν τα καταφέρει πάλι την αποθηκεύει στην βάση σαν αποτυχία και συνεχίζει κανονικά η ροή του προγράμματος στον επόμενο σύνδεσμο. Στο τέλος της εκτέλεσης παίρνει όλες τις αποτυχίες από τη βάση και τα ξανά προσπαθεί να τα κατεβάσει.

```
from threading import Thread, Event
import time


stop_event = Event()
def timer():
    i = 0
    while True:
        i += 1
        print(f"-- Retrying again... In: {i}")
        time.sleep(1)
        if stop_event.is_set():
            break


def TimerThread(threadName, targetFunction, time):
    threadName = Thread(target=targetFunction)
    threadName.start()
    threadName.join(timeout=time)
    stop_event.set()
    print("-- Restarted --")
```

Κώδικας 13 - Retry - Timeout

4.8 Redirecting

Το web bot καθώς κατεβάζει μια ιστοσελίδα υπάρχει περίπτωση να το κάνει αυτόματα redirect σε άλλη σελίδα. Αυτή τη κατάσταση την διαχειριζόμαστε με τον έλεγχο αν η σελίδα που κατεβάζει τώρα είναι ίδια με την είσοδο που ξεκίνησε αν δεν είναι λέμε ότι έγινε redirect η μια σελίδα στην άλλη και αποθηκεύουμε αυτή τη πληροφορία κατάλληλα στη βάση ώστε να γνωρίζουμε τη ροή του δέντρου και το πως περιηγήθηκε το bot.

4.9 Multiprocessing

Το web bot έχει δύο λειτουργίες εκκίνησης. Μπορεί να ξεκινήσει κανονικά ένα web bot να δεχθεί μια σελίδα ή ένα σύνολο σελίδων και ένα web bot θα τις πάρει σειριακά και θα κάνει την απαραίτητη

λειτουργία. Ωστόσο υπάρχει η δυνατότητα να δημιουργηθούν πολλά web bot παράλληλα και κάθε web bot να αναλάβει διαφορετική είσοδο ή περισσότερα bot να αναλάβουν μια σελίδα και να κατεβάζουν παράλληλα διαφορετικά κλαδιά του γράφου ώστε να επιτευχθεί καλύτερος χρόνος εκτέλεσης.

4.10 Διαχείριση των λαθών

Έχει υλοποιηθεί μια μέθοδος σαν wrapper για την διαχείριση όλων λαθών του προγράμματος παίρνει στο σώμα της μία μέθοδο για παράδειγμα τον αλγόριθμο που τρέχει και μπορεί να καταγράφει τα λάθη που συμβαίνουν και να τα στέλνει κατευθείαν στο αρχείο log. Στην ουσία έχει δημιουργηθεί μια μέθοδος για να ελέγχει όλα τα λάθη της εφαρμογής.

```
def exceptionsHandler(self, methodToRun, *args):
    try:
        methodToRun(*args)
        print("function running..")
    except StaleElementReferenceException as stl:
        print(stl)
        self.errors.append(stl)
    except WebDriverException as wbex:
        print(wbex)
        self.errors.append(wbex)
    except ConnectionError as conErr:
        print(conErr)
        self.errors.append(conErr)
        print("here an error..")
    except TypeError as typeErr:
        print(typeErr)
        self.errors.append(typeErr)
    except Exception as e:
        print(e)
        self.errors.append(e)
```

Κώδικας 14 - Error Handler

4.11 Αρχείο Log

Έχει δημιουργηθεί ένας πίνακας στη βάση που ονομάζεται log file και εκεί αποθηκεύονται όλες οι ενέργειες του bot. Αυτό σημαίνει ότι εκεί καταγράφονται από το πότε δημιουργήθηκε το bot και ποιο είναι αυτό, τα urls και τα links που κατεβάζει καθώς και όλα τα λάθη που καταγράφονται συνολικά. Τέλος αυτά εξάγονται σε αρχεία log για κάθε bot ξεχωριστά. Αυτή η διαχείριση του log γίνεται μέσω εναύσματα (triggers) στη βάση δεδομένων και έτσι καταγράφονται με αυτόματο τρόπο όλες οι ενέργειες.

```
    def saveErrorsToDbLog(self):
        if self.errors:
            for i in self.errors:
                sql = "INSERT INTO logs(bId, status, message,
errors) VALUES (%s,%s,%s,%s)"
                cursor = self.db.cursor()
                cursor.execute(
                        sql, (self.botId , "ERROR - TERMINATED",
"ERROR: See the errors tab", f"An Error has occured -- {i}.")
                        )
                self.db.commit()



    def exportLogFiles(self):
        now = datetime.now().strftime(
                    "%d-%m-%Y-%H-%M-%S")


        Log_Format = "%(levelname)s %(asctime)s - %(message)s"


        logging.basicConfig(filename = f"logs/{self.botId}-logfile-
{now}.log",
                    filemode = "w",
                    format = Log_Format,
                    level = logging.NOTSET)


        logger = logging.getLogger()


        sql = "SELECT * FROM logs WHERE bId = %s"
        cursor = self.db.cursor()
        cursor.execute(
                sql, ( self.botId,)
                )
        myresult = cursor.fetchall()
```

Κώδικας 15 - Log File

4.12 Συνέχιση του bot

Σε περίπτωση διακοπής του bot για οποιοδήποτε λόγο υπάρχει η δυνατότητα με βάση το αναγνωριστικό (id) του bot να μπορεί να συνεχίσει τη λειτουργία του από εκεί που είχε σταματήσει. Προφανώς δεν

ξανά κατεβάζει τις ίδιες ιστοσελίδες και τους συνδέσμους ώστε να αποφευχθούν τα διπλότυπα στη βάση. Απλά ακολουθεί τον προηγούμενο γράφο και αποθηκεύει μόνο τη νέα πληροφορία.

```
def botFind(self, id):
        sql = "SELECT * FROM bots WHERE id = %s LIMIT 1"
        cursor = self.db.cursor()
        cursor.execute(
                sql, ( id, )
                )
        myresult = cursor.fetchall()
        if myresult:
            #print(myresult)
            return myresult
        else:
             raise ValueError(
                'No bot with this id.')


    def botContinue(self, id):
        sql = "SELECT u.url FROM bots b inner join bots_urls bu
on(b.id = bu.bots_id) inner join urls u on(u.id = bu.urls_id) WHERE
b.id = %s"
        cursor = self.db.cursor()
        cursor.execute(
                sql, ( id, )
                )
        myresult = cursor.fetchall()
        if myresult:
            #print(myresult)
            mylist = []
            for i in myresult:
                mylist.append(i[0])
            return mylist
        else:
             raise ValueError(
                'An Error Occured!.')
```

Κώδικας 16 - Bot Continue (1)

```
    def saveLinksToDb(self, data):
        doublicateLinks = self.checkForDoublicateLinks(data)
        if (not doublicateLinks):
            now = datetime.now().strftime(
                    "%d/%m/%Y %H:%M:%S")
            sql = "INSERT INTO
links(urls_id,link,link_title,width,height,color,x,y,font_size,font_
name,font_attrs) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
            fontSize = data['font_size'].replace('px','')
            cursor = self.db.cursor()
            cursor.execute(
                sql, (self.urlid, data['link'], data['link_title'],
data['width'], data['height'], data['color'], data['x'], data['y'],
fontSize, data['font_name'], data['font_attrs'])
                )
            self.db.commit()



    def checkForDoublicateUrl(self, data):
        sql = "SELECT * FROM urls WHERE url = %s LIMIT 1"
        cursor = self.db.cursor()
        cursor.execute(
                sql, ( data["url"],)
                )
        myresult = cursor.fetchall()


        #print(myresult[0][0])
        if myresult:
            return myresult
        return False



    def checkForDoublicateBotsUrl(self, data):
        sql = "SELECT * FROM bots_urls WHERE urls_id = %s AND
bots_id= %s LIMIT 1"
```

Κώδικας 17 - Bot Continue (2)

4.13 Επίλογος

Τα bot είναι πολύ χρήσιμα στις μέρες μας γιατί μπορούν να εκτελούν επαναλαμβανόμενες και αυτοματοποιημένες εργασίες για εμάς χωρίς την ανθρώπινη επέμβαση. Ωστόσο τα bot ιστού είναι δημοφιλή διότι όλες οι εργασίες ενός ανθρώπου στον υπολογιστή εκτελούνται μέσω του περιηγητή

ιστού και μπορούν να αντικαταστήσουν πολλές καθημερινές λειτουργίες και να γίνονται αυτοματοποιημένα και πιο αποτελεσματικά.

Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

5.1 Προτάσεις Βελτίωσης

Τέλος το enhanced web bot θα μπορούσε να βελτιωθεί μελλοντικά όσον αφορά την οπτικοποίηση των αποτελεσμάτων αλλά και της λειτουργείας του σε πραγματικό χρόνο. Θα μπορούσε να φτιαχτεί ένα rest api σε κάποιο framework για παράδειγμα .net και να δημιουργηθούν κάποια endpoints που να αντιστοιχούν στις ενέργειες του web bot να το καλεί το api δηλαδή. Μετέπειτα θα μπορούσε να φτιαχτεί ένας client σε κάποιο άλλο framework για παράδειγμα vue js το οποίο θα επικοινωνεί με το api και θα μπορεί να οπτικοποιεί σε πραγματικό χρόνο τα bot πως ξεκινάνε, τι url και links θα βρίσκει όπως και τις επιπλέον τους πληροφορίες και γενικά να δείχνει με όμορφη αναπαράσταση τον γράφο των αποτελεσμάτων και πως περιηγείται γενικά. Θα μπορούσε να δείχνει και τα σφάλματα σε πραγματικό χρόνο αλλά και να υπάρχει γενικά αλληλεπίδραση ώστε να υπάρχει πλήρης έλεγχος γενικά του τι συμβαίνει.

5.2 Συμπεράσματα

Τα συμπεράσματα που προκύπτουν αφορούν τον τελικό στόχο του enhanced web bot στο οποίο ο χρήστης μπορεί να υπολογίσει τη σημασιολογία που έχουν οι σύνδεσμοι στη σελίδα μέσω ανάλυσης διαφόρων παραγόντων. Αυτοί οι παράγοντες μπορεί να είναι η θέση που έχει ένας σύνδεσμος μέσα σε μια σελίδα για παράδειγμα αν βρίσκεται στο κέντρο της ή στο τέλος της σελίδας, οι γραμματοσειρές που χρησιμοποιεί αν είναι μεγάλες ή μικρές αλλά και το είδος τους. Τέλος σε συνδυασμό και με τις επιπλέον πληροφορίες που παίρνουμε μαζί για κάθε σύνδεσμο όπως το όνομά του, το χρώμα του και τον ίδιο τον σύνδεσμο αλλά και το πως βρεθήκαμε σε αυτόν ή έγινε ανακατεύθυνση από άλλον μας βοηθάει να εξάγουμε τα τελικά μας συμπεράσματα για το πόσο σημαντικός είναι αυτός ο σύνδεμος.

Βιβλιογραφία

[1] R. Taracha, "medium.com," 4 Σεπτέμβριος 2017. [Online]. Available: https://medium.com/the-andela-way/introduction-to-web-scraping-using-selenium-7ec377a8cf72. [Accessed 09/06/2022 Ιούνιος 2022].

[2] From Wikipedia, the free encyclopedia, «World Wide Web,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/World_Wide_Web. [Πρόσβαση 18 08 22].

[3] B. Simplilearn, «What is Web 1.0, 2.0, and 3.0 and How They Compare,» Simplilearn, [Ηλεκτρονικό]. Available: https://www.simplilearn.com/what-is-web-1-0-web-2-0-and-web-3-0-with-their-difference-article. [Πρόσβαση 18 08 2022].

[4] Shubham Kumar, «Difference Between Library and Framework,» [Ηλεκτρονικό]. Available: https://www.c-sharpcorner.com/uploadfile/a85b23/framework-vs-library/. [Πρόσβαση 18 08 22].

[5] geeksforgeeks.org, «What is Web Scraping and How to Use It?,» [Ηλεκτρονικό]. Available: https://www.geeksforgeeks.org/what-is-web-scraping-and-how-to-use-it/. [Πρόσβαση 20 08 2022].

[6] Sri Manikanta Palakollu, «Scrapy Vs Selenium Vs Beautiful Soup for Web Scraping.,» [Ηλεκτρονικό]. Available: https://medium.com/analytics-vidhya/scrapy-vs-selenium-vs-beautiful-soup-for-web-scraping-24008b6c87b8. [Πρόσβαση 18 08 2022].

[7] Krishna Rungta, «What is Selenium? Introduction to Selenium Automation Testing,» [Ηλεκτρονικό]. Available: https://www.guru99.com/introduction-to-selenium.html. [Πρόσβαση 20 08 2022].

[8] David Loshin, «Data structures,» [Ηλεκτρονικό]. Available: https://www.techtarget.com/searchdatamanagement/definition/data-structure. [Πρόσβαση 25 08 2022].

[9] programiz.com, «Depth First Search (DFS),» [Ηλεκτρονικό]. Available: https://www.programiz.com/dsa/graph-dfs. [Πρόσβαση 30 08 2022].

[10] javatpoint.com, «BFS algorithm,» [Ηλεκτρονικό]. Available: https://www.javatpoint.com/breadth-first-search-algorithm. [Πρόσβαση 30 08 2022].

[11] favtutor.com, «Breadth First Search in Python (with Code) | BFS Algorithm,» [Ηλεκτρονικό]. Available: https://favtutor.com/blogs/breadth-first-search-python. [Πρόσβαση 30 08 2022].

[12] Milos Simic, «Depth-First Search vs. Breadth-First Search,» [Ηλεκτρονικό]. Available: https://www.baeldung.com/cs/dfs-vs-bfs. [Πρόσβαση 30 08 2022].

[13] Ben Lutkevich, «bot,» [Ηλεκτρονικό]. Available: https://www.techtarget.com/whatis/definition/bot-robot. [Πρόσβαση 31 08 2022].